ESO-MIDAS User Guide

Volume B: Data Reduction

MIDAS Release **98NOV**

Reference Number: MID-MAN-ESO-11000-0003

Section	Title	Date
Chapter 1	Introduction	31-March-1999
Chapter 2	Computational Methods	31-March-1999
Chapter 3	CCD Reductions	31-March-1999
Chapter 4	Object Search & Classification	1-November-1991
Chapter 5	Crowded Field Photometry	1-November-1990
Chapter 6	Long-Slit and 1D Spectra	1-November-1993
Chapter 7	Echelle Spectra	1-November-1994
Chapter 8	Inter-stellar/galactic Absorption Line Modelling	1-November-1989
Chapter 9	Test Data	15-January-1988
Chapter 10	Multivariate Analysis	1-November-1991
Chapter 11	DAOPHOT II	31-March-1999
Chapter 12	Time Series Analysis	1-November-1996
Chapter 13	PEPSYS General Photometry	31-January-1993
Chapter 14	The Wavelet Transform	1-November 1993
Chapter 15	Data Organizer	1-November 1993
Chapter 16	Astrometry	1-November 1994
Appendix A	Command Summary	31-March-1999
Appendix B	Detectors	31-March-1999
Appendix C	CES	1-May-1990
Appendix D	Echelle Reduction	1-November-1995
Appendix E	PISCO	1-November-1991
Appendix F	IRSPEC Reduction	1-November-1992
Appendix G	Reduction of Long Slit and 1D Spectra	1-November-1994
Appendix H	Optopus Package	1-November-1991
Appendix I	Photometry File Formats	1-November-1992
Appendix J	IRAC2	31-March-1999
Appendix K	CCD Test Procedures	31-March-1999
Appendix L	Multi-Object Spectroscopy	31-March-1999
Appendix M	FEROS	31-March-1999

EUROPEAN SOUTHERN OBSERVATORY
Data Management Division
Karl–Schwarzschild–Straße 2, D–85748 Garching bei München
Federal Republic of Germany

Contents

List of Figures

List of Tables

GNU GENERAL PUBLIC LICENSE

Version 2, June 1991

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software–to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Library General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

   Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You can copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

   You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

   a. You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.

   b. You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.

   c. If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

   a. Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,
   b. Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,
   c. Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

   The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

   If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

   If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

   It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

   This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

   Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING

BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

How To Apply These Terms To Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

one line to give the program's name and an idea of what it does.

```
Copyright (C) 19yy name of author

This program is free software; you can redistribute it and/or
modify it under the terms of the GNU General Public License
as published by the Free Software Foundation; either version 2
of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful,
but WITHOUT ANY WARRANTY; without even the implied warranty of
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
GNU General Public License for more details.

You should have received a copy of the GNU General Public License
along with this program; if not, write to the Free Software
Foundation, Inc., 675 Mass Ave, Cambridge, MA 02139, USA.
```

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this when it starts in an interactive mode:

```
Gnomovision version 69, Copyright (C) 19yy name of author
Gnomovision comes with ABSOLUTELY NO WARRANTY; for details
type `show w'.  This is free software, and you are welcome
to redistribute it under certain conditions; type `show c'
for details.
```

The hypothetical commands '`show w`' and '`show c`' should show the appropriate parts of the General Public License. Of course, the commands you use may be called something other than '`show w`' and '`show c`'; they could even be mouse-clicks or menu items–whatever suits your program.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the program, if necessary. Here is a sample; alter the names:

```
Yoyodyne, Inc., hereby disclaims all copyright
interest in the program 'Gnomovision'
(which makes passes at compilers) written
by James Hacker.
```

signature of Ty Coon, 1 April 1989

```
Ty Coon, President of Vice
```

This General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Library General Public License instead of this License.

Chapter 1

Introduction

1.1 How to use the MIDAS Manual

This document is intended to be a description of how to use the various facilities available in the MIDAS system. The manual consists of three volumes:

Volume A: describes the basic MIDAS system with all general purpose facilities such as MIDAS Control Language, data input/output (including graphics and image display), table system (MIDAS Data Base). A summary of all available commands as well as site specific features are given in appendices.

Volume B: describes how to use the MIDAS system for astronomical data reduction. Application packages for special types of data or reductions (*e.g.* long slit and echelle spectra, object search, or crowded field photometry) are discussed assuming intensity calibrated data. A set of appendices gives a detailed description of the reduction of raw data from ESO instruments.

Volume C: gives the detailed description for all commands available.

It is intended that users will mainly need Volume A for general reference. For specific reduction of raw data and usage of special astronomical packages, Volume B will be more informative. A printed version of the MIDAS help files is available in Volume C. Users are recommended to use the on-line help facility which always gives a full up to date description of the commands available.

1.1.1 New Users

To be able to use MIDAS, it is a great advantage to have some basic knowledge of computer systems such as how to login, use of the file editor and simple system commands. Such instructions can normally be found in local system documentation or in Appendix C of Volume A. After having acquired this knowledge, new users should read Chapter 2 Volume A carefully. This will give a basic introduction to the MIDAS system with some examples.

1.1.2 Site Specific Features

MIDAS is used at many different sites on a large variety of configurations. The main part of this manual does not refer to special configurations or hardware devices. Site specific implementations and details of the local installation can be found in Appendix C of Volume A.

1.2 Support

Most of our support services are available via the World Wide Web Homepage

`http://www.eso.org/esomidas/`

The MIDAS Users Support provides help for users that have problems or questions during installation or usage of the MIDAS system. It answers to questions cannot be obtained locally (e.g. through a manual) or after consulting our FAQ, please feel free to contact us for support. The most convinient and best way to contact us is using the **ESO-MIDAS Problem Report Form** that enable us to deal the reporting quickly using a set of processing tools. In addition this way of reporting offers us the possibility of maintaining a database of the questions and the update the FAQ listing.

The ESO-MIDAS Problem Form is available via the WWW

http://www.eso.org/esomidas/midas-support.html

and via the XHELP Graphical User Interface when running MIDAS. To start this interface enter: CREATE/GUI HELP. In case the ESO-MIDAS Problem Report System cannot be used the MIDAS User Support can be contacted via:

- `internet: midas@eso.org`
- `telefax: +49 89 32006480 (attn.: MIDAS HOT-LINE)`

Requests and questions are acknowledged when received and processed as soon as possible, normally within a few days. Also, users are strongly encouraged to send suggestions and comments via the MIDAS Problem Report Form that enables quick processing of your request.

1.3 Other Relevant Documents

There are several other documents relevant to the MIDAS system. General descriptions of the system can be found in the following references:

- Banse, K., Crane, P. Ounnas, C., Ponz, D., 1983 : 'MIDAS' in *Proc. of DECUS*, Zurich, p.87
- Grosbøl, P., Ponz, D. , 1985 : 'The MIDAS Table File System', *Mem.S.A.It.* **56**, p.429

- Banse, K., Grosbøl, P., Ponz, D., Ounnas, C., Warmels, R., 'The MIDAS Image Processing System in Instrumentation for Ground Based Astronomy: Present and Future, L.B. Robinson, ed., New York, Springer Verlag, p.431

For general bibliographic reference to the MIDAS system (VAX/VMS version), the first reference in the above list should be used.

Detailed technical information of software interfaces and designs used in MIDAS is given in the following documentation:

- MIDAS Environment
- MIDAS IDI–routines
- AGL Reference Manual

Users who want to write their own application programs for MIDAS should read the MIDAS Environment document which gives the relevant information and examples.

For users who have to work with both the IHAP and MIDAS systems a cross–reference document has been made for the most commonly used commands:

- MIDAS-IHAP/IHAP-MIDAS Cross-Reference

The above documents can be obtained by contacting the User Support Group.

Chapter 2

Computational Methods

Astronomical image processing applies a large variety of numerical methods to extract scientific results from observed data. The standard computational techniques used in this process are discussed with special emphasis on the problems and advantages associated with them when applied to astronomical data. It has not been the aim to give a full mathematical description of the methods; references to more detailed explanation of different techniques are given for further study. A general discussion of digital image processing can be found in e.g. Pratt (1978), Bijaoui (1981), and Rosenfeld and Kak (1982) while Bevington (1969) gives a good introduction to basic numerical methods.

The methods are presented in the order in which they are applied in a typical reduction sequence. A number of standard techniques are used at different stages of the reductions in which case they are treated only at the most relevant place. The general reduction sequence has been divided into three main parts. In Section 2.2 the transformation of raw observed data into intensity calibrated values is discussed while general image processing techniques are reviewed in Section 2.3. The evaluation of the resulting frames and the extraction of information from them are considered in Section 2.4 whereas Section 2.5 deals with the statistical analysis of the results. The terminology in this paper has been based of two dimensional image data although most of the techniques can equally well be applied to one dimensional data. Techniques which relate to special detectors or observational methods (e.g. speckle or radio observations) have not been considered.

2.1 Basic Concepts

It is important to understand the basic properties of the data which are being reduced since most computational techniques make implicit assumptions. The result of an algorithm may depend on things such as sampling and noise characteristics of the data set. Furthermore, the most common methods for estimation of parameters are discussed in this section.

2.1.1 Image sampling

The acquisition of a data frame involves a spatial sampling and digitalization of the continuous image formed in the focus plane of a telescope. The image may be recorded analog (e.g. on photographic plates) for later measurements or acquired directly when digital detectors such as diode arrays and CCD's are used. The individual pixel values are obtained by convolving the continuous image $I(x, y)$ with the pixel response function $R(x, y)$. With a sampling step of Δx and Δy the digital frame is given by

$$F_{i,j} = \int I(x, y)R(x - i\Delta x, y - j\Delta y)\; dxdy + N_{i,j} \tag{2.1}$$

where N is the acquisition noise. This convolution is done analog in most detectors except for imaging photon counting systems where it partly is performed digitally. The sampling step and response function are determined normally by the physical properties of the detector and the acquisition setup. The variation of the response function may be very sharp as for most semi–conductor detectors or more smooth as in image dissector tubes. If the original image I is band width limited (i.e. only contains features with spatial frequencies less than a cutoff value ω_c) all information is retained in the digitized frame when the sampling frequency $\omega_s = 2\pi/\Delta x$ satisfies the Nyquist criterion:

$$\omega_s = 2\,\omega_c. \tag{2.2}$$

In Equation 2.2 it is assumed that R is a Dirac delta function. This means that only features which are larger than $2\Delta x$ can be resolved. A frame is oversampled when $\omega_s > 2\omega_c$ while for smaller sample rates it is undersampled.

In astronomy the band width of an image is determined by the point spread function (PSF) and has often no sharp cutoff frequency. Many modern detector systems are designed to have a sampling step only a few times smaller than the typical full width half maximum (FWHM) of seeing disk or PSF. Therefore they will not fully satisfy Equation 2.2 and tend to be undersampled especially in good seeing conditions.

A typical assumption in image processing algorithms is that the pixel response function R can be approximated by a Dirac delta function. This is reasonable when the image intensity does not vary significantly over R as for well oversampled frames where the effective size of R is roughly equal to the sample step. If it is not the case, the effects on the algorithm used should be checked. Interpolation of values between existing pixels is often necessary e.g. for rebinning. Depending on the shape of R and band width of the image different schemes may be chosen to give the best reproduction of the original intensity distribution. In many cases low order polynomial functions are used (e.g. zero or first order) while sinc, spline or gaussian weighted interpolation may be more appropriate for some applications.

2.1.2 Noise distributions

Besides gross errors which are discussed in Section 2.2.1 the two main sources of noise in a frame come from the detector system N and from photon shot–noise of the image intensity I (see Equation 2.1). It is assumed that the digitalization is done with sufficiently high resolution to resolve the noise. If not, the quantization of output values gives raise to additional noise and errors.

A large number of independent noise sources from different electronics components normally contributes to the system noise of a detector. Using the central limit theorem, the total noise can be approximated by a Gaussian or normal distribution which has the frequency function :

$$P_G(x;\mu,\sigma) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{1}{2}\left[\frac{x-\mu}{\sigma}\right]^2\right) \tag{2.3}$$

where μ and σ are mean and standard deviation, respectively. The photon noise of a source is Poisson distributed with the probability density P_P for a given number of photons n :

$$P_P(n;\mu) = \frac{\mu^n}{n!}e^{-\mu} \tag{2.4}$$

where μ is the mean intensity of the source. It can be approximated with a Gaussian distribution when μ becomes large. For photon counting devices the number of events is normally so small that Equation 2.4 must be used while Gaussian approximation often can be used for integrating systems (e.g. CCD's).

In the statistical analysis of the probability distribution of data several estimators based on moments are used. The r^{th} moment m_r about the mean $\bar{x}$ and its dimensional form α_r are defined as

$$m_r = \frac{1}{N}\sum_{i=1}^{N}(x_i - \bar{x}) \quad , \quad \alpha_r = \frac{m_r}{\sqrt{m_2^r}}. \tag{2.5}$$

The second moment is the variance while first is always zero. The general shape of a distribution is characterized by the *skewness* which denotes its asymmetry (i.e. its third moment α_3) and the *kurtosis* showing how peaked it is (i.e. its fourth moment α_4). For a normal distribution, these moments are $\alpha_3 = 0$ and $\alpha_4 = 3$ while for a Poisson distribution are $\alpha_3 = 1/\sqrt{\mu}$ and $\alpha_4 = 3 + 1/\mu$. Besides these moments other estimators are used to describe a distribution e.g. *median* and *mode*. The *median* of a distribution is defined as the value which has equally many values above and below it while a *mode* is the local maximum of the probability density function.

2.1.3 Estimation

A number of different statistical methods are used for estimating parameters from a data set. The most commonly used one is the least squares method which estimates a parameter θ by minimizing the function :

$$S(\theta) = \sum_i (y_i - f(\theta; x_i))^2 \tag{2.6}$$

where y is the dependent and x the independent variables while f is a given function. Equation 2.6 can be expanded to more parameters if needed. For linear functions f an analytic solution can be derived whereas an iteration scheme must be applied for most non–linear cases. Several conditions must be fulfilled for the method to give a reliable estimate of θ. The most important assumptions are that the errors in the dependent variable are normal distributed, the variance is homogeneous, and the independent variables have no errors and are uncorrelated.

The other main technique for parameter estimation is the maximum likelihood method where the joint probability of the parameter θ :

$$l(\theta) = \prod_i P(\theta, x_i) \tag{2.7}$$

is maximized. In Equation 2.7, P denotes the probability density of the individual data sets. Normally, the logarithm likelihood $L = \log(l)$ is used to simplify the maximization procedure. This method can be used for any given distribution. For a normal distribution the two methods will give the same result.

2.2 Raw to Calibrated Data

The actual transformation of raw detector data to clean maps in intensity scale depends strongly on both the imaging and detecting systems. However, three typical steps can be identified, namely: detection and removal artifacts in the data, correction for non–linear effects or relative variations in sensitivity of the detector system, and correction for geometric distortions in the imaging device. Although the order of the first two steps can often be interchanged the geometric correction must be performed last because it involves a rebinning of the data which assumes them to be intensities.

2.2.1 Artifacts

Raw data from detector systems often contains artifacts originating from elements which have abnormal properties. Photographic emulsions and photocathodes can have dust or scratches while digital detectors (e.g. CCD and photodiode arrays) are affected by defects in the manufacturing process. Besides these imperfections in the detectors also cosmic ray events and electric disturbances can corrupt parts of the data. It is important to locate these gross errors to avoid that they degrade the correct data during the further reductions. Such bad pixels are either replaced by a local estimate or flagged as non–valid. Although the latter option is the most correct not all image processing systems are fully supporting the use of non–defined values (mostly due to programming and computer overheads).

Depending on the available data different methods are applied to detect and correct gross errors in the data. When only one frame is available artifacts are identified by their appearance; they are normally very sharp features. Most filter techniques assume that the image is oversampled so that the values in any region of a given small size can be regarded as taken from a random distribution. If the image is undersampled (i.e. the point spread functions is unresolved) it is impossible to distinguish between real objects and gross errors.

For a well sampled frame $f_{i,j}$ non–linear digital filters are used giving the resulting frame $r_{i,j}$:

$$r_{i,j} = \begin{cases} \mathcal{E}_{i,j}(f_{i+m,j+n}) & ,\text{if } L < |f_{i,j} - \mathcal{E}_{i,j}| \\ f_{i,j} & ,\text{if } L \geq |f_{i,j} - \mathcal{E}_{i,j}|, \end{cases} \tag{2.8}$$

where $\mathcal{E}_{i,j}$ is a local estimate for $f_{i,j}$. The modification level L may vary over the frame but is normally set to 5–10 times the dispersion σ of the noise, to avoid modifying its distribution. The local estimate $\mathcal{E}_{i,j}$ may or may not include the original value $f_{i,j}$. The latter is an advantage if most faults only have a size of one pixel. The simplest estimator is the arithmetic mean. The main problem is that it depends linearly on the data values of the bad pixels. If a few pixels with very large errors are located in the region used for the estimate it may be effected so much that normal pixels are modified. By applying Equation 2.8 with a mean estimator iteratively, it is possible to reduce the dependency on gross errors. This procedure is called $\kappa\sigma$–clipping and was investigated by Newell (1979).

To avoid this problem more stable estimators are preferred such as the mode or median. Since the mode may neither exist nor be uniquely defined, the median is normally used (Tukey, 1971). The median filter can only detect artifacts if they occupy less than half of the filter size. Therefore, its size must be larger than two times the largest defect which should be removed and smaller than the smallest object to be preserved.

Another group of non–linear filters is based on recursive filters which uses the already filtered values for the estimator $\mathcal{E}$. In the one dimensional case a frame f_i is transformed to :

$$r_i = \begin{cases} \mathcal{E}_i(r_{i-1}, r_{i-2}, \cdots, r_{i-n}) & ,\text{if } L < |f_i - \mathcal{E}_i| \\ f_i & ,\text{if } L \geq |f_i - \mathcal{E}_i| \end{cases} \tag{2.9}$$

where $r_i = f_i$ is assumed for $i = 1, 2, \cdots, n$. The estimator can either be a linear expression (e.g. average or a low order extrapolation) or be based on the median as above. Due to the numeric feedback these filters are intrinsicly more unstable, however, by including a limit L which depends on f_i a useful filter can be constructed (Grosbøl, 1980). The main advantage of this filter type, compared to the median filter, is its capability to remove artifacts larger than its own size. Figure 2.1 shows a CCD dark current exposure with cosmic ray events. It can be seen that all artifacts can be removed using either a large median filter or a recursive filter while small median filters are unable to remove the larger events. When real features are present such as spectra in Figure 2.2 the non–linear filters may modify spectral lines.

When more than two images of the same region are available, it is possible to compare the stack of pixels from the different exposures. The frames must be aligned and intensity calibrated before a comparison can be performed. Artifacts become more difficult to detect if an alignment, hence rebinning, is needed due to its smoothing effect. Thus, the stacking technique is best suited for removing cosmic ray events and electronic disturbances. Statistical weights must also be assigned to the individual images depending on exposure and signal–to–noise ratio. Outliers in the stack of pixel values are rejected either by comparing them to the median or by applying $\kappa\sigma$–clipping techniques (Goad, 1980). The resulting frame is then the mean of the remaining values. A set of CCD images of the galaxy A0526–16 are shown in Figure 2.3 including the resulting stacked image. By having different origins of the galaxy in the exposures the chip artifacts could also be removed. For comparison with non–linear filter techniques, Figure 2.2D shows removal of cosmic ray events from the spectral frame discussed above.

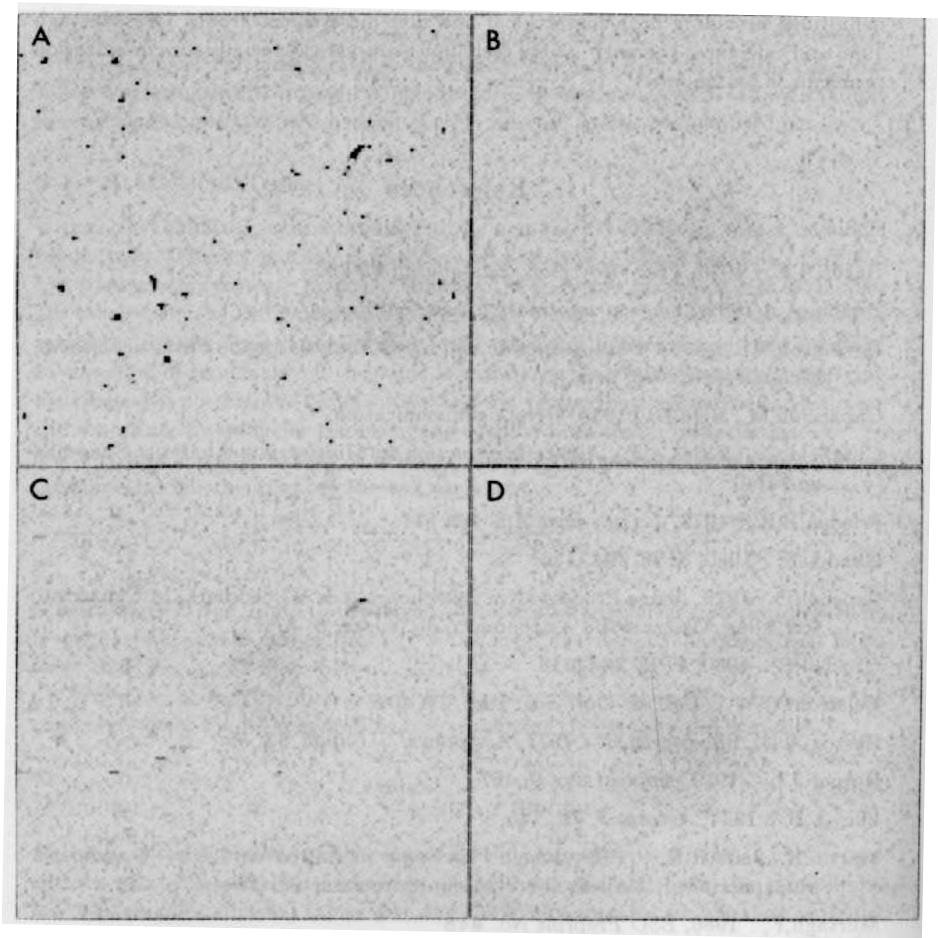

Figure 2.1: A dark current CCD exposure with cosmic ray events which are removed with non–linear filters. (A) original, (B) 5*5 median filter, (C) 5*1 median filter, and (D) 5*1 recursive filter.

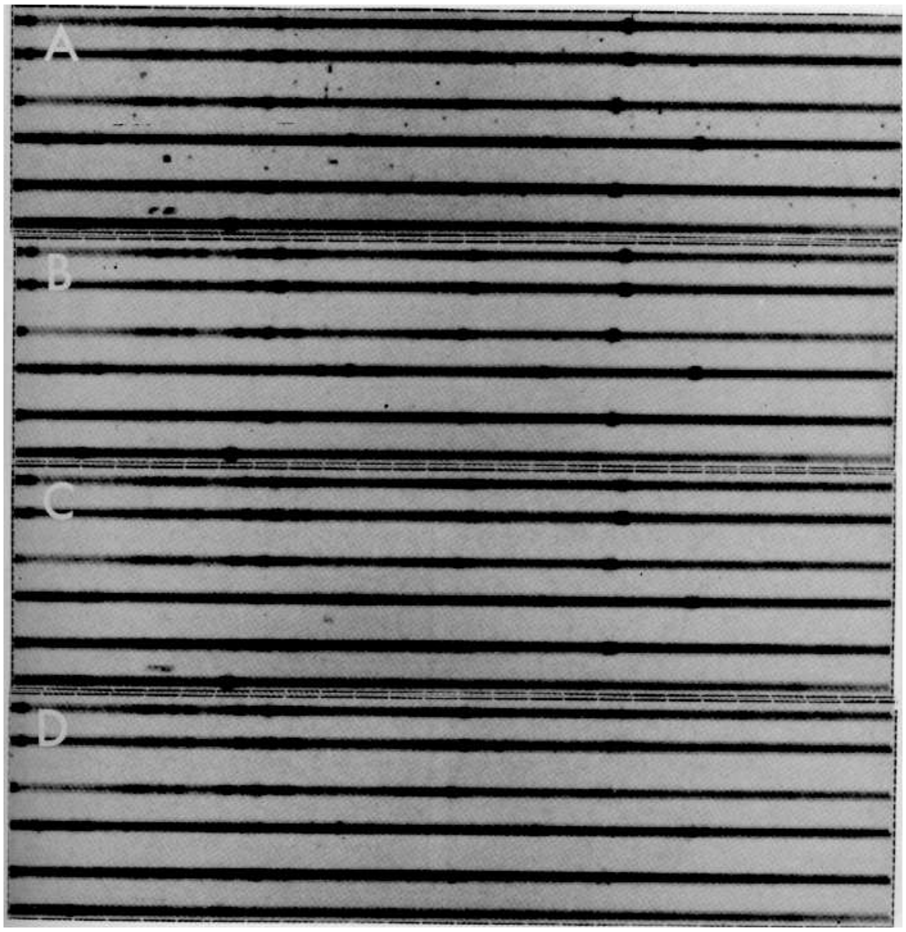

Figure 2.2: Removal of cosmic ray events on a CCD spectral exposure with different techniques: (A) original, (B) 5×1 median filter, (C) 5×1 recursive filter and (D) stack comparison.

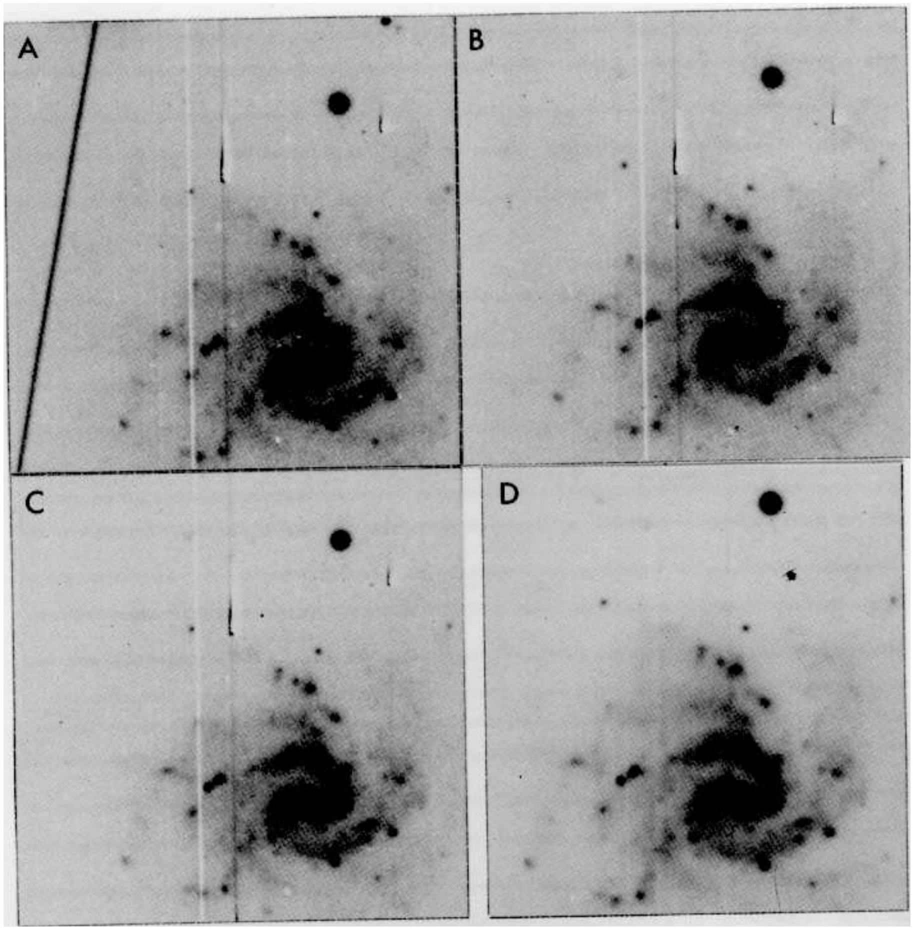

Figure 2.3: Removal of artifacts on CCD exposures (A,B,C) of the galaxy A0526–16 by stacking the frames yielding the combined image (D).

2.2.2 Response Calibration

The raw data values from the detector system have to be transformed into relative intensities which then can later be adjusted to an absolute scale by comparison with standard objects. The majority of modern detectors (e.g. CCD, diode–array or image tube) have a nearly linear response while photographic emulsions are strongly non–linear. Even for the 'linear' detectors, a number of corrections must be included in the intensity calibration. Some of these can be derived theoretically such as dead–time corrections for photon counting devices or saturation effects for electronographic emulsions while other non–linear effects are determined empirically. Systems which are assumed to be linear need only be corrected for a possible dark count and bias in addition to the relative sensitivity variation over the detector. The correction frames are determined from a set of test exposures from which artifacts are removed first as described in Section 2.2.1. A raw frame $C_{i,j}$ is then transformed to relative intensities $I_{i,j}$ by

$$I_{i,j} = \frac{C_{i,j} - D^c_{i,j}}{F_{i,j} - D^f_{i,j}} \tag{2.10}$$

where $D_{i,j}$ is the appropriate dark counts including bias and $F_{i,j}$ is a normalized flat field exposure.

A mathematical function is used to transform data from detectors with non–linear response to a more linear domain. For photographic emulsions Baker (1925) found the formula

$$\mathcal{D} = \log(10^D - 1) \tag{2.11}$$

which makes the lower part of the characteristic curve almost linear. In Equation 2.11, D is the photographic density above fog. These values can then, by means of least squares methods, be fitted with a power series

$$\log(I_{i,j}) = T(\mathcal{D}_{i,j}) = \sum_{k=0}^{n} a_k \mathcal{D}^k_{i,j} \tag{2.12}$$

where n for most applications is smaller than 7. The characteristic curves are shown in Figure 2.4 both using normal and Backer densities. The coefficients a_k depend not only on the emulsion type but also on the spectral range. For spectral plates this leads to a positional variation of the terms a_k.

The main problem with non–linear detectors is not so much to determine the response curve as the modification of the noise distribution. Thus, the gaussian grain noise on emulsions becomes skewed through the intensity calibration. Special care must be taken in the further processing to avoid systematic error due to non–gaussian noise (e.g. the average of a region will be biased). One possible way to make the distribution more gaussian again is to apply a median filter because it is less affected by the transformation.

2.2.3 Geometric Corrections

Most imaging systems contain intrinsic geometric distortions. Although they can often be disregarded for small field corrections they must be applied when image tubes or dispersive

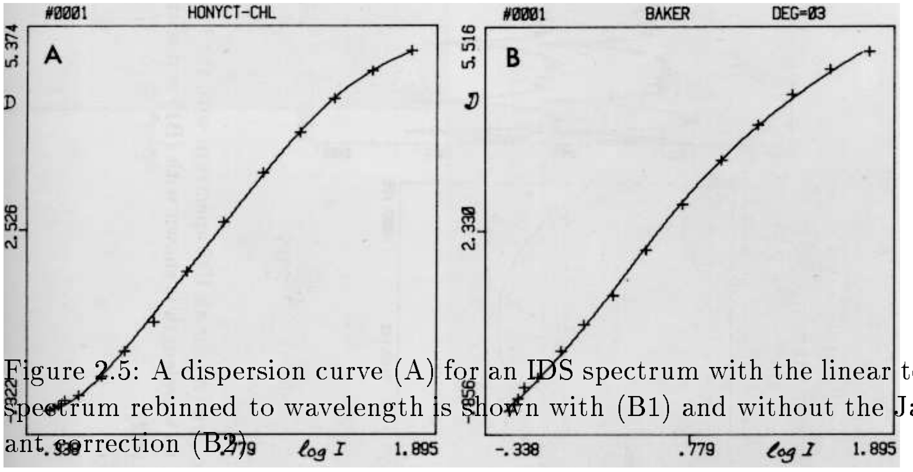

Figure 2.5: A dispersion curve (A) for an IDS spectrum with the linear term omitted. The spectrum rebinned to wavelength is shown with (B1) and without the Jacobian determinant correction (B2).

elements (e.g. in spectrographs) are used. The actual form of the distortions is determined by observing a known grid of points or spectral lines. Normally, a power series is fitted to the point giving the coordinate transformation

$$x = X(u,v) \;=\; \sum_{i=0}^{n}\sum_{j=0}^{m} a_{i,j}(u-u_0)^i(v-v_0)^j \tag{2.13}$$

$$y = Y(u,v) \;=\; \sum_{i=0}^{n}\sum_{j=0}^{m} b_{i,j}(u-u_0)^i(v-v_0)^j. \tag{2.14}$$

where (u_0, u_0) is an arbitrary reference point. The area of a pixel is changed by this transformation with an amount

$$dA = dx\,dy = |\mathbf{J}|\,du\,dv = \left|\frac{\partial(x,y)}{\partial(u,v)}\right| du\,dv = \left|\frac{\partial x}{\partial u}\frac{\partial y}{\partial v} - \frac{\partial x}{\partial v}\frac{\partial y}{\partial u}\right| du\,dv \tag{2.15}$$

where $\mathbf{J}$ is the Jacobian determinant. The intensity values in the transformed frame must be corrected by this function so that the flux is maintained both locally and globally. A wavelength transformation for an image tube spectrum is shown in Figure 2.5 where both resulting spectra with and without flux correction are given.

Although this is mathematically very simple, it gives significant numeric problems due to the finite size of pixels. The main problem is that one has to assume a certain distribution of flux inside a pixel e.g. constant. This assumption may affect the detailed local flux conservation and introduce high frequence error in the result. A further problem is the potential change of the noise distribution due to the interpolation scheme used. This can be solved be careful assignment of weight factors or by simply reducing the high frequence noise in the original frame by smoothing.

2.3 Image Manipulations

After the raw image data are calibrated into relative intensive values several operations may be applied to frames to enhance features which later should be studied. This involves manipulations with the Signal–to–Noise ratio (S/N), resolution, and coordinates of the images. Further, real but disturbing features may be removed to allow a better access to the objects of interest. The typical methods include digital filtering, coordinate transformations, and image restoration.

2.3.1 Digital Filters

The analog detector output is normally converted into integer values so that the internal detector noise is resolved. This noise can be reduced by replacing the pixels with an average of the surrounding values using a linear filter. In general, the image $I(m,n)$ is convolved by a filter function $F(i,j)$ giving the smooth image

$$S(m,n) = \sum_{i=-k}^{k} \sum_{j=-l}^{l} I(m-i, n-j) F(i,j). \tag{2.16}$$

In principle, the image S is smaller than the original because the convolution is undefined along the edge. For convenience, extrapolated values for F are used to avoid this reduction in size after each filtering. Several different filter functions are used depending on the application. Two typical 3×3 filters are given as examples, namely peaked smoothing filter

$$F_{smooth} = \frac{1}{16} \begin{pmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{pmatrix} \tag{2.17}$$

and a constant 'block' filter

$$F_{block} = \frac{1}{9} \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix}. \tag{2.18}$$

Both filters are normalized to unity in order to preserve the total flux in the image. Depending on the actual size and shape of the filter, different degrees of smoothing are achieved (i.e. larger size gives stronger smoothing). Filter functions which vary significantly or are non–zero at the edge (e.g. Equation 2.18) will tend to preserve some high frequencies in the output. The effects of applying linear filters to a CCD frame is shown in Figure 2.6. It can be seen that the 'block' filter leaves sharper features in the result frame than the 'smooth' filter. This is avoided by taking a smooth function like a gaussian giving

$$F_{gaus}(j,k) = A \exp\left(-\frac{1}{2}\left[\frac{j^2}{\sigma_x^2} + \frac{k^2}{\sigma_y^2}\right]\right) \tag{2.19}$$

where A is a normalization constant and σ defines the width of the filter. The parameters of the gaussian filter can normally be varied to satisfy most applications (see Equation 2.19).

The increase in the S/N is paid by a degradation in the resolution. When a fixed filter is used this blurring affects the whole frame; although a smoothing may be required only

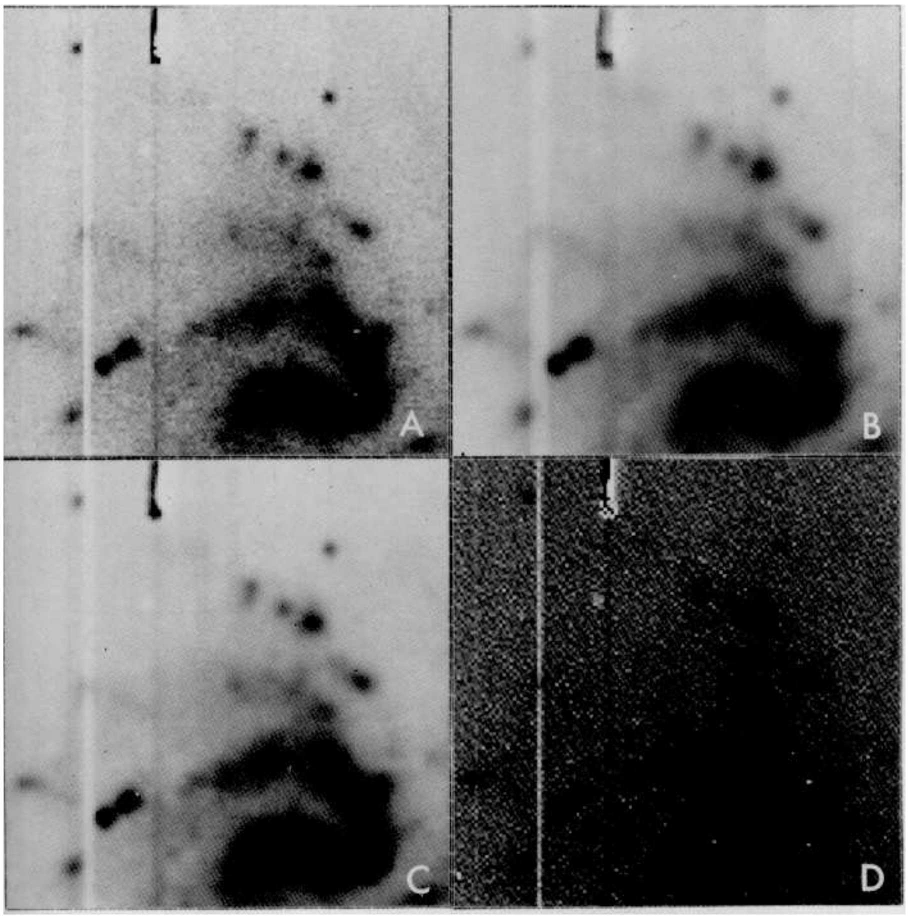

Figure 2.6: Different digital filters applied on a CCD image: (A) original, (B) block filter, (C) smooth filter, and (D) Laplacian filter.

in the low S/N regions. This problem can be avoided by applying a gaussian filter for which the width σ is a function of the local S/N (e.g. $\sigma \propto N/S$).

Other types of linear filters are used to emphasize edges and variations. They are based on differential operators like the Laplacian and have often $\sum\sum F = 0$ to remove the mean level of the input frame. A symmetric edge detection filter may be defined as

$$F_{Laplace} = \begin{pmatrix} 1 & -2 & 1 \\ -2 & 4 & -2 \\ 1 & -2 & 1 \end{pmatrix} \tag{2.20}$$

while a large variety of other filters may be constructed to enhance special features. The result of the $F_{Laplace}$ filter is shown in Figure 2.6.

In some cases types of objects (e.g. stars) may disturb the further analysis of an image. If these objects have a special appearance if is often possible to remove them with a non–linear filter (see Section 2.2.1). To remove stellar images from a picture of comet Halley, the results of applying different non–linear filters to the frame are given in Figure 2.7. For more complicated features they are deleted from the image by interpolation between pixels in nearby regions.

2.3.2 Background Estimates

In most astronomical observations the image intensity consists of contributions from both object and sky background. Depending on the complexity of the field and the type of object, different methods are used to estimate and subtract the background intensity. For linear detectors this can be done directly on the intensity calibrated frame while special consideration must be given when a non–linear response transformation is used (i.e. for photographic emulsions) due to the non–gaussian noise distribution. In the latter case a fit is normally done to the original data and the fitted values then transformed to intensities and subtracted.

An accurate determination of the background is extremely important for the latter analysis. Therefore, one prefers to use all pixels, which are not contaminated by sources, and fit a low order polynomial surface to the background. Non–linear filters are often used to remove stellar images and other sharp features (see Section 2.2.1) while extended objects are very difficult to eliminate automatically. If only point like objects are analyzed background following methods like the recursive filter described by Martin and Lutz (1979) can be used.

Also $\kappa\sigma$–clipping techniques are applied in an iterative scheme where pixels with high residual compared to a low order polynomial fit to the frame are rejected (Capaccioli and Held 1979). In this method areas containing extended objects can be excluded before the iteration starts. In Figure 2.8, a field with extended objects is shown with the mask defining the areas to be omitted in the computations.

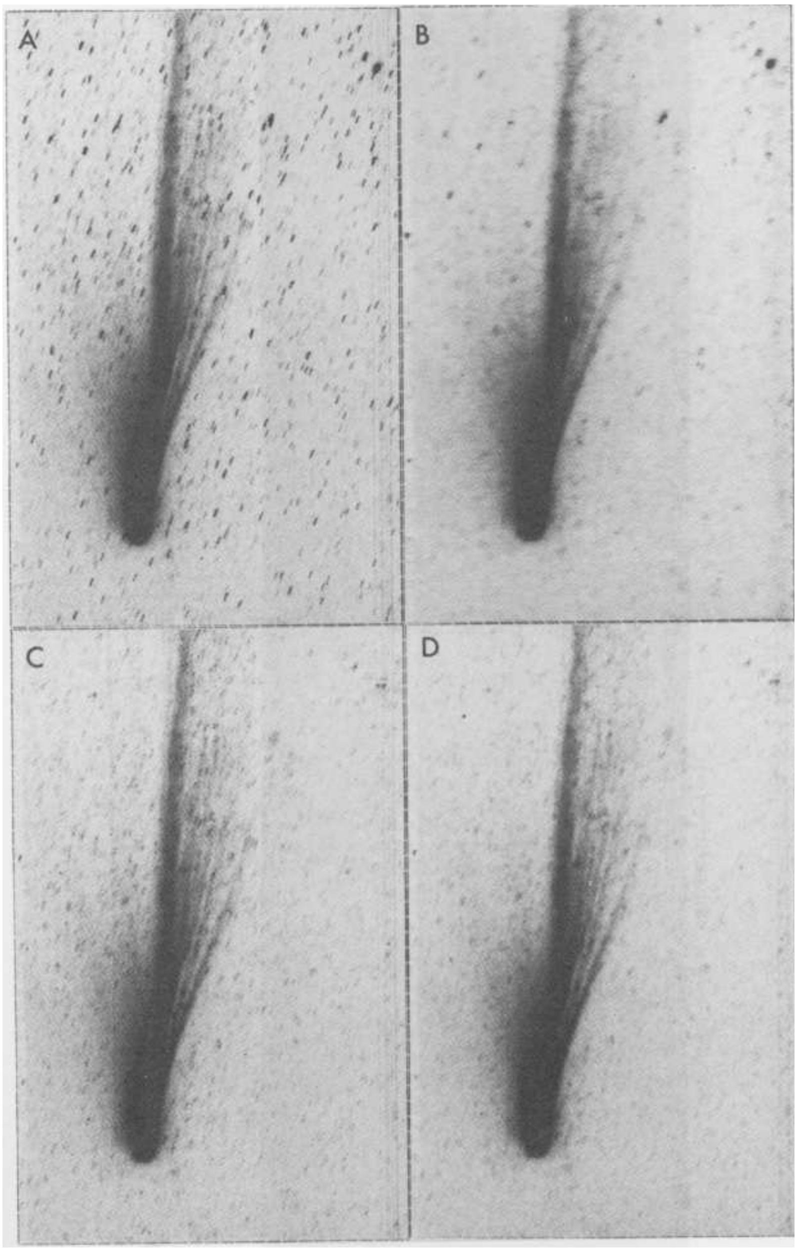

Figure 2.7: Removal of stars from a CCD frame of Comet Halley: (A) original, (B) 5×5 median filter, (C) 5×1 recursive filter, and (D) both recursive 5×1 filter and a 1×3 median filter.

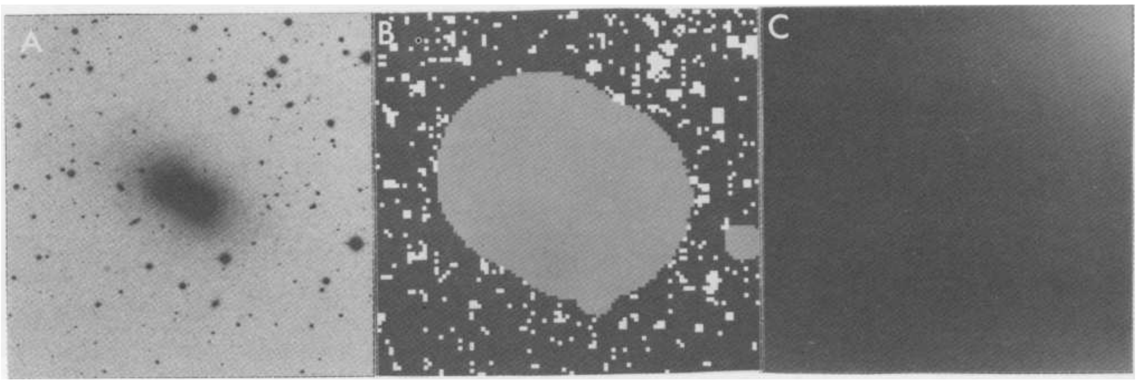

Figure 2.8: Background fitting with an iterative $\kappa\sigma$ technique: (A) original, (B) mask of included areas, and (C) fitted background.

2.3.3 Transformations

Depending on the further reductions the data may be transformed into the coordinate system which is most relevant for the physical interpretation. This will typically be used when certain characteristics of the data will appear as a linear relation in the new coordinates. These transformations involve non–linear rebinning as discussed in Section 2.2.3. To conserve flux in the new system, the pixel values must be corrected by the Jacobian determinant $\mathbf{J}$ in Equation 2.15.

For spectra a transformation from wavelength to frequency is used to identify spectral regions which follow a power law. This transformation is given by

$$\nu = c/\lambda, \quad \mathbf{J} = -c/\lambda^2 \tag{2.21}$$

where ν is the frequency and λ is wavelength. The intensities I are translated to a logarithmic scale (e.g. magnitudes $\mathcal{M} = -2.5\log(I)$) so that a power law spectrum appears linear.

In the classification of galaxies, ellipticals can be distinguished on their radial surface brightness profile which can be approximated by $\log(I) \propto r^{1/4}$. This gives the transformation formula

$$x = r^{1/4}, \quad \mathbf{J} = r^{-3/4}. \tag{2.22}$$

Since the intensity profile only should be resampled, the flux correction is not applied in this case. A logarithmic scale is also used here to achieve a linear relation. An example is given in Figure 2.9.

Whereas the transforms mentioned above only perform a non–linear rebinning of the data, the Fourier transform converts a spatial image into the frequency domain. This transform has two main applications namely analysis of periodic phenomena and convolution due to its special properties. The transform can be given as

$$\mathcal{F}(u,v) = \frac{1}{N}\sum_{j=0}^{N-1}\sum_{k=0}^{N-1} F(j,k)\exp\left[\frac{-2\pi i}{N}(uj+vk)\right] \tag{2.23}$$

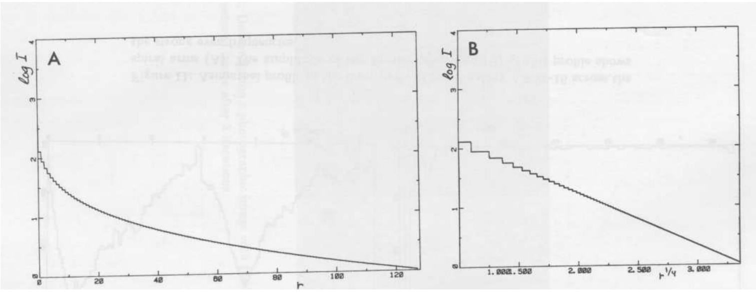

Figure 2.9: The radial profile of an elliptical galaxy shown with linear steps (A) and rebinned to $r^{1/4}$ increments (B).

where (u, v) are the spatial frequencies and i denotes the imaginary part. The corresponding inverse transform is then

$$F(j, k) = \frac{1}{N} \sum_{j=0}^{N-1} \sum_{k=0}^{N-1} \mathcal{F}(u, v) \exp\left[\frac{2\pi i}{N}(uj + vk)\right]. \tag{2.24}$$

Special numeric techniques, called Fast Fourier Transforms or FFT, can significantly decrease the time needed to compute these transforms. They are optimized for images with a size equal to a power of 2 (see e.g. Hunt 1969) but can also be used in other cases.

To analysis the periodic occurrence of features in time series, spectra, or images the amplitude of the Fourier transform or the power spectrum is used. The power spectrum $\mathcal{W}$ of the function F is given by

$$\mathcal{W}(u, v) = |\mathcal{F}(u, v)|^2. \tag{2.25}$$

Peaks in $\mathcal{W}$ indicate the presence of specific frequencies while the continuum originates from a combination of objects and noise. Since the Fourier transform assumes the image to occur periodically, discontinuities at the edges of the image will produce artificial contributions to the power spectrum. Thus, care should be taken to remove systematic background variations to avoid this happening. As an example of using Fourier transforms to extract information from a frame, an azimuthal intensity profile of the spiral galaxy A0526–16 is shown in Figure 2.10. The radius was chosen so that the profile intersects the spiral pattern in the inner parts of the galaxy. In the amplitude plot of the Fourier transform, it can be seen that the spiral pattern mainly contains even frequency components.

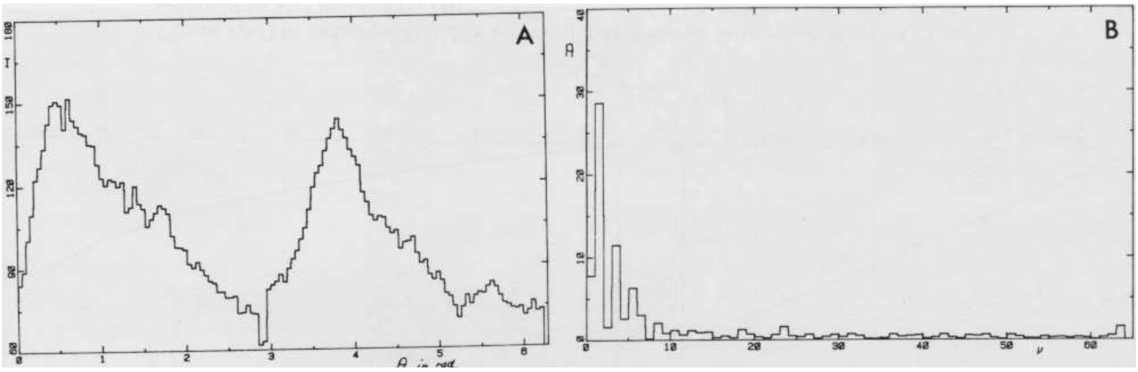

Figure 2.10: Azimuthal profile in the inner parts of a spiral galaxy A0526–16 across the spiral arms (A). The amplitude of the Fourier transform (B) of this profile shows the strong even frequencies.

The 4^{th} and 6^{th} harmonic indicates that the wave is asymmetric due to non–linear effects in the spiral density wave.

It is possible to use the transformation for convolutions because this operation corresponds to a multiplication in the frequency domain :

$$\mathcal{O}_{\mathcal{F}}[F(j,k) \otimes H(j,k)] = \mathcal{F}(u,v)\,\mathcal{H}(u,v) \tag{2.26}$$

where $\mathcal{O}_{\mathcal{F}}$ and $\otimes$ denote the Fourier transform and convolution operators, respectively. Especially when the convolution matrix is large it is more efficient to perform the operation in frequency than in spatial domain.

2.3.4 Image Restoration

Both the imaging system and observing conditions will cause a degradation of the resolution of the image. In principle it is possible to reduce this smearing effect by deconvolving the frame with the point spread function. The degree to which this can be done depends on the actual sampling of the image. Basically, it is not possible to retrieve information on features with frequencies higher than the Nyquist frequency (see Equation 2.2). Several different techniques are used depending on the data (see Wells 1980 for a general discussion of the methods).

Fourier transforms are often used since convolutions in the frequency space become multiplications (see Section 2.3.3). Combining Equation 2.1 and Equation 2.26 the original image $I = \mathcal{O}_{\mathcal{F}}(\mathcal{I})$ is obtained by division

$$\mathcal{I}(u,v) = \mathcal{O}_{\mathcal{F}}[F(j,k) \otimes P(j,k) + N(j,k)]/\mathcal{P}(u,v) \tag{2.27}$$

if the transformed PSF $\mathcal{P}$ is non–zero and the noise N is insignificant. For data with low or medium signal–to–noise ratio (i.e. $S/N < 100$), as for most optical observations, this technique introduces artifacts in the deconvolved image. These effects can be reduced by filtering the transforms with Wiener filters (Helstrom 1967, Horner 1970).

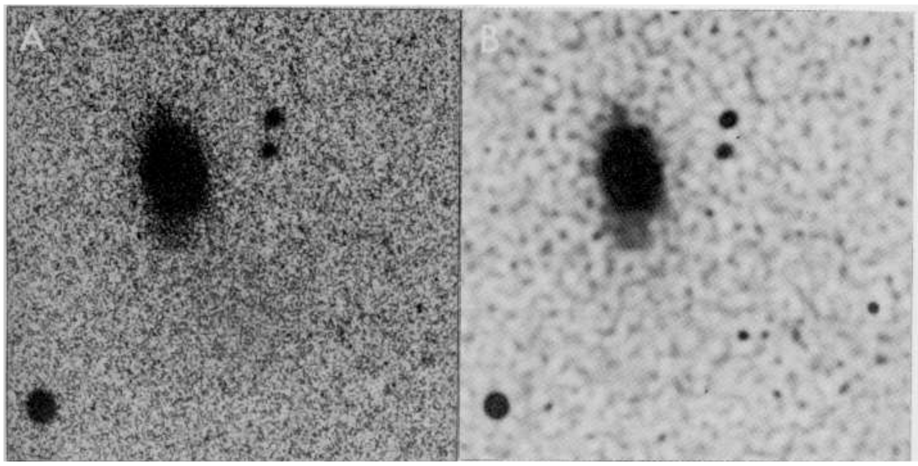

Figure 2.11: Deconvolution a photographic image with the Lucy method: (A) original and (B) restored image after 3 iterations.

Another group of image restoration algorithms use iterative methods to obtain a solution which is consistent with the data. The maximum entropy method uses either the entropy :

$$H1 = -\sum_j I_j \log(I_j) \quad or \quad H2 = \sum_j \log(I_j) \tag{2.28}$$

in the optimizing procedure (Frieden 1972). It tends to enhance sharp features but a solution may depend on the initial guess and therefore not be unique.

A different scheme was introduced by Lucy (1974) who uses a correction term based on the ratio between the image and the guess. A first guess Ψ^0 must be specified (e.g. a constant) to start the iteration. The first step in the iteration performs a convolution with the PSF :

$$\varphi^r_{i,j} = \Psi^r_{i,j} \otimes P_{i,j}. \tag{2.29}$$

The second step computes a correction factor based on this frame and the original image $\tilde{\varphi}$:

$$\Psi^{r+1}_{i,j} = \Psi^r_{i,j} \left(\frac{\tilde{\varphi}_{i,j}}{\varphi^r_{i,j}} \otimes P_{i,j} \right). \tag{2.30}$$

The procedure repeats these two steps until the corrections are sufficiently small. After Equation 2.30 is computed the iteration continues with Equation 2.29. This scheme reaches a stable solution very quickly (i.e. 3–5 steps) and is little affected by noise. This makes it very useful for low signal–to–noise data. A photographic picture of a galaxy is used to illustrate this technique (see Figure 2.11). A fit to the stellar image was used to define the PSF.

2.4 Extraction of Information

The previous sections have described the most frequently used computational techniques for image handling in optical astronomy. After the images have been calibrated and manipulated to give the best representation of the data the astronomical information has to be extracted. For frames which contain a large number of objects automatic methods are required to locate them. A set of parameters is then estimated for the individual objects (e.g. position, total intensity, or velocity) depending on type and observational technique.

2.4.1 Search Algorithms

For projects which perform statistical analysis on groups of objects, it is important to rely on an objective search method to extract them from the data frames. For this reason plus the need to search large areas efficiently it is necessary to use automatic algorithms for this task. Several different methods are applied for this purpose depending on the demands for speed, limiting magnitude, and location of special objects. They fall in four main categories depending on their detection criterion, namely : level, gradient, peak, and template match detection.

The simplest and fastest method is using a given level over a previously determined background value as the criterion to identify possible sources. All pixels with intensities over this value are flagged and later grouped together to form objects (Pratt 1977). The background estimation can be avoided by using a gradient of the intensity distribution (Grosbøl 1979) instead. If the background variation over small areas can be regarded as linear, a Laplacian filter (see Equation 2.20) will locate only sharp features such as edges of stars. Since the derivative has a larger noise than the original image, this method will be slightly less sensitive than using the level. It can be applied directly to data without first having to compute the background and may therefore be faster to use if only point sources should be detected. The peak detection method finds pixels which are higher than their surroundings and is also based on a derivative (Newell and O'Neil 1977; Herzog and Illingworth 1977). Especially in crowded fields where a background is difficult to determine and where objects may overlap, it is a better search criterion than the two previous schemes. Finally, it is possible to compare each position with a template (e.g. the PSF) and thereby determine the probability of having an object there. Although this gives the most sensitive search criterion because it uses all information, it requires much larger amounts of computer time than the other methods.

2.4.2 Fitting of data

The final step in data reduction is the extraction of astrophysical parameters from the data. This is often done by fitting a parameterized model of the objects to the data by means of least squares and maximum likelihood methods (see Section 2.1.3). The correct weighting of data is important in order to use all information in the image and minimize the effects of noise on the final parameters. For stellar images or line profiles, either analytical functions (e.g. weighted Lorentzian–Gaussian profiles) or empirical models of

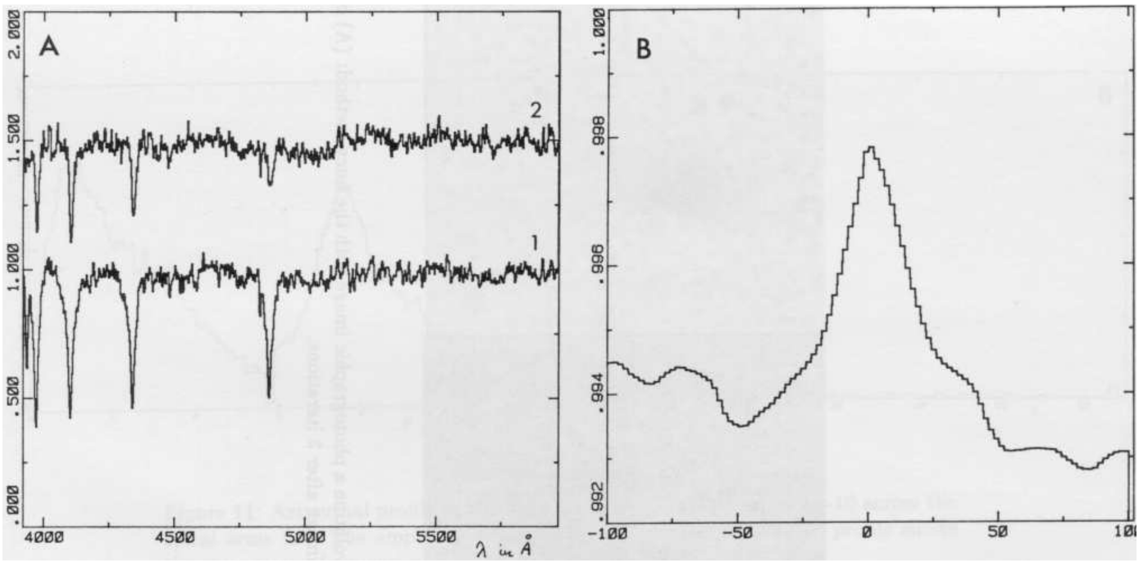

Figure 2.12: Two normalized early type spectra used as template (A1) and object (A2) yield the cross–correlation function (B).

the PSF are used to obtain flux and shape parameters. Very elaborate models may be applied to more complex objects such as galaxies where the flux are decomposed in a set of different components e.g. bulge, disk, bar and spiral.

When a set of objects have similar features and their relative shifts should be determined, the correlation between them and a template object T is analyzed using the cross–correlation function :

$$C(m,n) = \frac{\sum_j \sum_k I(j,k)T(j-m,k-n)}{\sum_j \sum_k I(j,k)^2} \tag{2.31}$$

where I is the object. Since this function is 1 for a perfect match between object and template, the maximum value will indicate how similar the objects are. The location of the main peak gives the translation and is used in spectroscopy to determine radial velocities of stars. This is shown in Figure 2.12 where the normalized spectra of two early type stars are cross–correlated. Since only the spectral lines should be used, it is important to subtract or normalize the continuum to avoid interference from it.

2.5 Analysis of Results

As the last step in the reduction sequence, the scientific analysis of the data is dealing with the statistical comparison between models and the extracted parameters. Although this falls outside image processing it is an important part of the reductions and is included in modern data reduction systems. Due to the large variety of data only the most commonly used techniques are mentioned.

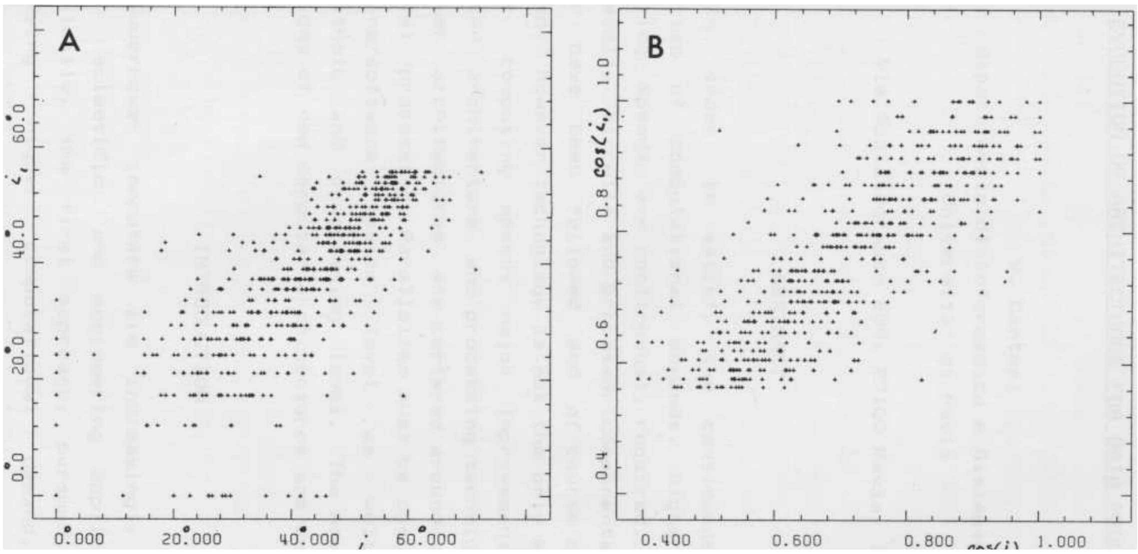

Figure 2.13: Correlation between two measures of the inclination angle of galaxies: (A) with angle i as variable, and (B) with $\cos(i)$.

2.5.1 Regression Analysis

The degree of correlation between two parameters can visually be estimated by plotting them. Numerically this is expressed by the correlation coefficient :

$$\rho = \frac{\sum(x-\bar{x})(y-\bar{y})}{\sqrt{\sum(x-\bar{x})^2\sum(y-\bar{y})^2}} \tag{2.32}$$

for the two variables x and y. This expression assumes that the quantities have normal distribution. When the distribution is unknown, the correlation can be given by the Spearman rank correlation coefficient :

$$r_s = 1 - \frac{6\sum D^2}{n(n^2-1)} \tag{2.33}$$

where n is the number of pairs and D is the difference of the rank between x and y in a pair.

When the correlation coefficient indicates a significant correlation, the functional relation is given by the regression line which is computed by least squares methods for normal distributions or maximum likelihood procedures (see Section 2.1.3). If non–linear relations are expected, the variables are transformed into a linear domain by means of standard mathematical functions. Such transformation may also be used to make the variance of a variable homogenous. In Figure 2.13, a correlation between two different measures of the inclination angle i of galaxies is shown. Since the angle is obtained from the axial ratio, a cosine transformation is used to achieve a homogenous variance. Unfortunately, it is not always possible to obtain both a linear relation and homogeneity of variance with the same transformation.

All methods assume that the single data points are unbiased and uncorrelated. Great care must be taken to assure that this is fulfilled. If this is not the case, significant systematic errors may be introduced in the estimates. A standard example is magnitude limited samples which may be affected by the Malmquist bias.

2.5.2 Statistical Tests

Often, the computed estimators have to be compared with other values or model predictions. Different statistical tests are used to compute the probability of a given hypotheses being correct. A typical null hypotheses is that two quantities or samples are taken from the same population. For single quantities, a confidence interval is estimated for the desired significance level. The null hypotheses is then accepted at this level of significance if the value is within the interval. When the underlying distribution is normal, the "Student's" t and the χ^2 distributions are used to estimate the confidence intervals for the mean and standard deviation, respectively.

It is also possible to test if a set of observed values are taken from a given distribution. For this purpose the test variable

$$\hat{\chi}^2 = \sum \frac{(O-E)^2}{E} \tag{2.34}$$

is used where O and E are the observed and expected frequencies, respectively. The level of significance is derived from $\hat{\chi}^2$ which is χ^2 distributed. The bins must be so large that E is larger than 5 for all intervals.

When the underlying distribution is unknown, two independant samples can be compared using the Kolmogoroff and Smirnoff test. It uses the test variable

$$D = \max \left| \left(\frac{F_1}{n_1} - \frac{F_2}{n_2} \right)_i \right| \tag{2.35}$$

where F_j is the cumulative frequency in the i^{th} interval with n_j values for the two samples $j = 1, 2$. The intervals must be of equal size and have the same limits for both samples. Special tables give the confidence interval for this test variable. Several other tests are available for comparing independent samples such as the U–test of Wilcoxon, Mann and Whitney which uses the rank in its test variable.

2.5.3 Multivariate Statistical Methods

When the analysis yields a large number of parameters for each object, it is difficult to overview relations between the individual variable. Multivariate statistical methods can be applied in such a situation to give an objective description of the data. The Principal Components Analysis is used to determine the true dimensionality of the data set and find the best linear combination of the parameter for following studies (see Chatfield and Collins 1980). A typical example of such analysis was performed by Okamura (1985) on photometric data from Virgo Cluster galaxies.

Structures and groups in large data sets can be located by means of a Cluster Analysis which constructs a set of groups in the data using a given distance measure. A large variety of methods for clustering are available with different distance definitions providing both hierarchical and non–hierarchical groups (see Murtagh 1986 and references therein). These techniques are used especially for classification problems in astronomy.

References

Baker, A.E. : 1925, *Proc. Roy. Soc. Edingburgh* **45** 166.

Bijaoui, A. : 1981, *Image et information*, Masson, Paris.

Bevington, P.R. : 1969, *Data Reduction and Error Analysis for the Physical Sciences*, McGraw–Hill, New York.

Capaccioli, M., Held, E. : 1983, private communication.

Chatfield, C., Collins, A.J. : 1980, *Introduction to Multivariate Analysis*, Chapman and Hall.

Frieden, B.R. : 1972, *J. Opt. Soc. Am.* **62**, 511.

Goad, L.E. : 1980, *SPIE* **264**, 136.

Grosbøl, P. : 1979, *Image Processing in Astronomy*, Eds. G. Sedmark, M. Capaccioli, R.J. Allen, Osservatorio Astronomico di Trieste, p. 371.

Grosbøl, P. : 1980, *SPIE* **264**, 118.

Helstrom, C.W. : 1967, *J. Opt. Soc. Am.* **57**, 297.

Herzog, A.D., Illingworth,G. : 1977, *Astrophys. J. Suppl.* **33**, 55.

Horner, J.L. : 1970, *Applied Op.* **9**, 167.

Lucy, L.B. : 1974, *Astron. J.* **79**, 745.

Martin, R., Lutz,R.K. : 1979, *Image Processing in Astronomy*, Eds. G. Sedmark, M. Capaccioli, R.J. Allen, Osservatorio Astronomico di Trieste, p. 211.

Murtagh, F. : 1986, *ESO Preprint No.* **448**

Newell, E.B. : 1979, *Image Processing in Astronomy*, Eds. G. Sedmark, M. Capaccioli, R.J. Allen, Osservatorio Astronomico di Trieste, p. 100.

Newell, B., O'Neil, Jr., E.J. : 1977, *Pub. Astron. Soc. Pac.* **89**, 925.

Okamura, S. : 1985, *Proc. of ESO Workshop on The Virgo Cluster*, Eds. O.–G. Richter and B. Binggeli, ESO, p. 201.

Pratt, N.M. : 1977, *Vistas in Astronomy* **21**, 1.

Pratt, W.K. : 1978, *Digital Image Processing*, Wiley–Interscience, New York.

Rosenfeld, A., Kak, A.C. : 1982 *Digital Picture Processing* Vol. **1-2**, Academic Press, New York.

Tukey, J.W. : 1971, *Exploratory Data Analysis*, Addison–Wesley, Reading, Mass.

Wells, D.C. : 1980, *SPIE* **264**, 148.

Chapter 3

CCD Reductions

3.1 Introduction

This chapter describes the design and the philosophy of the CCD package, its main features, and how to use the tools most efficiently.

With the installation of new instruments on the La Silla Telescopes in addition to the availability of CCDs offering large pixel areas and higher quantum efficiency, the variety of observing modes has grown and, as an obvious consequence, the amount and the diversity of data taken have dramatically increased. It is clear that the MIDAS CCD reduction software should be able to cope with these improvements and hence requires compatibility with the hardware as it is offered to the community.

When designing the basic layout of the CCD software the following basic requirements were kept in mind:

- it should be robust;
- it should be user friendly, easy to use;
- it should offer processing possibilities for a large variety of instruments;
- it should be able to operate both in an automatic mode (to handle large quantities of data) as well as in single command mode (for single image analysis);
- it should offer flexible reduction procedures;
- it should be intelligent.

How intelligent the system is depends on the capabilities of other parts in the data acquisition, archiving, and reduction systems. In this respect the development of the CCD package took place at the right time: the MIDAS Data Organizer package offers the possibility to quickly create a MIDAS table containing the science frames to be reduced and their associated calibration frames. The CCD package uses this facility that is based on selection criteria, similar to the ones that are used for the MIDAS table file system. Also, the ESO Archive project has accomplished that a number of telescope and instrument specifications, needed to come to a (partially) automated CCD reduction, can be retrieved from the frame descriptors.

In order not to re-invent the wheel, existing CCD packages have been consulted, and, when useful, ideas have been implemented in the MIDAS CCD software. Its design has largely profited from the IRAF CCDRED package written by Frank Valdes. Parts of its documentation are used for this manual. Also, discussions with Peter Draper, the author of the STARLINK CCDPACK package, are acknowledged.

The document is split into several sections. They describe in detail the various steps in the CCD reduction, starting from reading in the science and calibration frames and ending with the final cosmetic fix up of the final calibrated frames. Some background information about CCD detectors can be found in the MIDAS Users' Guide, Volume B, Appendix B. Additional information about the CCD commands can be obtained from their help descriptions.

The emphasis of the CCD package is on direct imaging. However, since the first processing steps for spectral data are rather similar to direct imaging major parts of its functionality can also be used for processing these data.

***** WARNING *****

The MIDAS CCDRED context is partly based on the current status of the ESO Archiving project as well as the Data Organizer Context, in particular with regard to exposure types and naming of files. Since both projects may be subject to changes in the future, and because of user experiences and suggestions for improvements, the CCDRED context may undergo adjustments accordingly.

3.2 Nature of CCD Output

The nominal output X_{ij} of a CCD–element to a quantum of light I_{ij} can be given as

$$X_{ij} = M_{ij} \times I_{ij} + A_{ij} + F_{ij}(I_{ij}) \tag{3.1}$$

where the additive contribution A_{ij} is caused by the dark current, by pre-flashing, by charge that may have skimmed from columns having a deferred charge (skim) and by bias added to the output electronically to avoid problems with digitizing values near zero. Quantum and transfer efficiency of the optical system enter into the multiplicative term M. The term I consist of various components: object, sky and the photons emitted from the telescope structure. It is known that the response of a CCD can show non-linear effects that can be as large as 5-10%. These effects are represented by the term F_{ij}.

In the following we ignore the pre-flash and skim term, and hence only take the bias and dark frames into account. The objective in reducing CCD frames is to determine the relative intensity I_{ij} of a science data frame. In order to do this, at least two more frames are required in addition to the science frame, namely:

- dark frames to describe the term A_{ij}, and
- flat frames to determine the term M_{ij}.

The dark current *dark* is measured in absence of any external input signal. By considering a number of dark exposures a medium < *dark* > can be determined:

$$\langle \mathrm{dark(i,j)} \rangle = \mathrm{dark(i,j) + bias} \tag{3.2}$$

The method to correct the frame for multiplicative spatial systematics is know as flat fielding. Flat fields are made by illuminating the CCD with a uniformly emitting source. The flat field then describes the sensitivity over the CCD which is not uniform. A mean flat field frame with a higher S/N ratio can be obtained buy combining a number of flat exposures. The mean flat field and the science frame can be described by:

$$\overline{\mathrm{flat(i,j)}} = \mathrm{M(i,j) \times icons + dark(i,j) + bias} \tag{3.3}$$

$$science(i,j) = M(i,j) \times intens(i,j) + dark(i,j) + bias \tag{3.4}$$

where $intens(i,j)$ represents the intensity distribibition on the sky, and *icons* a brightness distribution from a uniform source. If set to the average signal of the dark corrected flat frame or a subimage thereof:

$$\mathrm{icons} = \langle \mathrm{flat - dark} \rangle \tag{3.5}$$

then the reduced intensity frame *intens* will have similar data values as the original science frame *science*.

Combining Eqs.(3.2), (3.3) and (3.4) we isolate:

$$\mathrm{intens(i,j)} = \frac{\mathrm{science(i,j)} - \langle \mathrm{dark(i,j)} \rangle}{\overline{\mathrm{flat(i,j)}} - \langle \mathrm{dark(i,j)_F} \rangle} \times \mathrm{icons} \tag{3.6}$$

Here *icons* can be any number, and term $\langle dark(i,j) \rangle$ now denotes a dark frame obtained by *e.g.* applying a local median over a stack of single dark frames. The subscript in $\langle dark(i,j)_F \rangle$ denotes that this dark exposures may necessarily be the same frame used to subtract the additive spatial systematics from the raw science frame.

The mean absolute error of $intens(i,j)$ yields with $icons = 1$ (only the first letter is used for abbreviations):

$$(\Delta I)^2 = \left(\frac{\partial I}{\partial S}\right)^2 (\Delta S)^2 + \left(\frac{\partial I}{\partial D}\right)^2 (\Delta D)^2 + \left(\frac{\partial I}{\partial F}\right)^2 (\Delta F)^2 \tag{3.7}$$

Computing the partial derivatives we get

$$(\Delta I)^2 = \frac{(F-D)^2(\Delta S)^2 + (S-F)^2(\Delta D)^2 + (S-D)^2(\Delta F)^2}{(F-D)^4} \tag{3.8}$$

A small error ΔI is obtained if ΔS, ΔD and ΔF are kept small. This is achieved by averaging Dark, Flat and Science frames. ΔI is further reduced if $S = F$, then Equation (3.8) simplifies to

$$(\Delta I)^2 = \frac{(\Delta S)^2 + (\Delta F)^2}{(F-D)^2} \tag{3.9}$$

This equation holds only at levels near the sky–background and is relevant for detection of low–brightness emission. In practice however it is difficult to get a similar exposure level for the *flatfrm* and *science* since the flats are usually measured inside the dome. From this point of view it is desirable to measure the empty sky (adjacent to the object) just before or after the object observations. In the case of infrared observations this is certainly advisable because of variations of the sky on short time scales.

3.3 General Overview of the package

The CCDRED package provides a set of basic commands that perform the various calibration steps. These are the combination of calibration frames, subtraction of the bias level (determined from the overscan area or from a separate bias frame), correction for dark current, division by the flat field, correction for illumination, and correction for the fringe pattern. Also, tools are provided for trimming the frame, and for correction of bad pixels. By combining these basic reduction steps a complete reduction pipeline procedure is built, that enables the user to execute a complete automatic reduction of all science frames.

When the context `CCDRED` enabled, a keyword structure is created which contains the parameters for the calibration steps and to control these. These parameters determine which and how the available reduction steps will be executed. Obviously, in order to get the desired result, these keywords should be correctly filled. Other keywords contain general information, *e.g.* about the telescope/instrument being used. Finally, keywords are created to contain the names of important frame descriptors, like the standard MIDAS descriptor for the exposure time (`O_TIME`). In case information is absent sensible defaults will be used in mostly. Finally, a number of keywords contain information about the status of the reduction.

All operations on a frame that are successfully executed are recorded in its descriptors. This facility, which includes updating the HISTORY descriptor, avoids repetition of reduction sequences, and provides the user with information about what has been done to the data.

Apart from commands that do the actual work, a number of commands will help the user to manage keywords and descriptors and their contents/values. Basically, this means displaying and changing parameters. Also, commands exist to store the current parameter settings and to retrieve these after a session is restarted.

Most of this manual is geared towards the "automatic approach", meaning that it is assumed that the user will use the intelligence that has been built into the system. However, the manual also includes documentation about how to execute single basic steps.

The MIDAS CCD package works in combination with the MIDAS Data Organizer which generates, using a set of selection rules, a MIDAS table containing the science frames and their associated calibration frames. Within this chapter this table is referred to the **Association Table**. This table is important for the package: a number of the commands will only work if the table exists: pipe line processing is only possible with the Association Table. The MIDAS Data Organizer is extensively documented elsewhere in this MIDAS User's Guide.

3.4 Setting, Saving, and Retrieving CCD Keywords

Since the number of parameters involved in a complete CCD reduction is quite large, most command parameters have default values. These defaults are taken from the CCD keyword structure. So, after the data have been read in and organized, you can set the CCD keywords to control the CCD reduction process. Basically, one can divide the CCD keywords in three categories:

- general keywords e.g. for telescope, instrument/detector specifications (which CCD has been used; what is the useful area on the chip; what is the read-out noise, its gain; what is the overscan area) and the entries, needed for the various steps in the calibration sequence. Most of the keywords are filled by the `LOAD/CCD` command (see below).
- calibration keywords related to the handling of calibration data, like fitting of the overscan area, the creation of master calibration frames, the method of bad pixel correction, the creation of illumination and fringe frames. For combining the calibration frames in master calibration frames the system is provided with a set of keywords for each exposure type (see below);
- reduction keywords for controlling the reduction of the science frame(s). Examples are: should the overscan area be used to determine the bias; should we correct for bad pixels; is dark current to be subtracted.

Most of the CCD commands get their input parameters mainly from the CCD keywords. However, most commands also accept a limited number of input parameters on the command line. These parameters will supersede the corresponding keywords. However, apart from a few cases the keyword setting itself is not modified: principally, keywords can only be changed by the `SET/CCD` command. For handling all these keywords the CCDRED package in equipped with five commands: `HELP/CCD`, `SHOW/CCD`, `SET/CCD`, `SAVE/CCD` and `INIT/CCD`.

`HELP/CCD` without parameters gives an overview of all CCD commands available. With an existing CCD keyword as parameter the command will show the present values of that keyword and a short explanation of it use.

`SHOW/CCD` returns you the current CCD keyword contents. Given the number of keywords, to display the information the keywords are grouped and displayed according to their functionality:

- **GE** keywords concerning general information;
- **BS** keywords for combining bias frames;
- **DK** keywords for combining the dark frames;
- **FF** keywords for combining the flat fields;
- **SK** keywords for combining the sky frames;

- **OT** keywords for combining other exposure types;
- **OV** keywords for fitting the overscan region;
- **MO** keywords for the mosaicing commands;
- **FX** keywords for the bad pixel correction;
- **IL** keywords for the creation of an illumination frame;
- **FR** keywords for the creation of a fringe frame;
- **SC** keywords for control of the actual reduction sequence.

With `SET/CCD` a maximum of 8 keywords in the CCD context can be changed in one go.

After (partially) having finished a CCD reduction the user may want to store the current keyword setting. This can be done using the command `SAVE/CCD sav_table`. The command will store the keywords in descriptors of a CCD save table with descriptor information only. The keyword setting can be restored by the command `INIT/CCD sav_table`, where `sav_table` is the table containing the CCD keyword setting, previously saved.

3.5 Calibration Frames and Naming Convention

The CCD package is based on so-called data sets. A data set contains a science frame, all its associated raw calibration frames and the master calibration frames created by combining and processing the raw calibration frames. Depending on the processing to be done on the science frame one or more master calibration frames are to be created.

Basically, the creation of a master calibration frame can be done in two ways. Either one creates a MIDAS catalogue which contains the names of all single calibration frames to be combined in to the master frame, or, in the case of pipe line reduction, one use the Association Table. Since the first method is straightforward we concentrate on the use of the Association Table.

In order to achieve a maximum of efficiency and to interface the package with the Data Organizer, the naming convention for master calibration frames is identical to the naming convention for the naming of the master calibration frames in the latter package. Here the name of a master calibration frame is a composition of the generic prefix and all frame numbers of the calibration frames to be used and therefore selection in the DO context. *E.g.* the master calibration frame `susi_12_123_1245` is a combination of the frames `susi0012`, `susi0123`, and `susi1245`.

After the association process by the DO, the name of the master calibration frame is stored in a separate column in the Association Table. The name of this frame is defined as described above. The names of single calibration frames are however also available in the Association Table. This obviously offers the possibility of simply combining all single raw calibration frames in a master one. To execute this brute force combining the column for the master frame should contain an asterisk `*`. The name of the master will then be a composition of the name of the science frame to which the master calibration frame is associated plus the postfix `_exp`. Here, `exp` is the exposure type, stored in a MIDAS frame descriptor (`EXP_TYPE`), and the name of the column with this exposure type in the Association Table and containing the names of the raw calibration frames. *E.g.* the master calibration frame `susi0100_bias` is created by combining all bias frames associated to the science frame `susi0100` and stored in the Association Table.

Currently, the following exposure types are supported:

- `bias` - bias frames:
  These are zero second integration exposures obtained with the same pre-flash (if any) you have used for your scientific exposures. The bias frame will correct for the small positive voltage added to the true signal from the CCD and determines the photometric zero point of the electronic system.

- `dk` - dark current frames:
  These are long exposures taken with the shutter closed. Dark emission can be caused by several sources (*e.g.* overall background emission, luminescence form source on the CCD) and will add charge linearly with exposure time.

- `ff`, `ff-dome`, `ff-screen` - flat field frames:
  These are used to remove the pixel-to-pixel variations across the chip. In some cases dome flats (exposure of an illuminated screen) or projection flats (exposures of a quartz lamp illuminating the spectrograph slit) will be sufficient to remove the chip variations.

- `ff-sky` - blank sky exposures:
  As an alternative to the dome or projector flats many observers doing direct imaging try exposures of blank sky field(s). A clear advantage is that the sky field to be obtained from the blank sky exposures have exactly the colour of the night sky. However, this method of flat field can only be used in absence of fringing and low telescope background emission.

Other calibration frames that can be used in the calibration process are:

- illumination frames:
  This calibration frame may be used to correct for the fact that the flat field calibration frame do not have the same illumination pattern as the observations of the sky. If this is the case, applying the flat field correction may cause a gradient in the sky background.

- fringe frames:
  It may happen that using a (thinned) CCD a fringe pattern becomes apparent in the frame. The pattern is caused by interference of monochromatic light (*e.g.* night sky lines) falling on the chip and are not removed by other calibration and correction steps. To correct one needs to construct needs a really blank sky frame.

*** **WARNING** ***

In principle, the CCD package allows the use of any name for the calibration frames. However, to make the reduction of CCD frames as easy as possible it is recommended to use the above described naming scheme. It is highly recommended to use it. Using different names may, under particular circumstances, lead to complications, in particular in the case of pipe line reduction.

3.6 Setting up the Reduction Procedure

3.6.1 Loading the Telescope and Instrument Specifications

For reducing data from ESO instrumentation it is assumed that the FITS headers conform the standards set by the document "ESO Archive, Data Interface Requirements". In particular, the ESO specific parameters are stored in a special set of FITS keywords (the ESO hierarchical keywords). Using this standard, while reading in the data FITS keywords are stored in well defined MIDAS descriptors of the MIDAS frames.

As an example, in the ESO case the exposure type is stored in the ESO hierarchical FITS keyword 'HIERARCH ESO GEN EXPO TYPE'. The ESO-MIDAS FITS reader converts that keyword into the MIDAS descriptor _EGE_TYPE. Among the valid exposure types are (according the ESO Archive document): `bias`, `ff`, `ff-dome`, `ff-screen`, `ff-sky`, `dk`, `sci`. Therefore, when the CCD context is started or after executing the command `INIT/CCD` links are created between the frame descriptors and the corresponding CCD keyword. With these links the CCD package knows which exposure types are used and which frame descriptor contains the exposure type.

To fill the CCD keywords with these descriptor names a separate procedure is used: `eso_descr.prg`. A copy of this procedure is put in your working directory. If you need to change one or more descriptor names (*e.g.* the descriptor name of the exposure time), make the modification(s) and run the modified procedure, using `@@ eso_descr`.

The CCD package has been developed to reduce CCD data coming from ESO's telescopes on La Silla. However, some flexibility has been built-in to enable the reduction of data coming from non-ESO telescope/instrument combinations, cases in which the telescope and instrument parameters are different. To fill the CCD keywords with the telescope and instrument setup parameters, like name of the telescope, CCD used, read-out noise, frame size, etc. the command `LOAD/CCD` can be used. This command reads these parameters for the MIDAS table `eso_specs.tbl` that, in case it is not present in your working directory, will be created. At initialization of the CCD context a copy of this table is put in the user's directory. In case non-ESO observing facilities have been used, or you want to modify or append the table with your own setup parameters, you can use the standard table commands *e.g.* `EDIT/TABLE`.

3.6.2 Data Retrieval and Organization

The first step in the CCD reduction is the storage of the data on disk. The MIDAS FITS reader `INTAPE/FITS` provides this functionality. The data probably will contain frames of different exposure type (biases, darks, flat fields, science frames, etc), and possibly

SCI	BIAS	...	QUAL_BIAS	BIAS_MASTER
susi0097.bdf	susi0124.bdf	...		susi_124_125.bdf
susi0098.bdf	susi0124.bdf	...		susi_124_125.bdf
susi0099.bdf	susi0126.bdf	...		susi_124_125_126.bdf
susi0100.bdf	susi0124.bdf	...		susi_125_126_127.bdf

Table 3.1: Example of an Association Table

obtained with different filters. After the `INTAPE/FITS` command these different frames are stored on disk.

3.6.3 The Association Table

The second step is to inspect the data, to define a set of criteria to select the science frames and their associated calibration frames, and to use the MIDAS Data Organizer to do the selection and to create the Association Table. Typically, the Association Table looks as displayed in Table 3.1. Depending on available exposure types and on the selection, like for the bias exposure additional columns for darks and flats can be present in the table. The ... represent the three dimensional character of the Association Table, where in this case, the single bias frames are stored in the third dimension of column `BIAS`.

After inspecting the master calibration frame, the user may decide that this is not what (s)he wants and that using another master calibration frame would be better. For example, one can require that in the calibration process of frame `susi0099.bdf` the master bias `susi_124_125.bdf` is to be used, *i.e.* the bias frames associated with the science frames `susi0097.bdf` and `susi0098.bdf`. The change can either be made by running the DO package once more with slightly different selection rules, or by using the command `EDIT/TABLE`.

In addition to the standard naming convention three other input formats can be used in the Association Table. The first one is the use of non-standard name(s) (*i.e.* frame names that are not related to a data set), *e.g.* `stdbias.bdf` instead of `susi_124_125.bdf`. In this case the system requires that the master bias `stdbias.bdf` already exists. A second possibility is to have an asterisk `*`, indicating that all single raw calibration frame associated with a particular science frame are to be combined. In that case the names of the single frames will be retrieved for the Association Table. The third possibility is to store constants in the Association Table, *e.g.* `294` instead of `susi_124_125.bdf`.

An example of these three input possibilities is given in Table 3.2. Here, for the reduction of the frame `susi0097.bdf` a constant bias value of `294` is used while the frame `susi0099.bdf` will be calibrated using all available single bias frames associated with that science frame. Frame `susi0100.bdf` will be reduced with the standard bias frame `stdbias.bdf` that is assumed to be present in the working directory.

SCI	BIAS	...	QUAL_BIAS	BIAS_MASTER
susi0097.bdf	susi0124.bdf	...		295
susi0098.bdf	susi0124.bdf	...		susi_124_125.bdf
susi0099.bdf	susi0126.bdf	...		*
susi0100.bdf	susi0124.bdf	...		stdbias.bdf

Table 3.2: Example of a manually modified Association Table

3.7 Basic Reduction Steps

The software available in the CCD package takes care of the relative calibrations of the pixel intensities, of averaging, and of cleaning frames. Cleaning in this context means removal of the instrumental signature and other defects from the frames.

A full reduction of CCD data involves the steps outlined below:

- fit and subtract a readout bias given by the overscan strip;
- trim the frame of the overscan strip and other irrelevant columns and/or rows;
- combine the bias, dark, and flat calibration frames (if taken);
- subtract the average bias frame;
- remove the defects from the average dark frame;
- scale the average dark frame and subtract;
- remove the defects from the average flat frame;
- prepare the final flat (subtract dark and normalize; possibly apply illumination correction);
- divide by the flat field;
- fix the bad pixels in the output frame;
- correct for fringe pattern;
- combine science frames.

Some of these steps are optional and depend on the kind of data you have taken. In addition to the operations described here, MIDAS contains a number of other operations like removing of cosmic rays. As indicated above, all steps can be executed in an automatic (batch) mode provided the keyword settings have been done correctly.

In the automatic procedure, the highest level of processing is the loop over all science frames selected in the Association Table. If, for whatever reason, the processing of a frame fails, this frame is skipped and the processing is continued with the next one.

The CCD package system provides a single command for doing the entire calibration of the CCD frames: `REDUCE/CCD`. The command is a composition of a number of lower level procedures, each of them taking care of a particular part of the calibration procedure, and executable via a separate MIDAS command. These components are: overscan correction, trimming, combining, bias correction, dark subtraction, flat fielding, illumination correction, and fringe correction.

Whether you want one or more calibration steps be executed depends on the settings for the various options. Therefore, before starting be sure that the keywords for the reduction are set correctly. These keywords and their meaning are listed in Table 3.3. Also, the keywords for combining calibration frames should be checked. Furthermore, fill the keyword `CCDASSOC` with the name of the Association Table, and check that this table is correct. All keywords can be filled and displayed with the commands `SET/CCD` and `SHOW/CCD`.

Keyword	**Content**	Default	**Description**
`CCD_IN`	name	?	input frame or table
`SC_TYP`	exp. type	*	exposure type of input frame
`SC_PROC`	yes|no	no	list processing only
`SC_SCAN`	yes|no	no	applied overscan offset correction
`SC_TRIM`	yes|no	no	trim the frame from overscan area
`SC_FXPIX`	yes|no	no	bad pixel correction
`SC_BSCOR`	yes|no	yes	bias correction
`SC_DKCOR`	yes|no	yes	dark current correction
`SC_FFCOR`	yes|no	yes	flat field correction
`SC_ILCOR`	yes|no	no	illumination correction
`SC_FRCOR`	yes|no	no	fringing correction
`SC_BSFRM`	name		bias calibration frame
`SC_DKFRM`	name		dark current calibration frame
`SC_FFFRM`	name		flat field calibration frame
`SC_ILFRM`	name		illumination calibration frame
`SC_FRFRM`	name		fringing calibration frame

Table 3.3: Keywords for setting the reduction process

3.8 Preparing Your Calibration Frames

Before the actual processing can start, multiple calibration frames have to be combined into a master. The command that takes care of combining the calibration frames in the CCD package is `COMBINE/CCD`. It provides various methods to improve the S/N statistics in the resulting output frame. Depending on the actual parameter settings, the command can take into account the exposure times in the combining process, and can adjust for a (variable) sky background.

3.8.1 Input and Output

The `COMBINE/CCD` command offers the possibility to combine a number of input images using different combining methods. `COMBINE/CCD` takes three input parameters at maximum: the exposure type of the images to be combined, the input frames themselves and and output frame. The various command options can be chosen by setting a number of specific keywords.

The first input parameter should contain the exposure type of the images to be combined. Possible choices are: BS (for bias); FF (for flat fields), DK (for dark), SK (for sky images), and OT (for others). The combining options the command offers are controlled by a set of exposure type dependent keywords, all starting with this two letter identification that has been given as the first input parameter. These keywords control various combining methods, scaling and offset corrections, as well as weighting (see below).

The second input parameter is the input frames to combine. The input can be provided in different ways:

1. by a list of images; e.g. `frame01,frame02,frame03`;
2. by a MIDAS catalogue; e.g. `framecat.cat`;
3. by a MIDAS Data Organizer (DO) output table (with the extension `.tbl`), the Association Table (see Section 3.6.3).

The parameter for the output frame is required in case the input for the second parameter is a catalogue or a string of input frames. In the case of DO input (association) table, from the name of the output calibration frame in the table the command extracts the names of all requested single calibration frames and combines these frames in the output frame. The name of the output master frame can be indicated with an asterisk, meaning that all associated single calibration frames have to be combined. In that case the names of these single frames are taken for the calibration column in the Association Table, *e.g.* `BIAS`, `DK`, etc. See also Section 3.6.3. By default, the input is taken from the keyword `CCD_IN`.

In addition to the output calibration frame the combined sigma frame can be generated. This frame is the standard deviation of the input frames about the output frame.

Before the actual combining is done the exposure type descriptors of the input frames are compared with the descriptor type stored in the keyword `'exp'_TYP`. In case this keyword is filled with '*' or '?' all exposure types are allowed. Else, a fatal error will follow if the keyword content is not equal to the exposure type(s) of one or more input frames.

3.8.2 Combining Methods

Except for summing the frames together, combining frames may require correcting for variations between the frames due to different exposure times, sky backgrounds, extinctions, and positions. Currently, scaling and shifting corrections are included. The scaling corrections may be done by exposure times or by using statistics in each frame over a selected part of the image. The statistics can reveal (depending on the keyword `'exp'_STA`) setting, (where `'exp'` is the exposure type) for each image the mean, median, or the mode. In the following we refer to the value by MMM. Additive shifting is also done by computing the statistics in the frames.

The region of the frames in which the statistics is computed can be specified by the keyword `'exp'_SEC`. By default the whole frame is used. A scaling correction is used when the flux level or sensitivity is varying. The offset correction is used when the sky brightness is varying independently of the object brightness. If the frames are not scaled then special routines combine the frames more efficiently.

Below follows a simple overview how the weighting, scaling and offset parameters are determined. All obviously depend on the settings of the keywords `'exp'_SCA` `'exp'_OFF`, `'exp'_WEI`, and `'exp'_EXP`. The overview makes clear that offset corrections will only be applied if the scaling correction is switched off. The same is true for applying an exposure time correction.

```
==========================================================================
  o_i = 0.0
  w_i = 1.0
  s_i = 1.0

  exp_SCA=yes
     s_i = M_i
     exp_WEI=yes
        w_i = sqrt(N*s_i)

  exp_SCA=no
     exp_EXP=yes
        s_i = e_i
        exp_WEI=yes
           w_i = sqrt(N*s_i)

     exp_OFF=yes
        o_i = M_i/s_i
        exp_WEI=yes
           w_i = sqrt(N*s_i/o_i)

  s_i = s_i/s_mean
  o_i = (o_i - o_mean) * s_mean
  w_i = w_i/w_sum
```

```
-------------------------------------------------------------------------
  key: o_i:    offset for frame i
       o_mean: mean offset over all input frames
       s_i:    scale factor for frame i
       s_mean: mean scale factor over all input frames
       w_i:    weight factor for frame i
       w_sum:  sum over all weight factors of all input frames
       e_i:    exposure time of frame i
       M_i:    MMM of frame i
       N:      number of of frames previously combined
=========================================================================
```

In the combining no checks are done on the reduction status of the input frames and no attempts are made for any calibration correction like for bias or dark. Hence, in more complicated reduction sequences the user should be sure not to combine *e.g.* flat fields that have been corrected for bias and dark with flats fields that are not corrected.

Except for medianing and summing, the frames are combined by averaging. The average may be weighted by

$$weight = (N * scale) * *1/2 \tag{3.10}$$

where N is the number of frames previously combined (the command records the number of frames combined in the frame descriptor), *scale* is the scale factor depending on the keyword settings listed above (`s_i` or `s_i/o_i`). In most of the applications $N = 1$, *i.e.* the input calibration frames are the original ones and not the result of previous combinings.

There are a number of algorithms which may be used as well as applying statistical weights. The algorithms are used to detect and reject deviant pixels, such as cosmic rays. The choice of algorithm depends on the data, the number of frames, and the importance of rejecting cosmic rays. The more complex the algorithm the more time consuming the operation. For every method pixels above and below specified thresholds can be rejected. These thresholds are stored in the keyword `'exp'_MET`. If used the input frames are combined with pixels above and below the specified threshold values (before scaling) excluded. The sigma frame, if requested, will also have the rejected pixels excluded.

The following list summarizes the algorithms. Further algorithms are available elsewhere in MIDAS (see `COMPUTE/...`, `AVERAGE/...`), or may be added in time.

- Sum - sum the input frames.
  The input frames are combined by summing. Summing is the only algorithm in which scaling and weighting are not used. Also no sigma frame is produced.

- Average - average the input frames.
  The input frames are combined by averaging. The frames may be scaled and weighted. There is no pixel rejection. A sigma frame is produced if more than one frame is combined.

- Median, MMedian - (mean) median the input frames.
The input frames are combined by medianing each pixel. Unless the frames are at the same exposure level they should be scaled. The sigma frame is based on all input frames and is only a first approximation of the standard deviations in the median estimates. The second method does an averaging around the found median in a certain interval in order to take into account the distribution of the values near the median. This is in effect the same what `AVERAGE/IMAGE` also does using the parameter setting 'options = median,low,high'. The required data interval has to be defined by the `exp_CLP` keyword and is assumed to specify relative limits to the determined median – same as in `AVERAGE/IMAGE` (both limits positive).

- Minreject, maxreject, minmaxreject - reject extreme pixels.
At each pixel after scaling the minimum, maximum, or both are excluded from the average. The frames should be scaled and the average may be weighted. The sigma frame requires at least two pixels after rejection of the extreme values. These are relatively fast algorithms and are a good choice if there are many frames (>15).

- Sigclip - apply a sigma clipping algorithm to each pixel.
The input frames are combined by applying a sigma clipping algorithm at each pixel. The frames should be scaled. This only rejects highly deviant points and so includes more of the data than the median or minimum and maximum algorithms. It requires many frames (>10-15) to work effectively. Otherwise the bad pixels bias the sigma significantly. The mean used to determine the sigmas is based on the "minmaxrej" algorithm to eliminate the effects of bad pixels on the mean. Only one iteration is performed and at most one pixel is rejected at each point in the output image. After the deviant pixels are rejected the final mean is computed from all the data. The sigma frame excludes the rejected pixels.

- Avsigclip - apply a sigma clipping algorithm to each pixel.
The input frames are combined with a variant of the sigma clipping algorithm which works well with only a few frames. The images should be scaled. For each line the mean is first estimated using the "minmaxrej" algorithm. The sigmas at each point in the line are scaled by the square root of the mean, that is a Poisson scaling of the noise is assumed. These sigmas are averaged to get a line estimate of the sigma. Then the sigma at each point in the line is estimated by multiplying the line sigma by the square root of the mean at that point. As with the sigma clipping algorithm only one iteration is performed and at most one pixel is rejected at each point. After the deviant pixels are rejected the file mean is computed from all the data. The sigma frame excludes the rejected pixels.

The "avsigclip" algorithm is the best algorithm for rejecting cosmic rays, especially with a small number of frames, but it is also the most time consuming. With many frames (>10-15) it might be advisable to use one of the other algorithms ("maxreject", "median", "minmaxrej") because of their greater speed.

The choice of the most optimal combining algorithm will clearly depend on the nature of the data and on the exposure type. Therefore, for every supported exposure type the

CCD context contains a default combining setup. Currently, there are five combining setups stored in the CCD keywords, all starting with a specific two letter prefix: for bias `BS_`, dark `DK_`, dome flats `FF_`, sky flats `SK_`, and for all other exposure types `OT_`. At initialization these keywords are filled with sensible defaults. Below we will shortly comment on combining the various calibration frames and list the default keywords settings.

3.8.3 Combining Bias Frames

Combination of the bias frames is straight forward: filters used and integration times do not play a role, and in most cases the bias frames can be treated with the same weight. The default keyword setting for the bias combining is displayed in Table 3.4.

Keyword	**Content**	**Default**	**Description**
`CCD_IN`	input name	?	table of input frames
`BS_TYP`	descriptor value	*	exposure type to check, if any
`BS_SIG`	yes|no	no	create sigma image
`BS_MET`	comb. method	maxrej	type of combining operation
`BS_DEL`	yes|no	no	delete cal. frames afterwards
`BS_STA`	mean|median|mode	mode	correct. by Mode/Median/Mean
`BS_EXP`	yes|no	no	scale by exposure time
`BS_SCA`	yes|no	no	scale by MMM
`BS_OFF`	yes|no	no	add offset from MMM
`BS_WEI`	yes|no	yes	use a weighted average
`BS_SEC`	area	[<,<:>,>]	area for finding MMM
`BS_RAN`	real,real	-99999,99999	valid pixel range
`BS_CLP`	real	3.,3.	low/high sigma clipping factor
`BS_NUL`	real	`NULL(2)`	value for rejections

Table 3.4: Keywords for combining bias calibration frames

3.8.4 Combining Dark Frames

Similarly to the bias combination one can obtain an average dark current frame. However, one should consider whether the dark correction should really be applied: one needs a reasonable number of dark frames in order not to degrade the S/N, and obviously this takes telescope time. Because the dark level depends on the exposure time, weighting the input dark frames should be considered. Another possibility would be simply to take the average dark value and to scale that number with the exposure time. Filter type is not important. Table 3.5 shows the default setting for combining dark frames.

If the bias is stable enough to allow taking averages, one might argue that it is not really needed. In that case, provided a good linearity of the CCD, one could do with the subtraction of an average dark frame. If the bias is unstable, one should be careful with

<table>
<tr><th>Keyword</th><th>Content</th><th>Default</th><th>Description</th></tr>
<tr><td>CCD_IN</td><td>input name</td><td>?</td><td>table of input frames</td></tr>
<tr><td>DK_TYP</td><td>descriptor value</td><td>*</td><td>exposure type to check, if any</td></tr>
<tr><td>DK_SIG</td><td>yes|no</td><td>no</td><td>create sigma image</td></tr>
<tr><td>DK_MET</td><td>comb. method</td><td>avsigcl</td><td>type of combining operation</td></tr>
<tr><td>DK_DEL</td><td>yes|no</td><td>no</td><td>delete cal. frames afterwards</td></tr>
<tr><td>DK_STA</td><td>mean|median|mode</td><td>mode</td><td>correct. by Mode/Median/Mean</td></tr>
<tr><td>DK_EXP</td><td>yes|no</td><td>yes</td><td>scale by exposure time</td></tr>
<tr><td>DK_SCA</td><td>yes|no</td><td>no</td><td>scale by MMM</td></tr>
<tr><td>DK_OFF</td><td>yes|no</td><td>no</td><td>add offset obtained from MMM</td></tr>
<tr><td>DK_WEI</td><td>yes|no</td><td>yes</td><td>use a weighted average</td></tr>
<tr><td>DK_SEC</td><td>area</td><td>[<,<:>,>]</td><td>area for computing MMM</td></tr>
<tr><td>DK_RAN</td><td>real,real</td><td>-99999,99999</td><td>valid pixel range</td></tr>
<tr><td>DK_CLP</td><td>real</td><td>3.,3.</td><td>low/high sigma clipping factor</td></tr>
<tr><td>DK_NUL</td><td>real</td><td>NULL(2)</td><td>value for rejections</td></tr>
</table>

Table 3.5: Keywords for combining dark calibration frames

simply combining bias frames. In that case the better solution might be an average of two bias frames, taken just before and after each flat or sky exposure.

3.8.5 Combining Flat Fields

In combining the flat field frames the filter type is of importance. Hence, the combining command will check for consistency of the keyword containing the filter type. The combining input parameters can be set up in the flat parameter table, similar to the dark frames. Table 3.6 show the default keyword settings.

3.8.6 Combining Sky Frames

The procedure is similar to the flat field combining procedure, except that you may want to scale the sky level using the mean, median or the mode of the intensity distribution or taking into account the exposure time. Hence, in the combining, like in the case of combining flat fields, correct weighting of the frame is important. Combining should be done filter by filter. See Table 3.7 for the keywords.

3.8.7 Combine Example

As an example of the use of a Association Table similar to the one displayed in Table 3.1. Let us create a master flat frame to be used for the reduction of the science frame `susi0100.bdf`. Suppose the Association Table contains the name of the master flat to be created: `susi_140_142_143.bdf`. Hence, three flats have to be combined, namely

Keyword	Content	Default	Description
CCD_IN	input name	?	table of input frames
FF_TYP	descriptor value	*	exposure type to check, if any
FF_SIG	yes|no	no	create sigma image
FF_MET	comb. method	avsigcl	type of combining operation
FF_DEL	yes|no	no	delete cal. frames afterwards
FF_STA	mean|median|mode	mode	correct. by Mode/Median/Mean
FF_EXP	yes|no	yes	scale by exposure time
FF_SCA	yes|no	no	scale by MMM
FF_OFF	yes|no	no	add offset obtained from MMM
FF_WEI	yes|no	yes	use a weighted average
FF_SEC	area	[<,<:>,>]	area for computing MMM
FF_RAN	real,real	-99999,99999	valid pixel range
FF_CLP	real	3.,3.	low/high sigma clipping factor
FF_NUL	real	NULL(2)	value for rejections

Table 3.6: Keywords for combining flat field calibration frames

`susi0140.bdf`, `susi0142.bdf`, and `susi0143.bdf`. To combine these frames into the master flat field we enter:

```
SELECT/TABLE asso_tbl
COMBINE/CCD FF asso_tbl
```

The command will go the Association Table, checks the output name for the flat to be created for science frame `susi0100`, makes a list of single flats to be combined and does the combining. The output on the screen and stored in the MIDAS log file will look like (`VERBOSE=YES`):

```
Combining FLAT frames: Input=asso_tbl.tbl; output=susi_140_142_143.bdf

Statistics of frame no. 0:
area [@60,@10:@1070,@1020] of frame
minimum, maximum:                  5.560000e+02   8.654000e+03
mean, standard_deviation:          6.101316e+03   1.913127e+02

Statistics of frame no. 1:
area [@60,@10:@1070,@1020] of frame
minimum, maximum:                  8.390000e+02   1.276900e+04
mean, standard_deviation:          8.313817e+03   2.569877e+02

Statistics of frame no. 2:
area [@60,@10:@1070,@1020] of frame
minimum, maximum:                  4.220000e+02   1.046500e+04
```

Keyword	Content	Default	Description
CCD_IN	input name	?	table of input frames
SK_TYP	descriptor value	*	exposure type to check, if any
SK_SIG	yes|no	no	create sigma image
SK_MET	comb. method	avsigcl	type of combining operation
SK_DEL	yes|no	no	delete cal. frames afterwards
SK_STA	mean|median|mode	mode	correct. by Mode/Median/Mean
SK_EXP	yes|no	yes	scale by exposure time
SK_SCA	yes|no	no	scale by MMM
SK_OFF	yes|no	no	add offset obtained from MMM
SK_WEI	yes|no	yes	use a weighted average
SK_SEC	area	[<,<:>,>]	area for computing MMM
SK_RAN	real,real	-99999,99999	valid pixel range
SK_CLP	real	3.,3.	low/high sigma clipping factor
SK_NUL	real	NULL(2)	value for rejections

Table 3.7: Keywords for combining sky calibration frames

```
mean, standard_deviation:          5.622723e+03   1.722370e+02

Method=avsigclip, low=  0.00, high=  0.00
lowclip=  3.00, highclip=  3.00
    frame #        Ncomb   Exp_time   Mode    Scale    Offset    Weight
susi0140.bdf         1       1.00 571.878   1.093        -0    0.333
susi0142.bdf         1       5.00 862.392   0.725        -0    0.333
susi0143.bdf         1       5.00 441.692   1.416        -0    0.333
--------------------------------------------------------------------
Statistics of output frame susi_140_142_143.bdf:
area [@60,@10:@1070,@1020] of frame
minimum, maximum:                  6.045866e+02   1.023942e+04
mean, standard_deviation:          6.886717e+03   2.060752e+02

                  Ncomb   Exp_time   Mean     Mode  N_undef
susi0100_ff          3      3.933 6886.72 623.478        0
Undefined pixels, set to 'null value' (=     0.000000)
--------------------------------------------------------------------
```

3.9 Processing the Data

Having obtained some background, let us now start to process the input frames according to the scheme in Section 3.7. As indicated in that section the reduction sequence consists, in a simplified version, of the following steps:

1. determine overscan correction and trim area;
2. preparing calibration frames;
3. do the corrections;
4. apply the bad pixel corrections.

3.9.1 How the Data is Processed

Before any processing on the input frame is done, the `REDUCE/CCD` command will first collect all resources needed for the calibration of the science frame(s). These include the master calibration frames, the overscan offset, and scaling parameters. So, at this point no operations are done yet. This is done for efficiency reasons: all standard calibration arithmetic on the input frame is done in one go. As an example, suppose the science frame is supposed to be corrected for dark current and to be flat fielded. From the Association Table the command first identifies the names of the master dark and flat frames, and checks for their existence. If they are not present they will be created. Next, the master dark and flat field will be checked on their processing status. If they have not been processed yet, that will first be do by another (recursive) run of the `REDUCE/CCD` command. In this second run also the scaling factors (*i.e.* exposure times and the mean of the flat field frame) will be determined.

The standard calibration operation is done by a large `COMPUTE/IMAGE` with the following input:

$$out = (in - scan - biasfrm - darkscale * darkfrm) * flatscale/flatfrm \tag{3.11}$$

Here, out is the output frame, in is the input frame, $scan$ is the overscan bias value or frame, $biasfrm$ is the master bias, $darkfrm$ and $darkscale$ are the master dark frame and scaling factor, and $flatfrm$ and $flatscale$ are the master flat field and the mean value of the flat field. In the `COMPUTE` $biasfrm$ and $darkfrm$ can also be constants.

3.9.2 Running `REDUCE/CCD`

The command `REDUCE/CCD` can process one or more science frames automatically provided that:

- the association table correctly describes the associations between the science frames and the calibration frames;
- the CCD reduction table or the `SC_...` keywords contains the correct names of the master calibration frames;
- the keywords for creating the calibration frames are correctly set.

The standard default calibration procedure involves the following processing: overscan subtraction, trimming, bias and dark subtraction, and flat fielding. The keywords to be set for controlling these options are listed in Table 3.3. Below we describe how the command will try to correct your data.

REDUCE/CCD will first identify the Association Table. From this table it gets the names of the input science frames. Next, it will try to find the master calibration frames listed in the table, and will create these if they do not exist yet. After having collected all required calibration data the input frame is checked for its exposure type. Obviously, a dark frame should not be flat fielded and hence the processing options for the dark exposures should be set differently than for the science frames. After having determined the processing options the actual processing starts.

Overscan correction and trimming

If the keyword SC_SCAN is set the first action will be to determine the bias offset. For that you can use the overscan region on the chip. This possibility is certainly helpful in case you did not obtain separate bias calibration frames or in case you want to determine the bias offset for each science frame individually.

At initialization, keywords are created to contain the overscan and image areas: OV_SEC and IM_SEC, respectively. However they are not filled yet. You can find these numbers using a number of standard MIDAS core commands to display one or more frames, or to plot a number of lines or columns. Then use the command SET/CCD to fill in or to adjust the keywords.

The basic command that does the overscan fitting is called OVERSCAN/CCD. It needs at least two input parameters: the input and output frame. A third parameter, the overscan area, may be added. If absent it will be taken from the CCD keyword OV_SEC. To determine the bias offset a number of CCD keywords, all starting with 'OV_' will be read, and hence have to be filled correctly. They are displayed in the Table 3.8

Keyword	**Options**	**Default**	**Description**
OV_SEC	coord.	?	area to be used for fitting
OV_IMODE	yes|no	yes	interactive fitting?
OV_FUNCT	lin|pol	lin	type of function to fit
OV_ORDER	integer	3	order of fit (polyn. fit only)
OV_AVER	integer	1	number of points to combine
OV_ITER	integer	1	max. number of iterations
OV_REJEC	real	3	low/high rejection factor

Table 3.8: CCD keywords for overscan fitting

Depending on the readout direction of the CCD (keyword DIRECT) the command will average the data in the overscan region, and fit these values as a function of line-number (*i.e.* average in the 'x' direction within the overscan region, and fit these as function of 'y'). The fit will be subtracted from each column (row) in your frame. In case of a zero order fit the correction is a simple constant. In the other case it will be one dimensional array.

After an interactive fit has been completed the command will ask you if you want to fit the next frame interactively as well, or if you want to apply the overscan correction to all subsequent frames. In the latter case the overscan correction will be stored as a separate calibration frame or, in case a zero order fit has been used, a single constant.

After the overscan correction is determined, the frames can be trimmed to keep only those parts of the frames that contain valid data. Obviously, this action will speed up the data processing of the subsequent reduction steps. The relevant keyword is `SC_TRIM` (default value `yes`) and `IM_SEC`, containing the data section of the frame.

Bias, Dark and Flat Field Correction

This part has already been discussed in Section 3.8. Depending on the keywords `SC_BSCOR`, `SC_DKCOR`, and `SC_FFCOR` the calibration procedure checks for the availability of the master bias, dark, and flat field calibration frames. If they can not be found they will be created using the input frames listed in the Association Table, and according to the keyword setting for combining.

The next step after having obtained the calibration frames is to process these frames. For example, after the flat is created/found it will be checked for its processing status. If it has not been processed any on-going calibration (*i.e.* of a science frame) will be interrupted, and the flat field will be processed. After this has been completed processing of the science frame will continue.

3.10 Additional Processing

3.10.1 Sky Illumination Corrections

The flat field calibration frame may not have the same illumination pattern as the observations of the sky. In this case when the field field correction is applied to the sky data, instead of being flat there will be gradients in the background. You can check this by plotting several lines over the reduced sky frame. In case of no clear variation you can continue with the correction of your science frame(s) using the standard flat. In some cases the application of the simple flat field correction does not do a good job, and there may be an illumination problem. If such deviations are present one can try to correct for these by applying an illumination correction.

The illumination correction is made by smoothing the reduced blank sky frame heavily. The illumination frame is then divided into the frames during processing to correct for the illumination difference between the flat field and the objects. Like the flat field frames, the illumination correction frames may be data set dependent and hence there should be an illumination frame for each data set.

The smoothing algorithm is a moving average over a two dimensional box. The algorithm uses a box size that is not fixed. The box size is increased from the specified minimum at the edges to the maximum in the middle of the frame. This permits a better estimate of the background at the edges, while retaining the very large scale smoothing in the center of the frame. Other tools in MIDAS can also be used for smoothing, but this may need more of the user and may take more processing time.

Blank sky frames may not be completely blank, so a sigma clipping algorithm may be used to detect and exclude objects from the illumination pattern. This is done by computing the rms of the frame lines relative to the smoothed background and excluding points exceeding the specified threshold factors times the rms. This is done before each frame line is added to the moving average, except for the first few lines where an iterative process is used. If this approach is not successful manual removal of objects (stars) is required.

Keyword	**Value**	**Description**
`CCD_IN`	name	input sky frame
`IL_TYP`	SKY	exposure type
`IL_XBOX`	5.0,0.25	smoothing box in x
`IL_YBOX`	5.0,0.25	smoothing box in y
`IL_CLIP`	yes	clipping the input pixels
`IL_SIGMA`	2.5,2.5	low and high clipping sigma

Table 3.9: Keywords for making the illumination frame

Both in the pipe line reduction and in the manual reduction the illumination corrections to the science frames will be done provided the keyword `SC_ILCOR` is set to 'yes'. In both cases it is assumed that the illumination frames are available. In addition, in the case of pipe line processing the names of the illumination corrections must be stored in the Association Table. If an illumination correction frame is absent an error message will be issued and the illumination correction for the associated science frame will not be done.

3.10.2 Creation of Sky Correction Frames

For creation of one or more sky illumination correction frames the command (`SKYCOR/CCD`) is available. Similar to other corrections, sky frames will first be processed according to the processing keywords (`SC_SCAN`, `SC_TRIM`, `SC_PXFIX`, `SC_BSCOR`, `SC_DKCOR`, `SC_FFCOR`). After this calibration step, the command will smooth the reduced sky frame(s) to create the final sky illumination frame(s).

Input for the command can be either a single sky frame, or the Association Table containing a column with the names of the master sky frames. In the first case the output is, obviously, a single illumination frame. In case the Association Table is given as input the name of the output frame names will be the names of the input frames extended with '`_ill`'. In addition, the illumination frames will be stored in an illumination column in the Association Table. This column can be used in the pipe line processing of the science frame. Default input is taken for the keyword `CCD_IN`. Smoothing parameters are taken from the the `IL_...` keywords, which are listed in Table 3.9.

After the illumination frame has been created, one can multiply the original flat field by the illumination frame, resulting in an adjusted flat field. This approach clearly has the advantage of speeding up the calibration process since it requires one calibration frame and two computations (scaling and dividing the illumination correction) less. The output frame is called sky flat. It is the flat field that has been corrected to yield a flat sky when applied to the observations.

Having done this, this sky flat can be used as the final one. How good this new flat field is can be checked by correcting the blank sky once more, using this sky flat. If the result is not satisfactory one can try to play with the smoothing parameters, or else ask for help of an experienced observer.

To create the sky flat fields the command `SKYFLAT/CCD` is available. As input it takes the blank master sky frame, processes it if needed, and creates the output sky flat from the smoothed processed sky and the appropriate flat field. The smoothing parameters are stored in the keywords displayed in Table 3.9. The way the command works is identical to the command `SKYCOR/CCD`. The default names of output sky flats are the same of the original master flat field frames. Hence, the input master flat fields, use to flat the master sky frame, will be overwritten.

3.10.3 Illumination Corrected Flat Fields

A second method to account for illumination problems in the flat fields is to remove the large scale pattern from the flat field itself. This is useful if there are no reasonable blank sky calibration frames and the astronomical exposures are evenly illuminated but the flat fields are not. This is done by smoothing the flat field frames instead of blank sky frames. As with using the sky frames there are two methods, creating an illumination correction to be applied as a separate step or fixing the original flat field. The smoothing algorithm is the same as that used in the other illumination commands. The commands to make these types of corrections are `ILLCOR/CCD` and `ILLFLAT/CCD`. The usage is virtually identical to the previous sky illumination correction commands. Obviously, after having obtained the illumination corrected flat field it is reasonable to replace the original flat fields by the corrected flat fields in the reduction table. Like is the case of `SKYCOR/CCD`, by default the command `ILLFLAT/CCD` will replace the original flat field by the illumination corrected one.

3.10.4 Fringe correction

The fringe correction should be made from regions of really empty sky. For that the sky frames should be combined such that cosmic rays and faint stars are eliminated. Hereafter, by smoothing the frame determine the average intensity level and subtract this value for the frame. After all objects have been removed from this sky frame the frame is essentially zero except for the fringe pattern. The frame is scaled to the same exposure time as the science frame and then subtracted. Because the night sky lines are variable, matching the fringe amplitude to the one in the science frame may not be as straightforward as expected, but should be possible with robust estimation.

3.10.5 Bad Pixel Correction

The CCD package includes possibilities, before doing the actual calibration step(s), to correct for bad pixels in the CCD calibration frames. This correction is applied via the command FIXPIX/CCD and based on existing MODIFY commands. Except for the input and output frame the FIXPIX/CCD command accepts two more parameters in the command line: the correction method to be applied and the MIDAS table containing the bad pixel(s) of bad pixel areas(s). Default value for these two parameters are stored in the keywords FX_METH and FX_FILE. In addition to these keywords three other keywords exist to control the correction procedure: FX_FACT, FX_FPAR, and FX_NOISE. The use of these keywords depends on the correction method applied.

Currently, the required format of the table file depends on the correction method to be applied. Below you can find a listing of the method available and with FX_ keywords involved.

- method area (MIDAS commands MODIFY/AREA):
  parameter degree taken from keyword FX_FPAR(1);
  parameter constant=0.
- method pixel (MIDAS command MODIFY/PIX):
  parameter arfacts taken from keyword FX_FACT;
  parameter xdeg,ydeg,niter taken from keyword FX_FPAR;
  parameter noise taken from keyword FX_NOISE.
- method column (MIDAS command MODIFY/COLUMN):
  parameter col_type=V.
- method row (MIDAS command MODIFY/ROW):
  parameter row_type=V.

3.11 Mosaicing of Frames

In particular in the IR range of the spectrum but also in the optical the user may want to combine several reduced science frames into a mosaic. To support this, the CCDRED context contains five mosaicing commands.

In order to align or match a number of images the user first has to run the command CREATE/MOSAIC. This command creates a master frame containing all input frames to be aligned and intensity matched as sub rasters. The order in which the sub rasters are put in the mosaic image is determined by a number of keywords all starting with 'MO_'. *E.g.* the keyword MO_CORN determines the origin of the mosaic. Apart from the mosaic image, CREATE/MOSAIC also creates a MIDAS table containing all relevant information about the subrasters in the mosaic image and the ways it was created.

Using the mosaic frame and the database table, one can now do the alignment or alignment and matching of the sub raster. The alignment can be done by the command ALIGN/MOSAIC, the matching by two commands: MATCH/MOSAIC and FIT/MOSAIC. To be successful these commands need additional information were the images have to be glued together. The commands `ALIGN/MOSAIC` and `MATCH/MOSAIC` accept that information in three different formats: a fixed shift, a table with the x and y shifts for each subraster, or a table of common objects. The command `FIT/MOSAIC` only needs the database table that should contain the shifts per subraster.

The command `SHIFT/MOSAIC` is created in order to support the selection of common objects in adjacent subrasters. This command can be used to create a MIDAS table containing these objects and that can be used as input for the commands `ALIGN/MOSAIC` and `MATCH/MOSAIC`.

The keyword setting for the mosaicing can be inspected by the command `SHOW/CCD MO`. Changes can be done by the `SET/CCD` command. Below follows an overview of the mosaicing keywords and their default settings. The help files of the five commands give the full details.

Keyword	Options	Default	Description
MO_SEC	coord.	[<,<:>,>]	section for statistics
MO_SUBT	yes|no	no	subtraction option?
MO_CORN	ll|lr|ul|ur	ll	type of function to fit
MO_DIREC	row|column	row	add subraster row/column wise
MO_RAST	yes|no	no	add in raster pattern
MO_OVER	integer	1,1	space between adj. frames
MO_INTER	interpolant	lin	interpolation method
MO_MNPX	integer	0	minimal number of pixels
MO_TRIM	integer	1,1,1,1	trim columns and rows
MO_NUL	real	NULL(2)	value for rejections

Table 3.10: CCD keywords for mosaicing

3.12 Miscellaneous

Besides the standard product described in the previous sections, a number of additional commands is implemented or will be considered and implemented later. E.g. within MIDAS a command to remove cosmic rays already exists. Hence, this command can be implemented in the standard CCD pipeline procedure.

After the data have been calibrated, one might need to combine a number of science frames into a combined frame. A tool for doing this should correct for possible differences in sky background, exposure time, positions, etc. In a previous version of the CCD context such a tool did exist. We might consider porting this tool to the new CCD context.

3.13 Commands in the CCD package

Below follows a brief summary of the CCD commands and parameters for reference. Except for the already existing command `FILTER/COSMIC`, the commands in Table 3.11 are initialized by enabling the CCD context with the MIDAS command `SET/CONTEXT CCDRED`. Parameters will generally obtained from the CCD keywords; however some of the keyword settings can be overruled by specifying them on the command line.

HELP/CCD	[keyword]
	provide help information for a CCD keyword
INITIAL/CCD	[name]
	initialize the ccd package
LOAD/CCD	[instr]
	load telescope/instrument specific default
SET/CCD	keyw=value [...]
	set the CCD keywords
SAVE/CCD	name
	save the keyword settings
SHOW/CCD	[subject]
	show the CCD keyword setup

Table 3.11: CCD commands

REDUCE/CCD	[in_spec] [out_frm] [proc_opt]
	(partial) calibration of one or more science frames
COMBINE/CCD	exp_type [in_spec] [out_fram]
	combine CCD frames using catalogue input
OVERSCAN/CCD	in_fram out_fram [sc_area] [mode]
	fit bias offset in the overscan region and correct
TRIM/CCD	in_fram out_fram [im_sec] [del_flg]
	extract the useful data fro the ccd frame
FIXPIX/CCD	in_fram out_fram [fix_file] [fix_meth]
	correct bad pixels using a pixel file
BIAS/CCD	in_fram out_fram bs_fram
	correct input frame for the additive bias offset
DARK/CCD	in_fram out_fram dk_fram
	correct the input frame for additive dark offset
FLAT/CCD	in_fram out_fram ff_fram
	do a flat fielding of the input frame
ILLUM/CCD	in_fram out_fram il_fram]
	do an illumination correction of the input frame
FRINGE/CCD	in_fram out_fram fr_fram
	do a fringe correction of the input frame
FRCORR/CCD	[in_spec] [out_spec] [xboxmn,xboxmx] [yboxmn,yboxmx] [clip] [losig,hisig]
	make fringe frame
ILLCORR/CCD	[in_spec] [out_fram] [xboxmn,xboxmx] [yboxmn,yboxmx] [clip] [losig,hisig]
	make flat field illumination correction frame(s) (TBD)
ILLFLAT/CCD	[in_spec] [out_fram] [xboxmn,xboxmx] [yboxmn,yboxmx] [clip] [losig,hisig]
	make illumination corrected flat field frames (TBD)
SKYCORR/CCD	[in_spec] [out_fram] [xboxmn,xboxmx] [yboxmn,yboxmx] [clip] [losig,hisig]
	make sky illumination correction frame(s)
SKYFLAT/CCD	[in_spec] [out_fram] [xboxmn,xboxmx] [yboxmn,yboxmx] [clip] [losig,hisig]
	apply sky observation to flat field
ALIGN/MOSAIC	in_frm in_tab out_frm method,data [xref,yref] [xoff,yoff] [x_size,y_size]
	align the elements of the mosaiced frame
CREATE/MOSAIC	in_cat out_frm out_tab nx_sub,ny_sub
	mosaic a set of (infrared) ccd frames
FIT/MOSAIC	in_frm in_msk in_tab out_frm [match] [nxrsub,nyrsub] [xref,yref] [x_size,y_size]
	align and match the elements of the mosaiced frame
MATCH/MOSAIC	in_frm in_tab out_frm method,data [match] [xref,yref] [xoff,yoff] [x_size,y_size]
	aign and match the elements of the mosaiced frame
SHIFT/MOSAIC	out_tab [curs_opt] [csx,csy] [clear_opt]
	get x and y shifts of subrasters in the mosaiced frame

Table 3.12: CCD command continued

Chapter 4

Object Search and Classification

This chapter describes the use of commands which produce an inventory of astronomical objects present in an analysed two dimensional image. The end product is a list of objects classified as *stars* or *galaxies* and containing also parameters such as various kinds of magnitudes, sizes, some characteristics of image shapes, and optionally also cleaned one–dimensional image profiles. These commands taken together constitute what is known as the `INVENTORY` program. They belong to the `INVENT` context which can be activated with the help of the command: `SET/CONTEXT INVENT`.

The programs described here were developed by A. Kruszewski during several visits to ESO in the last years. Most of the documentation below was written by him and has only slightly been modified since then. Users are kindly asked to refer all detected inconsistencies of the implementation of the package in MIDAS to the ESO–Image Processing Group.

***** IMPORTANT NOTE *****

The default MIDAS implementation of the INVENTORY allows for a maximum 16384 objects in SEARCH and ANALYSE commands and 32768 detections in the SEARCH command. The commands will issue a warning message if these bounderies are exceeded. On the other hand, since INVENTORY only uses fixed arrays sizes, these numbers can be too large on small memory-sized systems. In case of problems the parameters can be modified in the INVENTORY include file. They are called located MAXCNT and MAXDET, respectively, and and are located in the file `/midas/$MIDVERS/contrib/invent/incl/invent.inc`. If you encounter problems with these settings please consult your local MIDAS responsible for adjusting them.

4.1 General Information

INVENTORY has been originally designed as a medium speed and medium accuracy universal program for finding, classifying, and investigating astronomical objects on two–dimensional image frames in a way that is as automatic as possible. In a course of further development, both speed of execution and attainable precision have been improved, and the package differs from other related programs mainly by its tendency to minimise the amount of time spent by a user on interactive work at the terminal. Though it can be used for most of the relevant applications, `INVENTORY` is best suited for analysing numerous similar frames, like in cases of surveys or variable stars observations. It is also convenient for a first look preview of the material that will be analysed with more elaborate methods. It is not recommended for users who are not willing to give up the possibility of interactive control.

A major requirement imposed on `INVENTORY` was that it should be able to classify detected objects into *stars*, *galaxies,* and *image defects.*

Note

While it is possible to run `INVENTORY` *for the first time without many preparations, achieving good results requires some experience. The program runs in an automatic mode when some parameters are set to proper values. Different applications may require different values for these parameters. Using* `INVENTORY` *with a new kind of material requires some preparation and trial runs are necessary for adjusting the parameters. The time spent on tuning up the program is well paid off when there is a sufficiently large number of objects and/or frames so that the use of other techniques would be much more time consuming. The tuning up may be difficult in a crowded field with many overlapping objects, or in a case where there are big bright galaxies. On the other hand the use of* `INVENTORY` *should be relatively easy when the investigated field is populated mainly by not too densely packed stars.*

These photometric packages which aspire to high accuracy of results have to solve the problem of deblending of overlaping images. In its base mode `INVENTORY` does it in a fast but approximate way. A somewhat more accurate deblending of stellar objects using one–dimensional point spread function is now optional. Two dimensional point spread function fit, its extension to strongly undersampled images, and more precise deblending of galaxies are in preparation.

The program is functionally divided into three discrete steps. It will be possible to perform these steps using the commands: `SEARCH/INV`, `ANALYSE/INV`, and `CLASSIFY/INV`. The additional commands `SHOW/INV` and `SET/INV` serve for displaying and updating values of the `INVENTORY` keywords.

Before using `INVENTORY` commands it is necessary to give command `SET/CONTEXT INVENT`. It may be convenient to include this command into the `login.prg` file.

The (optional) `SEARCH/INV` command prepares a preliminary list of objects. The `SEARCH/INV` can be omitted once we have a list of objects with accurate positions in a MIDAS table format.

The `ANALYSE/INV` command evaluates and updates an input list of objects. It can be used in a `VERIFY` or `NOVERIFY` mode. In `VERIFY` mode, the `ANALYSE/INV` command verifies the used table of objects. Some entries are deleted but usually a larger number of new ones are added. The objects positions are improved. In `NOVERIFY` mode this verification process is omitted. In both modes the `ANALYSE/INV` command calculates several image parameters, which can be used as final results and/or as input to the `CLASSIFY/INV` command.

The `CLASSIFY/INV` command uses the output table produced by the `ANALYSE/INV` command for dividing the objects into *stars, galaxies* and *spurious objects.* It accepts only input of MIDAS table files that have been produced by `ANALYSE/INV`.

4.2 What Data Frames can be Used?

Practically all kinds of astronomical images, containing a number of stars and/or galaxies, can be treated with the `INVENTORY` commands. Up to now `INVENTORY` has been used or tested on data from the following instruments: Schmidt plates, CCD–frames taken with several telescopes, direct unaided and electronographic plates taken with the ESO 3.6m telescope.

The applicability of `INVENTORY` commands is limited to frames that contain more than say 100 and less than 8000 objects. The lower limit concerns a single frame. If one has a number of frames each containing 20 or even a smaller number of objects, it is still worthwhile to use `INVENTORY`. A `CLASSIFY/INV` command requires:

- at least 20 stars and 20 galaxies to work at all, and
- at least 100 objects on a good clean CCD–frame, or
- at least 300 objects on a Schmidt frame

in order to work well. The upper limit is set by the dimensions of some internal arrays. Frames containing more than 8000 objects should be divided onto smaller subframes to be run separately.

Note

The CPU time required depends approximately linearly on the total number of pixels and on the total number of found objects (including defects). The demo frame with little over 100000 pixels and 200 objects needs 12.4, 10.7, and 2.5 sec of CPU time for executing the `SEARCH/INV` *command, the base version of* `ANALYSE/INV` *command, and the* `CLASSIFY/INV` *command respectively with the use of the ESO VAX 8600 computer.*

4.3 Procedures to Follow

The user is advised to start working with this new kind of observational material by applying commands to small regions of an investigated frame in order to properly tune control parameters. This should help to obtain good fast results and in effect could also save a lot of time.

4.3.1 Preparing Data Frames

Data frames to be run with `INVENTORY` commands should be *clean*. It is desirable that they do not contain any dark spots situated close to objects. This could result in an unpredictable outcome! Bright spots are less dangerous. They are partly detected by the `SEARCH/INV` command, but the `CLASSIFY/INV` command usually classifies them as defects, unless they look like galaxies. Unwanted frame defects can be removed using the `MODIFY/PIXEL` command. Frames that have not been properly cleaned or even frames which have not been cleaned at all can also be used but the user should in this case check visually which objects may be affected by defects.

Irregular or sloping background is not a problem. One has only to know if background variations are related to changes in sensitivity or are caused by some additive factor. Depending on what is the case, the keyword `FIELDVAR` should be set to 0 (sensitivity changes) or to 1 (additive factor - default).

Unused parts of CCD–frames should be removed with the help of the `EXTRACT/CCD` command. There is a problem when valid data is on a circular image. In this case it is convenient to set the unused part of a frame to the value close to the average background value or to the value of low cut. This can be done with the help of a properly shaped mask. The same holds true for other cases of non–rectangular images.

When more than one frame of the same field is analysed, it is convenient to align all frames to common physical coordinates. This can be done by using commands such as `CENTER`, `ALIGN`, or `REBIN`.

When a user plans to manually select the standard stars for point spread function determination, he should identify these stars by means of MIDAS command `GET/CURSOR STARS/DES` before using the `ANALYSE/INV` command.

4.3.2 Setting the Low and High Cuts

Great attention should be paid to proper settings of the low and high cuts. They were previously entered as a parameter of the `NORMALIZE/IMA` command. Now only the original frame is used and it is not truncated by the applied cuts. The program needs proper values of cuts in order to avoid using incorrect data. In order to find acceptable values of cuts one has to redisplay an image several times with varying values of display cuts. One can, in this way, fix low cut as a level just below all the good data with only bad pixels having smaller values. High cut should be placed below the saturation level. The data may be distorted even at some distance from the saturation level and the program tries to take this into account. Therefore, in a case where none of the frame stars are saturated the high cut can take any value, that is to say, at least twice as large as the highest pixel value for any stellar object.

As the frames need not be normalised, low and high cuts should be entered in the same units as pixel data.

4.3.3 Setting the Keywords used by SEARCH/INV Command

Besides LHCUTS there are a number of other keywords used by the `SEARCH/INV` command that can be set using the `SET/INV S` command. Each of them will be discussed here in turn.

TRESHOLD — This keyword sets a limit on central brightness of objects to be found. It is expressed in measurement units. It is recommended *to set a rather high value at the beginning*, then execute `SEARCH/INV` and inspect an output frame on the image display with the detected objects marked by means of the `LOAD/TAB` command. A few tries with decreasing values of TRESHOLD should be enough to find a correct value. Using too low a value of the limiting threshold may result in an abnormally long execution time.

HALFEDGE — This keyword determines a size of a subarray that is used for calculating local sky background level. That initial local background is used for detection purposes only. Therefore such a subarray should be a few times larger than visible sizes of faintest or most common objects. The value of HALFEDGE corresponds to a number of pixels in horizontal or vertical direction between the central pixel and an edge of a subarray. The resulting dimension of the subarray is $(2 \times \mathrm{HALFEDGE} + 1)^2$. For example if $\mathrm{HALFEDGE} = 10$ then the subarray size is 21×21 pixels. The allowed range of values for HALFEDGE is from 4 to 50.

PAIRSPRT — This keyword controls the minimum separation of detected double objects. It gives the smallest allowed separation in pixels between two catalogued objects. It can be set with the help of inspecting examples of double objects in an analysed frame on the zoomed image display. It cannot be smaller than 2.

MULTDTCT — This keyword is similar to PAIRSPRT but plays a different role. It is used only in the `SEARCH/INV` routine when combining multiple detections of the same object. *Usually it is best to have the values of PAIRSPRT and MULTDTCT close to each other*, but in some cases (e.g. when there is a prominent spiral galaxy in a frame) it is useful to have MULTDTCT two or three times larger than PAIRSPRT. At the beginning, it may be set equal to PAIRSPRT, and increased when there are still multiple detections of a single object in an output from the `ANALYSE/INV` command.

BRGTCTRL — Many spurious faint objects are detected around very bright ones when this keyword is set to 0. Increasing it helps to eliminate spurious objects. Too large a value may result in disapearance of some real objects.

NETHEDGE — Half edge of the regions used to determine preliminary sky background in a net of regions.

SKYDETER — Sets accuracy of the sky background determination.

FILTER — Sets slevel of bright pixel filtering.

MARGINS — Sets the width in pixels of frame margins where no object detection is made. Previously it was identical with HALFEDGE.

The command `SET/INV S` concerns only the most often changed keywords. All the keywords connected with the searching routine can be updated with the help of the command `SET/INV S A`. Here are the additional keywords:

PHYSICAL — Tells if searched area is defined by physical coordinates.

IJBORDER — Sets limits of the investigated area.

XYBORDER — Limits of investigated area in physical units.

CTRLMODE — Controls calculation of sky background. The modal value of the pixel distribution is returned when this keyword is set to 3.0 (default). Repetitively clipped mean is returned if it is set to 0.0.

CVFACTOR — Controls iterations at calculating sky background.

SGFACTOR — Clipping factor at calculating sky background.

ELONGLMT — When several detections have been joined into a single object, then such configuration of detections is tested for elongation and this keyword sets a limiting elongation beyond which an object in question is tested for duplicity.

YFACTOR — The test for duplicity is based on a scan along the major axis of detections configuration, in a rectangle with the ratio of its sides set by this keyword.

DFVALLEY — This sets how deep a minimum must be present in such a scan for assumming object duplicity.

4.3.4 Executing the ANALYSE/INV Command

The `ANALYSE` command offers a number of options for its execution:

In default VERIFY mode the `ANALYSE/INV` command assumes that the input table has relatively inaccurate identifications and positions of objects as in the case of the output from the `SEARCH/INV` command. Consequently it will verify the reality of each object and will improve its position. Many objects will be removed from the list and some others will be added.

In NOVERIFY mode this verification process will be omitted. In `NOVERIFY` mode the program assumes that columns with labels `X` and `Y` contain precise object x and y coordinates.

Use of preset isophotal radia. It is also possible to read preset sizes of an isophotal radius (in pixels)from an input table for each object. In order to use this facility one has to set the keyword ISOPHOTR to 1 and ensure that the input table column with isophotal radii has the label `:RAD_ISO`.

Creating an output table is activated by giving its name in the command line.

4.3.5 Setting the Keywords used by the ANALYSE Command

Several keywords can be set with the help of the `SET/INV A` command. This command also displays some keywords that have already been used by the `SEARCH/INV` command. For your convenience these keywords are the same as those used in `SEARCH/INV`. Below the additional keywords are listed as well as those already described above:

LHCUTS — See `SEARCH/INV`

THRESHOLD — See `SEARCH/INV`

HALFEDGE — See `SEARCH/INV`

PAIRSPRT — See `SEARCH/INV`

ANALITER — Default value of this keyword equal to 0 corresponds to the case that only the base version of the command `ANALYSE/INV` is used. Any positive value sets the number of additional iterations. In each of these iterations the contributions from stellar neighbours are subtracted using the empirical one-dimensional point spread function. In order to obtain error estimates in the output table (columns 24 to 26) the value of this keyword should be greater than 0. In addition, the extend of the 2-dim psf has to be defined (keyword FULLPSF, see below).

PRFLCTRL — Number of central points of the Point Spread Function that are adjusted by the program. It is done automatically when a positive number is given. The program assumes the existance of the descriptor STARS in the input image frame with positions of standard stars when a negative number is given.

CLASSPAR — Maximum absolute value of relative gradient for considering the object as a star.

ZEROMAGN — Zero point of magnitudes. Should be set in accordance with the value of the keyword EXPRTIME.

EXPRTIME — The units of the exposure time should be consistent with the value of the keyword ZEROMAGN. May be set to 1 if the knowledge of exposure time is irrelevant

STMETRIC — Radii in pixels of two concentric circular apertures used to measure aperture magnitudes. The second of these circular apertures defines also an area over which the convolved magnitudes are calculated.

FILTER — See `SEARCH/INV`

UNITS — Switch between background units (0) and instrumental units (1)

MARGINS — See `SEARCH/INV`

Additional keywords are displayed for updating after giving the command `SET/INV A A`. Only those that have not been described already will be listed here.

FULLPSF — Specifies the extend in pixels of 2-dimensional p.s.f.

UNDRSMPL — Specifies the undersampling factor.

OUPROFIL — Specifies a number of pixel spaced points of one–dimensional intensity profile of objects to be recorded in output table.

PHYSICAL — See `SEARCH/INV`

IJBORDER — See `SEARCH/INV`

XYBORDER — See `SEARCH/INV`

BRIGHTOB — Helps to remove faint spurious objects close to bright ones. It may distort central parts of objects, therefore it should be disabled by setting it to 0 when photometric accuracy is at issue. Default is 0.

PRFLCLNG — Parameters used for profile cleaning.

ISOPHOTR — If set to 1 then the program expects to find isophotal radii of all objects in a column of an input table with label RAD_ISO. Relevant only in `NOVERIFY` option. Default is 0.

ETAFUNCT — Petrosian η–function. Used in measuring the Petrosian magnitude and radius.

NEBLIMIT — Relevant for dense configurations of galaxies. Prevents, when set to a positive value smaller than unity, faint galaxies from obtaining abnormally large sizes. Should be set to zero (default) in most other applications.

SPROFIL1 – SPROFIL5 — First 25 points of the initial one–dimensional Point Spread Function.

4.3.6 Helpful Hints

The keywords EXPRTIME and ZEROMAGN set the zero point for magnitudes. An independent calibration is necessary for adjusting them properly.

The keywords STMETRIC and ETAFUNCT should be set whenever one wants to use the corresponding aperture or Petrosian magnitudes.

Handling the keywords SPROFIL1 – SPROFIL5 and PRFLCTRL requires some care. The program automatically determines a Point Spread Function when it gets proper initial values from the keywords SPROFIL. Initial values may be wrong by 0.1 each, and the program still can converge to the right Point Spread Function. In the case of rich stellar fields the automatic Point Spread Function determination usually gives excellent results even with drastically wrong initial values. However in frames which contain more galaxies than stars, the automatic procedure tends to return an average profile of galaxies rather than the Point Spread Function. The same may happen when the initial values are too small for fields with a moderate number of galaxies.

In order to check whether the Point Spread Function is correct, it is possible to use the MIDAS `PLOT/TABLE commands` to plot the dependence of the relative gradient on isophotal magnitude. When the Point Spread Function is correct, the stars will cluster around a (relative gradient = 0) line. If not, then the stellar sequence is shifted, usually upwards, and most often it is not longer linear. The value of the average shift of the stellar sequence should then be added to the first few points of obtained Point Spread Function, which is displayed on the terminal screen and written into the image frame descriptor DPROFILE.

If there are difficulties in finding the initial Point Spread Function one can use the manual mode for choosing objects to determine the Point Spread Function. The keyword PRFLCTRL should then be set to a negative value, and an input frame should contain the descriptor STARS with standard star coordinates. The descriptor STARS with positions for up to 200 stars is produced as follows:

Get an input image frame onto the image display screen.

- Set cursor box: cursors on, `Track` off, `Rate` on.
- Type: `GET/CUR STARS/DESCR`. The cursor appears on the screen.
- Point the cursor to the selected star and press `Enter`.
- After recording all standard stars, set cursors off and press button `Enter` once more. Now the frame is ready for the `ANALYSE/INV` command with keyword PRFLCTRL set to a negative value.

When selecting stars with the cursor, you should also use very bright saturated stars. The program can handle them properly, and they are very useful in extending the range of the Point Spread Function determination. However, in the case of CCD–frames the program does not yet know how to deal with saturated vertical columns that spread out from the central regions of bright stars. Therefore, in the case of CCD–frames, only stars without saturated vertical columns should be used.

Note

No method of determining the Point Spread Function can be successful when the data has not been properly transformed into intensity units.

The used point spread function is written as a descriptor DPROFILE into the input image frame. This descriptor is modified whenever `ANALYSE/INV` is run with keyword PRFLCTRL not equal to zero. The program looks for an initial point spread function into that descriptor at first and it uses data from the keywords SPROFILx when the descriptor DPROFILE is absent.

4.3.7 In Case of Trouble

It may happen that program execution is aborted or that the output is clearly wrong. In most cases it is caused by setting wrong values for keywords. Check carefully whether keyword values look reasonable. If there is no obvious mistake made in setting the keywords, run the ANALYSE/INV program again using the DEBUG option. This can be done by using DEBUG as one of the parameters of the ANALYSE/INV command line. A lot of information will be displayed for each object. It is possible to follow the program flow and to see how the cleaning of blended images is done. Because of the large amount of information displayed, using the DEBUG option is feasible only for very small frames. Identify on the image display the region which gives most of the trouble such as not detecting obvious objects, failure to resolve pairs, or multiple detections of single objects, and use the EXTRACT command to create a subframe of dimensions between 50×50 and 150×150 pixels with up to $10 \ldots 20$ objects. Then use the command SEARCH/INV and finally ANALYSE/INV with the DEBUG option and try to find out, based on the displayed data, what the source of the trouble is.

Displaying data concerning a particular object starts with its identification number, and its pixel coordinates. Next comes three kinds of information, each consisting of 80 values. They are arranged in eight columns. The first presents a profile in eight octants. Each column corresponds to a particular octant, starting with an octant pointing to the right and going counter clockwise. The first row gives the value of the central point. The second kind of 80–values array uses only four first columns. The first column gives average profile, the next three give the first three amplitudes of Fourier expansion of the objects profile over the octants. In both these cases data has been divided by background and multiplied by 1000 for easier management of displays. The third kind of an array contains only 0 and 1. This array indicates how the object was cleaned. The value of 0 indicates that the actual data is used. The value of 1 means that the data is interpolated. The arrangement of columns and rows is the same as in the case of displaying the profile. Only the row corresponding to the central point is missing.

After intermediate data is displayed for all objects, the final results are shown. They are presented in three separate chunks of data for each object. The first gives *ID, x–coordinate, y–coordinate, distance to nearest other object*, *extent of unblended central part*, *approximate radius of an object*, and *radius of saturated part* of an image, all except ID expressed in pixels.

The second gives data that is stored in an output table. A list of table column numbers and labels can be used for identifying displayed quantities. Data from columns :IDENT, :NR_OTH_OBJ, and :AR is not presented here, as it has been previously displayed.

The third gives first 21 points of *one–dimensional profile.*

4.3.8 The Classification

The command CLASSIFY/INV performs a classification of listed objects on *stars*, *galaxies*, members of a broad class of *defects*, and *unclassified*—mostly very faint—objects. Only the table produced by the ANALYSE/INV command can be used as input for the CLASSIFY/INV command. An additional column containing the result of the classification will be appended to this table. The user can have some influence on the resulting classification by setting few keywords. In some cases it is known beforehand what kind of objects predominate at the faintest magnitudes where accurate classification is impossible. Proper setting of keywords CLASSPAR and DISTANCE can accomplish subjective segregation of the faintest objects into stars and galaxies. The results of classification are reliable only for objects that are considerably brighter than the limiting magnitude.

The following additional keywords can be set with the help of the `SET/INV C` command:

BRGHTLMT — This keyword controls the division of objects into "bright" and "faint". Different algorithms are used for classifying these two groups. The value of this keyword is expressed in magnitudes and *should be fainter than the saturation limit.*

DISTANCE — This keyword controls convergence of an iterative classification process. *Usually the value of 30.0 is sufficiently good.* It may be changed when the iterations fail to converge.

ITERATE — Sets conditions for terminating iterations. Each iteration usually changes the classification of some objects. The number of such changes is evaluated and compared with the first value of this keyword, and iterations are terminated when the number of changes is not larger than the value of the keyword.

The second value of this keyword gives the allowed number of iterations. *Recommended values are 0,20.* It happens sometimes that iterations converge initially into a proper solution, but fail to stop and begin to diverge at an accelerating pace. It is good in such cases to increase the first component and to decrease the second value of the keyword ITERATE.

CLASSPAR — This keyword controls the selection of "seed objects", which teach the program how typical stars and galaxies look. This selection is done on the isophotal magnitude versus the relative gradient diagram. There are three parameters. The first sets a half–width of a band centered on relative gradient = 0.0, and extends from bright objects down to a limit set by the third parameter, which gives the difference of magnitudes between this limit and the detection limit. The second parameter sets an upper limit of the relative gradient for selecting a seed galaxy. *Recommended values are 0.05, -0.15, 1.0.*

4.4 Description of INVENTORY Keywords

There are several keywords in the context keyword area which hold the values of parameters which control the performance of `INVENTORY`. They are listed here. Additional information is also supplied (separated by slashes): Keyword type — Integer (I) or Real (R) — and the maximum number of values stored by a keyword. A description is added following each keyword.

ANALITER — Default value of this keyword is equal to 0 and corresponds to the case that only the base version of the command `ANALYSE/INV` is used. Any positive value sets the number of additional iterations. In each of these iterations the contributions from stellar neighbours are subtracted using the empirical one-dimensional point spread function.

BRGHTLMT/R/1 — Divides objects into bright and faint for classification purposes. Should be close to the magnitude corresponding to the beginning of saturation effects.

BRIGHTOB/R/1 — Parameter, which helps to remove spurious faint components of large bright objects. May modify central parts of real objects. Should be disabled by setting to zero when photometric accuracy is important.

BRGTCTRL/I/1 — Many spurious faint objects are detected around very bright ones when this keyword is set to 0. Increasing it helps to eliminate spurious objects. Too large a value may result in disapearance of some real objects.

CLASSPAR/R/3 — Controls the selection of seed objects for classification. The first component defines half–width of stellar objects band in a magnitude versus relative gradient plot. The second value sets an upper limit for relative gradient for seed galaxies. The third component tells how much brighter in magnitudes the seed objects should be in relation to the limiting magnitude. All three components are used in `CLASSIFY/INV`. The first component is also used in `ANALYSE/INV` for picking out objects similar to stars.

CTRLMODE/R/1 — Controls the calculation of sky background with the help of the subroutine MODE. This subroutine returns the modal value of the pixel distribution when CTRLMODE is set to 3.0. Repetitively clipped means it is returned when CTRLMODE is set to 0.0, with some saving of the execution time.

CVFACTOR/R/1 — Stops iterations in subroutine MODE, when the difference between the last two iterations for mode is smaller than the last σ multiplied by CFCT. *Recommended value is 5.0.* Smaller values increase execution time with only little improvement of accuracy. Larger values may harm the accuracy.

DEBUG/I/1 — us used ti invoke debuging outputs.

DFVALLEY/R/1 — Required depth of valley between two components that is necessary for accepting them as two separate objects. *Recommended value is 0.02.*

DISTANCE/R/1 — Limiting distance in a parameters space for including object in a particular class of objects. Used in `CLASSIFY/INV`

ELONGLIM/R/1 — Image elongation above which an object is checked for duplicity. *Recommended value is 0.25.*

ETAFUNCT/R/1 — Petrosian η–function used in the determination of the Petrosian radius and Petrosian magnitude. Berkeley astronomers are using a value of 2.0. When working with many automatically detected faint objects this large value often fails to give consistent results for all objects. Therefore a smaller value, such as *1.5, is recommended.* The value 1.39 corresponds roughly to de Vaucouleur's definition of "half luminosity radius".

EXPRTIME/R/1 — Time of exposure expressed in units referred to in keyword ZERO-MAGN.

FIELDVAR/I/1 — Concerns the character of variations of sky background. It is set to 0 when these variations are additive. It is 0 when they are due to sensitivity changes.

FILTER/R/1 — Sets slevel of bright pixel filtering.

FULLPSF/I/1 — Extend in pixels of 2-dimensional p.s.f.

HALFEDGE/I/1 — Defines size of a subarray that is used for calculating local sky background, and for profile analysis. It is equal to a distance, measured in pixels, between the central pixel and an edge of a subarray. The size of a subarray is equal to $(1 + 2 \times \mathrm{IHED})^2$. IHED cannot be larger than 50, so that the maximum size of a subarray is 101×101 pixels. Objects of interest should be smaller than the subarray. If the value 50 is too small, then the image should be squeezed. Minimal value is 4. *Recommended value for most of applications is between 8 and 12.*

IJBORDER/I/4 — Sets limits of the investigated area.

ISOPHOTR/I/1 — Controls use of outside values of isophotal radia. When set to 1, then isophotal radia are read from column RAD_ISO into an input table. It is useful when measuring colours of galaxies.

ITERATE/I/2 — Condition for terminating iterations during classification and allowable number of iterations.

LHCUTS/R/2 — Low and high cuts applied for more correct treatment of bad pixels and saturated regions. The *low cut* should be *smaller than any valid data.* The *high cut* should be *little less than the image saturation level.* It is advisable to look carefully at images of bright stars with the help of the image display in order to set a high cut.

MARGINS/I/1 — Width in pixels of frame margins where objects are not detected.

MULTDTCT/R/1 — Allowed separation of pixels at multiple detections. Individual detections separated not more than by MULTDTCT pixels are treated as the same detection.

NEBLIMIT/R/1 — Used in cases when faint members of clusters of galaxies pretend to be very large. In all other cases should be disabled by setting to zero.

NETHEDGE/I/1 — Half edge of the regions used to determine preliminary sky background in a net of regions.

OUPROFIL/I/1 — Sets number of pixels spaced points of objects' profiles recorded in output table.

PAIRSPRT/R/1 — Allowed minimal separation of double objects expressed in pixels. Should be *little larger than* the size of *the seeing disk.*

PRFLCLNG/R/3 — Parameters used for profile cleaning.

PHYSICAL/I/1 — Tells if searched area is defined by physical coordinates.

PRFLCTRL/I/1 — Gives the number of central points of the standard Point Spread Function to be updated by program. Input profile as given by keywords SPROFILx and is used when PRFLCRTL is 0. N central points are updated with the help of stars selected by program when PRFLCRTL is $+n$, and the program will ask for selecting stars with the help of the cursor when PRFLCRTL is $-n$. *Usually 5 or 6 is sufficient.* May depend on frame origin, seeing and pixel size.

SGFACTOR/R/1 — Factor used to set clipping limits in subroutine MODE. *Recommended value is 2.0.*

SKYDETER/I/1 — Sets accuracy of the sky background determination.

STMETRIC/R/2 — Radii of two circular apertures used to determine aperture magnitudes. They are expressed in pixels.

SPROFIL1/R/5 – SPROFIL5/R/5 — Innermost 25 points of one dimensional logarithmic differential Point Spread Function. The spacing of entries is by one pixel. Each k–th value is equal to decimal logarithm of k-1 value of the stellar Point Spread Function divided by its k–th value.

TRESHOLD/R/1 — Limiting threshold above sky background level used at object detection and at calculating isophotal magnitudes and sizes. It is expressed in units of sky background. Too low a value results in many spurious detections and corresponding long execution time. It is advisable to start with a relatively *large* value of TRESHOLD and check it on a small part of a frame defined with the help of the EXTRACT command.

UNDRSMPL/I/1 — Undersampling factor.

UNITS/I/1 — Switched between background units (0) and instrumental units (1)

XYBORDER/R/2 — Limits of investigated area in physical units.

YFACTOR/R/1 — The ratio of "Y" to "X" dimensions of a box used at duplicity check. *Recommended value is 0.6.*

ZEROMAGN/R/1 — Sets a zero point of magnitudes. Its use is controlled by keyword MGNTCTRL.

4.5 Formats of Tables

Intermediate inputs for and outputs from the `INVENTORY` package are stored as MIDAS tables. These table formats are described here.

4.5.1 Input Table

In some applications there is no need to run the SEARCH program because a list of objects with accurate positions is already available. In this case, the ANALYSE command should be used with the NOVERIFY option. The ANALYSE command with the NOVERIFY option expects the values of the X and Y coordinates (in physical units) to be stored as columns in an input table. There are no limitations to the positions of these columns in that table. The columns are identified by the column labels :X_COORD and :Y_COORD. It is also possible to enter an isophotal radius for each object as input value. This mode is activated by setting the parameter ISOPHOTR to 1. The input table should have a column labeled with :RAD_ISO, which holds the isophotal radii.

4.5.2 Intermediate Table

The structure of an output table from the SEARCH/INV, which serves as an input table for the ANALYSE/INV command is transparent for the user; there is no need to know its format.

4.5.3 Output Table

The output table contains the results and therefore it is necessary to know in what columns different quantities are stored. Table 4.1 gives a list of columns together with their numbers and labels. Column :CLASS is filled with data by the CLASSIFY command.

4.6 Inventory Commands Summary

Table 4.2 lists all commands discussed in this chapter.

Column	Name	Description
#1	:IDENT	Identification.
#2	:X	Physical X–coordinate.
#3	:Y	Physical Y–coordinate.
#4	:MAG_CNV	Convolved magnitude. Applicable to stars.
#5	:MAG_AP_1	Aperture magnitude no.1.
#6	:MAG_AP_2	Aperture magnitude no.2.
#7	:REL_GRAD	Relative gradient. Should be 0.0 for stars.
#8	:SIGMA_GR	Sigma of profile deviations from P.S.F.
#9	:BG	Local sky background.
#10	:INT	Average intensity of 9 central pixels.
#11	:RADIUS_1	Intensity weighted average radius in pixels.
#12	:RADIUS_2	Intensity weighted average root square radius in pixels.
#13	:ELONG	Image elongation.
#14	:POS_ANG	Position angle of an image major axis.
#15	:RAD_ISO	Isophotal radius in pixels.
#16	:MAG_PET	Petrosian magnitude.
#17	:RAD_PET	Petrosian radius.
#18	:RAD_KRON	Intensity weighted average inverse root square radius.
#19	:ALPHA	Alpha gradient at the edge of aperture magnitude no.1.
#20	:MAG_ISO	Isophotal Magnitude.
#21	:CLASS	Object's classification: 2=Galaxies, 1=Stars, 0=Defects.
#22	:NR_OTH_OBJ	Distance to nearest other object.
#23	:AR	Active radius.
#24	:SIGMA_X	r.m.s. in computed x coordinate.
#25	:SIGMA_Y	r.m.s. in computed y coordinate.
#26	:SIGMA_MAG	r.m.s. in computed magnitude

Table 4.1: Inventory Output Table

```
ANALYSE/INV frame table1 [table2] [ver_opt] [debug_opt]
CLASSIFY/INV table
SEARCH/INV frame table
SET/INV command [ALL]
SHOW/INV command [ALL]
```

Table 4.2: Inventory Commands

Chapter 5

Crowded Field Photometry

5.1 Introduction

This chapter describes the set of commands implemented in MIDAS to do crowded field photometry on digital astronomical images. The package has been developed by R. Buonanno, G. Buscema, C. Corsi, I. Ferraro, and G. Iannicola, all at the Osservatorio Astronomico di Roma. The package runs as a part of the data reduction system at Rome Observatory and is well known from its name: ROMAFOT. Since in Italian the suffix "fot" refers also to the word "photometry", the name ROMAFOT was chosen as an easy homophony with DAOPHOT.

The idea of ROMAFOT was born (and realised) in the second half of the seventies. The need for such a package has arisen from the technical evolution (for instance the arrival of the PDS etc.) and from the huge amounts of data becoming manageable. A rich bibliography of authors who worked on the same objective (e.g. Newell and O'Neil, 1974 [1], Van Altena and Auer, 1975 [2], Butcher, 1977 [3], Herzog and Illingworth, 1977 [4], Chiu, 1977 [5], Auer and Van Altena, 1978 [6] Buonanno et al., 1979 [7], Buonanno et al., 1983 [8] Stetson, 1979 [9], Stetson, 1979 [10]) shows that efficient handling of digital astronomical images was widely felt for the astronomical community. In addition, since the numerical problem requires very classic solutions, neither the *idea* nor the *solution* asks for particular emphasis.
Where ROMAFOT has perhaps merit is in having taken dozens of decisions such as:

- should the sky background be calculated before or together with the star?
- which size of the subwindow optimises the ratio photometric quality/computing time?
- when should two objects be considered blended and be fitted together?
- which is the optimum degree of interaction?

All these choices require experiments, time, and naturally effort.

In this description Section 5.2 gives some theoretical background about how ROMAFOT operates. Section 5.3 presents an overview of the commands available in the ROMAFOT context. Section 5.4 describes the commands in detail. The section is split into a part for the automatic reduction and a part for the more interactive reduction. Finally, Section 5.5 gives a summary of all ROMAFOT commands.

5.2 Theory

ROMAFOT uses a linearised least squares analytical reconstruction method on two–dimensional data frames. Introductions of modern tests of significance (e.g. χ^2) to measure the discrepancy between observations and hypothesis can be found in many excellent text books (see e. g. Bevington, 1969). Hence, it is not really useful to give any theoretical summary here.

Although minimisation of the absolute deviations or, more precisely, the square of deviations between data and a parametric model is not the only technique to derive optimum values for parameters, it is generally used because it is straightforward and theoretically justified. There are no basic difficulties in extrapolating this method for the case of a model function with non–linear parameters if χ^2 is continuous in parameter space. Problems may arise if local minima occur for "reasonable" values of parameters and methods need to be employed to search for absolute minima.

Nevertheless, if the PSF parameters are reasonably estimated one can assume that the localisation of the region of absolute minimum is generally well determined. This is an important condition since the method ROMAFOT uses requires that high order terms in the expansion of the fitting function are negligible, a condition satisfied if the starting point is close to the minimum.

This method is to expand the fitting function to first order in a Taylor expansion, where the derivatives are taken with respect to the parameters and evaluated for the initial guess in this space. The calculation of the model parameters to fit the data has led to the solution of the usual system:

$$\frac{\delta\,\Sigma}{\delta\,P_j} = 0,$$

where Σ is the summation over all the n points of the square of the difference between the model function and the data, and P_j is the current parameter ($j = 1, ..., m$). As the method requires an approximation of the function instead of the function itself, the parameter adjustments the system reinstitutes are not exactly those which correct the trial values, and the operations must be iterated.

Using the technique just outlined, ROMAFOT carries out stellar photometry by fitting a sum of elementary Point Spread Functions, analytically determined, to a two dimensional array. The sky contribution is taken into account with an analytical plane, possibly tilted. This plane is fitted (simultaneously) with the stars.

The analytical formulation of the PSF has obvious advantages in case of undersampled images allowing the comparison of the intensity of each pixel to the *integral* of the function over the pixel area.

ROMAFOT does not attempt to discriminate between stars and galaxies in the phase of searching the objects. It prefers to take into account their photometric contribution and delete them from the final catalogue by checking the fit parameters with a suitable program. This program also eliminates cosmic rays and defects, if present.

In several parts in this documentation reference is made to the sigma (σ) of either the Gaussian or the Moffat fitting function. In both cases this sigma is **NOT** the sigma in the statistical sense (i.e. the standard deviation) but the Full Width Half Maximum (FWHM) of the distribution. In case of the Moffet distribution, sigma is a function of the parameter β, i.e. not equal to the FWHM except for $\beta = 3.1$. This definition of sigma will be used throughout this chapter.

5.3 Overview of ROMAFOT

The current MIDAS implementation of ROMAFOT consists of 17 commands which are listed in Table 5.3. In order to be able to execute these commands the context ROMAFOT should be enabled first. This can be done using the command `SET/CONTEXT ROMAFOT`.

Command	Description
ADAPT/ROMAFOT	Adapt trial values to a new image frame
ADDSTAR/ROMAFOT	Add pre-selected subarrays to the original image frame
ANALYSE/ROMAFOT	Select objects or analyses the results of the fit procedure
CHECK/ROMAFOT	Estimate number and accuracy of artificial objects recovered
CBASE/ROMAFOT	Create the base-line for transformation of frame coordinates
CTRANS/ROMAFOT	Execute transformation of coordinates on intermediate table
DIAPHRAGM/ROMAFOT	Perform aperture photometry
EXAMINE/ROMAFOT	Examine quality of the fitted objects; flags them if needed
FCLEAN/ROMAFOT	Select subarrays containing artificial objects
FIND/ROMAFOT	Select objects from ROMAFOT frame using the image display
FIT/ROMAFOT	Determine characteristics of objects by non-linear fitting
GROUP/ROMAFOT	Perform an automatic grouping of objects
MFIT/ROMAFOT	Fit the PSF using the integral of the PSF (for undersampled data)
MODEL/ROMAFOT	Determine subpixel values to be used for the integral of the PSF
REGISTER/ROMAFOT	Compute absolute parameters and stores results in final table
RESIDUAL/ROMAFOT	Compute difference between original and reconstructed image
SEARCH/ROMAFOT	Perform actual search of object above certain threshold
SELECT/ROMAFOT	Select objects or stores parameters in intermediate table
SKY/ROMAFOT	Determine intensity histogram and background in selected areas

Table 5.1: ROMAFOT commands

ROMAFOT is as interactive as the user wishes. In fact, during a reduction session the user is often asked to choose between an interactive or an automatic procedure, including "greytone" extremes. Hence, the package leaves it up to the user:

- to select program stars,
- to model the subframes to fit,
- to choose the number of components in each subframe,
- to check the data reconstruction.

For the user one cannot define a fixed set of guidelines for these decisions, apart from the two obvious considerations that no software can take the place of the human brain and that the benefit of photometric precision should be proportional to the cost in terms of human effort.

As stated above, apart from a part which has to be interactive, the user can choose between running ROMAFOT in automatic or in interactive mode. Hence, a number of interactive ROMAFOT commands have non–interactive "sister" command(s) with the same functionality. Figure 5.1 shows the flow diagram of the ROMAFOT package. From this diagram it is clear that after determining the PSF one can continue along two paths. Figure 5.2 visualises the procedure to insert artificial stars in the original frame in order to estimate the ability of the program to recover an object and the general reproducibility of the photometry. Finally, Figure 5.3 shows the procedure to use the photometric results (positions and intensities) from the frame with the best resolution as input for all the other program frames.

		SET/CONTEXT ROMAFOT			
		SELECT/ROMAFOT			study PSF
		$\Downarrow$			
		FIT/ROMAFOT			
		$\Downarrow$			
		ANALYSE/ROMAFOT			
Interactive		$\Downarrow$		**Automatic**	
	$\Longleftarrow$		$\Longrightarrow$		
FIND/ROMAFOT				SKY/ROMAFOT	search
				SEARCH/ROMAFOT	
$\Downarrow$		$\Longleftarrow$		$\Downarrow$	
ANALYSE/ROMAFOT				GROUP/ROMAFOT	input
	$\Longrightarrow$		$\Longleftarrow$		
		$\Downarrow$			
		FIT/ROMAFOT			fit
	$\Longleftarrow$		$\Longrightarrow$		
ANALYSE/ROMAFOT				EXAMINE/ROMAFOT	check
	$\Longrightarrow$		$\Longleftarrow$		
		$\Downarrow$			
		REGISTER/ROMAFOT			store
		$\Downarrow$			
		RESIDUAL/ROMAFOT			residuals

Figure 5.1: Romafot reduction scheme

		ADDSTAR/ROMAFOT			artificial
Interactive		$\Downarrow$		**Automatic**	
	$\Longleftarrow$		$\Longrightarrow$		
				SKY/ROMAFOT	search
				SEARCH/ROMAFOT	
$\Downarrow$		$\Longleftarrow$		$\Downarrow$	
ANALYSE/ROMAFOT				GROUP/ROMAFOT	input
				FCLEAN/ROMAFOT	
	$\Longrightarrow$		$\Longleftarrow$		
		$\Downarrow$			
		FIT/ROMAFOT			fit
	$\Longleftarrow$		$\Longrightarrow$		
ANALYSE/ROMAFOT				EXAMINE/ROMAFOT	check
	$\Longrightarrow$		$\Longleftarrow$		
		$\Downarrow$			
		REGISTER/ROMAFOT			store
		$\Downarrow$			
		CHECK/ROMAFOT			

Figure 5.2: Romafot procedure to determine accuracy and degree of completeness

	CBASE/ROMAFOT	base
	$\Downarrow$	
	CTRANS/ROMAFOT	transform
	$\Downarrow$	
$\Longrightarrow$	ADAPT/ROMAFOT	
$\Uparrow$	$\Downarrow$	
$\Longleftarrow$	ANALYSE/ROMAFOT	check

Figure 5.3: Romafot procedure to transfer inputs to other program frames

ROMAFOT is fully integrated into the MIDAS environment, which means that the image display part runs on all display system supported by MIDAS and that all intermediate and the final results are stores in MIDAS tables. Obviously, the use of the MIDAS table file system adds a great deal of flexibility to ROMAFOT, since a large number of MIDAS commands can be used (e.g. for table emanupulation, displaying, selection, etc.).

5.4 How to use ROMAFOT

This section contains detailed information on what a typical reduction session with the ROMAFOT package can look like. It describes, in much more detail than in the on–

line help files, the functionality of the ROMAFOT commands, in particular how several algorithms work (e.g. the search and fitting algorithms). Also, it shows how the user can shift from the automatic to the interactive mode, and gives suggestions as to what action to take in particular circumstances. The first–time–user is advised to read this documentation before starting; for the more experienced user this section may serve as a trouble shooting section and as a reference.

In the description of the ROMAFOT package below, the schemes in Figure 5.1, Figure 5.2 and Figure 5.3 will be followed. First, the commands needed to determine the Point Spread Function (PSF) will be described. Thereafter, the interactive and the automatic paths will be explained. Then, the procedure to determine the photometric accuracy and the degree of completeness will be illustrated. Finally, an efficient path to measure several frames of the same field is presented. For each of the steps described in the documentation the command and the command syntax are given.

5.4.1 Study of the Point Spread Function

SELECT/ROMAFOT

`SELECT/ROMAFOT frame [int_tab] [wnd_size]`

With this command the user can select a number of well behaving stellar images to determine the Point Spread Function on the frame. These well behaving stellar objects (of the order of 10 to 15) should be chosen over the whole frame.
`SELECT/ROMAFOT` displays the frame (or part of it) on the image display and allows the user to select the objects. The image display cursor and its control box is used to select the objects .

Since this command is also used to select special classes of objects, (e.g. photometric standards) it asks for magnitudes and names of the objects selected. If these are unknown the default can be used.

After execution a set of subframes with given dimensions (typically 21 by 21 pixels) centered on the input cursor position and each containing one star is stored in the intermediate table. This table can be feeded to the the command `FIT/ROMAFOT` which actually performs the fit to the data. The table also contains trial values both for the sky background and the central intensity of the stars.

FIT/ROMAFOT

`FIT/ROMAFOT frame [int_tab] [thres,sky] [sig,sat,tol,iter] [meth[,beta]] [fit_opt] [mean_opt]`

This command determines the characteristics (position, width, height) of each selected stellar object through a non–linear best fit to the data. It assumes that a Gaussian or a Moffat function is adequate to describe the PSF and that a (possibly tilted) plane is a good approximation of the sky background.

This command can be used for many purposes. For instance, the shape of the object can be determined by performing a best fit with all parameters allowed to vary; alternatively, a complex object (e.g. a blend of ten or more objects) can be reconstructed using some a priori knowledge, such as the width of the PSF or the positions. In the first case, an object with an informative content which is as high as possible is necessary to settle the parameters involved; in the latter case this information is added to the data having a low degree of information.

Experience has shown that a Moffat function with appropriate parameters is always able to follow the actual profile of the data satisfactorily; a Gaussian is adequate in case of poor seeing. In general the fitting function can be described by the expression:

$$F(a_i, p_i) \;=\; a_1 x + a_2 y + a_3 + \sum_{k=1}^{36} f_k(x, y, p_{i,k}),$$

where the a_i's are the sky background coefficients, the p_i's the parameters of the k elementary components and the f_k given by:

$$f_k \Longrightarrow \begin{cases} I = p_1 \exp -4\ln 2\{\frac{(x-p_2)^2+(y-p_3)^2}{p_4^2}\}, & \text{Gaussian} \\ I = p_1\{1 + \frac{(x-p_2)^2+(y-p_3)^2}{p_4^2}\}^{-\beta}, & \text{Moffat} \end{cases}$$

As has be mentioned in the Introduction, in the both expressions above, the σ is **NOT** the sigma in the statistical sense (the standard deviation). For the Gaussian function σ refers to the Full Width Half Maximum (FWHM) of the distribution; in case of the Moffet distribution, σ is a function of the parameter β.

Suppose at the beginning of the session some isolated objects have been selected in order to derive the PSF. The number of components per window, k, is set to 1 and the command runs with p_1, p_2, p_3, p_4 and β all allowed to vary. However, experience shows that p_4 (hereafter often referred to as σ) and β are not totally independent. Therefore, it is preferable to fix β at the typical value of $\beta = 4$, to derive the corresponding σ and to check the quality of the fit interactively. If the fit is unsatisfactory, change β and derive the new σ. Since profiles are not a strong function of β the parameter can be changed by a couple of units. Remember that if the fitted profile is wider than the object, β should increase and vice versa. Typically, β must be kept greater than 1 and it should not exceed 10. Naturally, these considerations do not apply if the Gaussian function is used. However, in case of stellar photometry the use of the Moffat function is strongly recommended.

Besides the best fit, `FIT/ROMAFOT` also computes the quality of the fit by the χ^2 test and the *semi–interquartile interval* for each individual object. These data are stored together with the fit parameters and will be used by other commands (see `EXAMINE/ROMAFOT`).

During the execution the user will realise that the command occasionally makes several trials on the same window. This happens when the command is requested to fit a window with several objects and when one or more of these falls into the category "NO CONVERGENCY". In this case the command continues by ignoring such objects. When finally the convergency is found, the objects so far ignored are added and a new trial will start. The program will never delete objects on its own. The only exception is if an object falls under the threshold selected by the user, and after the background has been properly calculated considering, for instance, "tails" of nearby stars.

It should be emphasised that, even if the window is marked "NO CONVERGENCY", some objects in that window (in general the most luminous ones) have been found with adequate convergency. These objects will be flagged "1", while the objects flagged "3", "4" or "5" will be those responsible for "NO CONVERGENCY". Finally, objects flagged "0" are those under the photometry threshold. These flags should not worry the user; they will be used by subsequent commands.

Now, the trial values contained in the intermediate table have been substituted by the result of fit in each record. To check their quality interactively in order to define the PSF, the user should execute the next command.

ANALYSE/ROMAFOT

`ANALYSE/ROMAFOT frame [cat_tab] [int_tab] [sigma,sat]`

This command is the most interactive part in the reduction procedure. Because of its flexibility, its use may not seem straightforward. Therefore, after starting up the command, the user is advised to press the key "H": it will display the help information the screen. For the sake of clarity only those commands which are useful at this stage will be discussed below; other commands of the `ANALYSE/ROMAFOT` will be discussed later.

After having received the appropriate input from the MIDAS command line, the command waits for a command input. Following the present example, the user should type "D" and a number. The command then asks for a number, say n, and will read the data for the n^{th} group of components stored in the intermediate table. `ANALYSE/ROMAFOT` will display the data profiles and the fit superimposed on the upper part of the image display. On the lower part of the screen, three images are shown: the subframe containing the real data, the subframe containing the fit and the difference of the two, respectively.

The user can squeeze the interval over which the 256 colours are distributed through the command "W" or vary the minimum of such an interval with the command "U". In order to have a better check of the correspondence between the data and the fit, the command "X" displays isophotes at three different levels. The key "Y" has the same function but starts at a level given by the user (typical numbers are 0.01 to 0.001). Another command useful for display is "6" which allows the smoothing of the subframe containing the real data. This smoothing is performed via a 3×3 gaussian convolution mask on the array stored in memory and does not affect the original array.

By means of the displayed information one can decide if the fit is adequate or for example systematically wrong (for instance, all the images have overestimated σ's or heights). In the latter case `FIT/ROMAFOT` can be run with different parameters. The procedure continues with "D", $n+1$, $n+2$, ... etc.

A mean of all fits obtained will finally give the σ of the PSF. Since β had been already fixed, the PSF is now determined.

5.4.2 The Interactive Path

As mentioned above, ROMAFOT is interactive or automatic to the extent that the user chooses. It should be clear that the user is **not** obliged to choose between the automatic **or** the interactive procedure. In other words the user can shift in the next steps of the reduction sequence from the automatic to the interactive mode or vice versa.

Below, the interactive procedure will be described first, simply because it is easier. Definitely, it is not recommended to follow this procedure for photometry on more than 100 stars. However, a certain degree of interactivity has the advantage that it makes the user aware of what he/she asks the computer to do and he/she shares with it responsibility for the results.

Of course, to start interactive photometry of program stars one should select the stars. This can be done with the command:

FIND/ROMAFOT

`FIND/ROMAFOT frame [cat_tab]`

This command works exactly like `SELECT/ROMAFOT` and therefore needs no detailed explanation. The only difference is that `FIND/ROMAFOT`, designed to select many stars, stores the positions of the objects in a catalogue table which has a different structure than then the intermediate table created by `SELECT/ROMAFOT`. This table serves as as input for the command `ANALYSE/ROMAFOT`. The reason for this different table structure will become clear later.

Having selected the program stars, one has now to settle the shape and dimensions of the windows in which to perform the fit. In addition, the number of elementary components that should be included in the fit has to be indicated. This operation can be done by running the command `ANALYSE/ROMAFOT` once again.

ANALYSE/ROMAFOT

`ANALYSE/ROMAFOT frame [cat_tab] [int_tab] [sigma,sat]`

To examine the catalogue table created by the command `FIND/ROMAFOT` one should use the command "M" followed by the sequential position of the object. The commands "W", "U", "6", "X" and "Y" will set the display.

A subframe with the selected star at the center is presented to the user. This subframe can be reduced (or enlarged) by the command "R" followed by the decrement (or increment) in x and y: delta_x, delta_y, respectively.

The command "S", followed by the shift in x and y coordinate provokes the displacement of the window. "R" and "S" are useful to include or exclude stars in the subframe.

A "bar" allows the display of the current status of the window.

The intensity profiles are displayed with a scale which saturates at an intensity of $I \simeq 950$. This can be changed with the command "B" followed by a factor for the intensities, typically 0.1 or smaller.

In crowded fields the maps, isophotal contours and intensity profiles are not enough to visualize the situation. In this case the commands "K" and "L" may help. With "K" one gets the pixel coordinates (both absolute in the frame and relative in the window) and the pixel intensity at the cursor position. The command "L" gives the height (above the background) at the vertical cursor position.

If the window has been properly set, shaped and understood, the user is expected to position the cursor where he thinks a star exists and give the command "I". This command passes the trial position of the star and its height and will be repeated for all the stars in the window. The maximum number of objects allowed is 36, but the user is discouraged from approaching this limit for many reasons. The most simple one is that 36 stars (i.e. 36×3 parameters) which are fitted separately in 36 subframes with N pixels each require $36 \times 3 \times N = 108N$ calculations at each iteration: the same fit all together requires $36 \times 3 \times 36 \times N = 3888N$ calculations.

The command "J" works like "I" with the difference that the user is expected to provide the trial value for the height: this can be useful in case of difficult convergence when e.g. several local minima exist in the parameter space.

If some unwanted feature is present in the window (cosmic rays, scratches, tails of other stars and so on) "E" is most useful. This command provokes a hole in the window whose radius will be given by the operator. The hole is centered at the cursor position and the pixels included within its area will be ignored in the subsequent calculations as far as this window is concerned. There is no effective limit to the number of holes (≤ 50). For this reason, it is not practical to have hole positions reported on the screen (in fact, they could fill the whole screen). Therefore by default the holes will not be listed. To change that use the command "-" which enables the report hole positions and sizes. This command executes the inverse operation, as well.

During the examination of the objects passed by `FIND/ROMAFOT` it often happens that several stars are asked to fit together in the same window and some of these are in the catalogue table. In order not to repeat photometry of the same objects, `ANALYSE/ROMAFOT` flags in the catalogue table all the objects already considered and the user is supplied with the message "object already considered". It is possible to disable this feature with the key "Z".

The user may realize that the windows presented are too wide (or too narrow) depending on the telescope scale, seeing, crowding and so forth. In this case the command "/" can be used to change the default window size.

A minor feature is the key "A". This allows the default display of level contours (instead of colour maps) without using command "X" or "Y".

Finally, the command "5" selects the mode to examine the catalogue table, either to give the record (manual mode$\rightarrow$ "M") or running through the table (automatic mode $\rightarrow$ "A"). It is possible also to examine the list selectively ($\rightarrow$ "S"). For instance all the objects above a given height threshold are disregarded by the command `GROUP/ROMAFOT`.

After execution of `ANALYSE/ROMAFOT` all the program stars are arranged with their trial values in (normally multiple) windows with selected dimensions and shapes: these data are stored in the intermediate table. To continue the user should now run the command `FIT/ROMAFOT`.

FIT/ROMAFOT

`FIT/ROMAFOT frame [int_tab] [thres,sky] [sig,sat,tol,iter] [meth[,beta]] [fit_opt] [mean_opt]`

Obviously the previously determined PSF should now be used. Therefore, in the interactive enquire the user should set the logical character for fixing σ to "Y".

FIT/ROMAFOT will fit the data following the trial values contained in the intermediate table. In order to check the results the user can now run the ANALYSE/ROMAFOT command once again.

ANALYSE/ROMAFOT

ANALYSE/ROMAFOT frame [cat_tab] [int_tab] [sigma,sat]

In case the user wants to examine all windows, the command "4" should be used with the option for the automatic analysis of the intermediate table. Inspection of selected windows (no convergence, more than N iterations and so on) is also possible.

Looking at the display, one can see the reasons why the fit has some problems and can take action. With the command "C" one can delete one component, with "I" (or "J") one can add one or more components. With the "E" command one can add a hole and with "G" one can delete the hole and restore the subframe. Both "C" and "G" ask for the component (or hole) to delete.

Selecting the contribution of one individual component is not an easy task in crowded windows. The problem can be overcome with the commands "P" and "T". After "P" one must give the sequential position of the star to examine in the window: this flags with "0" all the other components disabling their display. The user is therefore left with only one component on the screen.

It must be noted that after the command "P", the key "Q" always has to be used to restore the window. Otherwise, all the other stars will be maintained with the "0" flag, and consequently not considered in following operations (registration, fit and so forth).

The command "T" is complimentary to the "P" command since it adds the "0" flag only to the sequential component selected by the user. "T" is also used to restore the $n^{\rm th}$ component.

If for some reason the window is totally irrecoverable ANALYSE/ROMAFOT offers the possibility to start again with the window. One should delete all the components and holes with "C" and "G" and then call the window with "N". In this way the intermediate table is read in **input** mode allowing for commands "S" and "R", not permitted when the table is read with the "D" command.

A useful feature is the command "V" which enables new tables to be opened, for instance a new frame. This is sometimes used to look at variables on different frames or at objects with extreme colour indexes.

Finally, the command "@" allows the substitution of one component in a window. In principle, this operation could be performed with the commands "C" and "I". However, in that case the new component is appended. With "@" the component can be inserted at a given position of the list. The actual fit is completely independent of whichever of the two possibilities one chooses; however, since ROMAFOT attributes names to the stars according to their sequential position, the two operations produce different names. This could be important for subsequent procedures. The same is true if one deletes one component or flags it with the "T" command. The result for photometry is identical since the object disappears in both cases but, in case of "T", the object survives as far as the name is concerned.

After all the necessary corrections have been carried out, the user can now select the windows to be fitted by running the command `SELECT/TABLE`. The selection should be put on the windows which should be untouched by a subsequent command `FIT/ROMAFOT`. This command will then fit only the indicated windows. Of course, to find a satisfactory solution, the loop `ANALYSE/ROMAFOT` → `FIT/ROMAFOT` can be executed as many times as needed. Finally, one should run the command:

ANALYSE/ROMAFOT

`ANALYSE/ROMAFOT frame [cat_tab] [int_tab] [sigma,sat]`

Here, the "4" and "A" commands should be executed first. Then, execute the "D" to examine all the windows.

Finally, after all the windows have been examined, the data must be registered in a final MIDAS table with the command `REGISTER/ROMAFOT`. This command assigns identification number to the objects corresponding to the group identification: the first component will have identification N*100+1, the second N*100+2, etc, where N is the group identification number.

5.4.3 The Automatic Path

Above, the interactive photometry of stars in a frame has been described. Although instructive, this procedure is impractical if one deals with thousands of stars. Therefore, ROMAFOT provides automatic procedures to substitute the many interactive operations needed to obtain the same result. These automatic commands are very useful provided they are not used as black boxes. On the contrary, even following the automatic way, the user can take advantage of facilities offered by the interaction.

The first operation which can be performed automatically is the search of the objects. ROMAFOT performs the operation with two modules: the first one, `SKY/ROMAFOT`, maps the sky; the second one, `SEARCH/ROMAFOT`, detects the objects above the sky.

SKY/ROMAFOT

`SKY/ROMAFOT frame [sky_tab] [area] [nrx,nry]`

This command divides the frame into N smaller rectangular areas, with $N = nrx \times nry$. It computes the intensity histogram in each region and determines the sky values of the areas by means of the mode of the distribution. After having found the sky value of the individual regions one can do the actual search for objects above a certain threshold.

SEARCH/ROMAFOT

`SEARCH/ROMAFOT frame [sky_tab] [cat_tab] [area] [psf_par] [thresh] [height]`

This command performs the actual search of the objects. It examines the region indicated by the user and detects all the maxima above the defined threshold (or above a given relative threshold). Note that the pixel values and maxima are all relative, i.e. with respect to the sky background. The table produced by `SEARCH/ROMAFOT` is compatible with the table created by the command FIND/ROMAFOT. Therefore, after this automatic search for objects the user can pass to the interactive procedure to group the objects with the command ANALYSE/ROMAFOT. This is the first decision point of the two procedure paths.

Of course this does not mean that the user is recommended to leave the automatic route (apart from a few special cases). However, the interactivity may now be useful to look at the objects found and to decide if the threshold is appropriate or if the sampling of the background was sufficient to keep the detection threshold nearly constant all over the image.

Since it is expected that the automatic search will be generally followed by automatic grouping, SEARCH/ROMAFOT calculates an additional parameter with respect to the command FIND/ROMAFOT. With this parameter the output table of SEARCH/ROMAFOT can be used as if they were created by the command FIND/ROMAFOT. The inverse is not possible, the output table created by the command FIND/ROMAFOT cannot be used in the command GROUP/ROMAFOT. This additional parameter is the "Action Radius" (hereafter AR) defined as the distance from the center of a star at which its photometric contribution drops to a small fraction of the faintest program star. From this definition it can be inferred that the AR is a function both of the intensity of the star and of the photometric limit defined by the user. To calculate the AR the program requires the PSF parameters.

How do we define the photometric limit? Suppose one is interested in studying only stars more luminous than, say, 1000 units. Obviously, these stars are photometrically disturbed by nearby companions of 900 units. Therefore, in principle the search should be limited to the "disturbers" more than to the "targets". The grouping module will then be passed at threshold 1000 and stars in the list fainter than this limit will be considered only if they can affect photometry of nearby program stars.

At this point the function of the AR is clear: it defines an area surrounding each object within which the object has some influence in the given context. Photometry of fainter and fainter images requires larger and larger associated Action Radii, an increasing number of connections and, ultimately, more computer time.

After SEARCH/ROMAFOT one has obtained a list of candidates (often thousands). Although a quick inspection with ANALYSE/ROMAFOT is always instructive, the manual grouping of the objects is a waste of time. This can be avoided by using the command GROUP/ROMAFOT.

GROUP/ROMAFOT

`GROUP/ROMAFOT frame [area] [cat_tab] [int_tab] [thres] [wnd_max] [end_rad,sta_rad] [wnd_perc]`

This command groups the objects automatically. The catalogue table created by the command SEARCH/ROMAFOT is required as input. The command produces an intermediate table identical to the one created by ANALYSE/ROMAFOT.

The command works as follows: it starts examining the first object in the catalogue table. If the AR of this object does not intercept any other AR, the program continues to establish the window for the fit. Contrarily, if the AR does intercept the AR of another star the command examines whether a third intersection exists and so forth. When no more intersections exist, the program goes on to define the associate subframe to fit.

Going to faint photometric limits the intersections grow because of the higher number of stars and because the AR are larger. The maximum number per window technically acceptable to ROMAFOT is 36, but this figure is normally kept lower (typically lower than 20). The program, in the case of crowded fields, may refuse to group certain objects. This situation will be discussed later.

To establish the subframe to fit, a balance of opposite requirements must be achieved. For instance, since the diffraction creates images with "wide" wings one is tempted to integrate over "large" windows, causing the undesirable inclusion of many objects. On the other hand, considering that the central pixels of the image are those with higher signal to noise ratios, "small" windows could be preferable. However, this necessitates the sky background to be known "a priori", a heavy requirement indeed in case of crowded fields!

These considerations and the idea that the sky background should be computed together with the star because the two data are naturally coupled, led to the choice of window–sizes as wide as 9 times the FWHM in the case of isolated objects. If the command is faced with a multiple configuration, the window is determined by a frame, 3 times the FWHM width, surrounding the rectangulus circumscribing the Action Radii.

Often new Action Radii, not intersecting the one under examination, fall into the window. These will be ignored during the fit. To visualise this, a "hole", as wide as the relative AR, will be created at the positions of these objects. In this operation some pixels are lost for the fitting, this is the reason for the quite conservative choice of the window size.

After `GROUP/ROMAFOT` has finished, a histogram of the result (groups, objects which failed to group and so on) is prepared printed in the user terminal and in the logfile.

With the default values the user groups a percentage of all the objects in the list. This fraction varies a lot and depends on the crowding, on the seeing, and on the AR. The latter depends again on the photometric limit requested. Typically, somewhere between 70% and 100% of the catalogue list will turn out to be grouped by `GROUP/ROMAFOT`.

In case of the default values for AR the command starts to group the objects holding their original AR. Thereafter, if a group exceeds the maximum number of objects (e.g. 15) GROUP/ROMAFOT tries to prepare an acceptable window $AR(new) = 0.90 \times AR(original)$. This limits the intersections and the groups can turn out to have an acceptable number of components. It is important to note that the size of holes is not affected by this reduction and its original value is conserved.

To group remaining objects, one can execute the command once more with the same catalogue and intermediate table, but with an increased maximum number of objects per window and a reduced AR (in the sense just explained). Here, normally AR reductions exceeding 70% of the original value will produce windows with too many holes, and the final window to fit could contain too few pixels. In this case it is wise to assign a value 100 to the parameter [wnd_perc] in order to provide the fit with enough pixels to compute the sky background.

A situation where a large fraction of the objects are not grouped, even after the AR has been reduced to 70%, can be caused by the following.

1. The photometry is extremely difficult (!).
   If this is the case, the user can only continue to group these objects interactively. To do so he/she should execute the command ANALYSE/ROMAFOT using the command "5" followed by "S" and "R" or in some cases by "C". In case of the latter command he/she is asked for a threshold. Starting from this point and using the command "M" the objects (possibly above the given threshold) which GROUP/ROMAFOT failed to group are presented to the user. Obviously, objects grouped interactively can be added to the same intermediate table generated by GROUP/ROMAFOT.

2. The user is trying to group noise peaks.
   This case can be visualised immediately with the same commands as indicated for case 1. One should remember that, unless the object in question is at the frame edge, it is located at the center of the window presented by ANALYSE/ROMAFOT. The solution, in this case, is to start again with SEARCH/ROMAFOT and with an increased threshold to get a new catalogue list.

3. The AR are too big.
   The same solution (execute SEARCH/ROMAFOT again) can be used in this case. However, the minimum height must now be increased, while the photometric threshold can remain unchanged.

Following these considerations one should still be aware that there are advantages with interactive processing. If, for instance, GROUP/ROMAFOT was successful in grouping 95% of the program objects, it may be worthwhile to look at the remaining 5%. This operation to obtain (as a first approximation) complete photometry can take 10 to 20 minutes. Of course, if one is not interested in completeness, these last objects can be dropped, since their photometry will be poor in comparison with the others.

After having grouped the objects, the user tries to fit the subframes by executing the command FIT/ROMAFOT.

FIT/ROMAFOT

`FIT/ROMAFOT frame [int_tab] [thres,sky] [sig,sat,tol,iter] [meth[,beta]] [fit_opt] [mean_opt]`

Since the user is now at the same point discussed in the interactive procedure, detailed information is not needed here.

The logfile will contain the results of the fit. If the command has found difficulties for more than a few percent of the objects, something is wrong (e.g. wrong PSF parameters). If the statistics are acceptable the user may wish to display some windows with ANALYSE/ROMAFOT (see above).

Normally in this phase, in order to realise and to correct the "bad" windows (NO CONVERGENCY or "more than ... iterations") one should look at them. This is easily accomplished with ANALYSE/ROMAFOT using the commands "4", "S", "O". These commands will give the windows which did not find appropriate convergence on all the stars, or "4", "S", "I" to give windows which needed more than n iterations.

The user can intervene as described in the interactive procedure. If needed, fits can be repeated for these windows to have the situation where all windows have been reconstructed as a sum of a number of elementary components. To check the quality of such reconstruction, a further crossing-over from the automatic path to the interactive path may be done. In fact, `ANALYSE/ROMAFOT` allows the interactive check of all the windows. To perform the automatic check one can execute the command `EXAMINE/ROMAFOT`.

EXAMINE/ROMAFOT

`EXAMINE/ROMAFOT int_tab [hmin,hmax]`

This command uses the quality parameters calculated by `FIT/ROMAFOT` for each object and allows the user to select a threshold to accept or refuse the fit. At the moment, two parameters are calculated by `FIT/ROMAFOT`: the reduced χ^2 and the semi–interquartile interval.

`EXAMINE/ROMAFOT` works on the intermediate table and starts plotting the distributions of these two fit estimators on the graphics terminal or window. A gaussian fit to these two histograms is then performed in order to define the mode and the sigma of the two distributions. At this stage the user can cut the wings to get parameters not influenced by occasional values.

Once the two distributions have been parametrised, a plot of SD versus SIQ appears. In order to detect the objects beyond n times σ of the distribution in this plot immediately, the mode is marked on both axes and the scale is in units of sigma. Using the cursor the user can select objects whose χ^2 exceeds the value given by the vertical cursor position (x–axis) or objects whose SIQ exceeds the quantity corresponding to the horizontal cursor position (y–axis), or both. Commands to perform the operations are described in the help documentation and can be displayed using the command “H”.

Then,`ANALYSE/ROMAFOT` allows an automatic examination of objects flagged by `EXAMINE/ROMAFOT` using the sequence of commands “4”, “S”’, “T” and “D”. Again, if completeness is not required, it is possible just to ignore these objects in the final registration.

After execution of the command `EXAMINE/ROMAFOT` the user is in a position similar to that which he finds at the end of the interactive path. However, checking is now limited to objects flagged by `EXAMINE/ROMAFOT`. To do so one has to execute `ANALYSE/ROMAFOT` using the commands “4”, “S” and “T” (appropriate sets). Then, repeat “D” to examine the next window with one star flagged and, if that is the case, correct it.

5.4.4 Registration of the Results

To conclude both the interactive and the automatic reduction paths the user should fix the results in a MIDAS table.

REGISTER/ROMAFOT

`REGISTER/ROMAFOT int_tab reg_tab [wnd_opt] [obj_opt]`

This command creates a table with the results of the analysis. However, all the photometric information obtained from the frame as the absolute quantities (magnitudes, colour equations etc.) still have to be derived. In this output table every object occupies one row and the data are stored according the Table 5.2.

Column	Name	Description
#1	`:IDENT`	Identification
#2	`:X`	Physical X–coordinate
#3	`:Y`	Physical Y–coordinate
#4	`:INT`	central pixel intensity resulted from the fit
#5	`:LOC_BG`	Local sky background; see text
#6	`:MAG1`	Aperture magnitude no.1; see text
#7	`:MAG2`	Aperture magnitude no.2; see text
#8	`:MAG3`	Aperture magnitude no.3; see text
#9	`:MAG_CNV`	Instrumental magnitude; see text
#10	`:SIGMA`	Sigma of fit function (if not fixed by the user)
#11	`:BETA`	Parameter of the Moffat fit function
#12	`:SIQ`	Semi interquartle resulting from the fit
#13	`:CHI_SQ`	χ^2 resulting from the fit

Table 5.2: Romafot Registration Table

The magnitudes MAG1, MAG2 and MAG3 (e.g. U, B and V) are accepted by the commands `FIND/ROMAFOT` or `SELECT/ROMAFOT` and are necessary for calibration. In case of program stars (search made with `SEARCH/ROMAFOT`), these columns are filled with zeros.

The instrumental magnitude MAG_CNV is calculated as $-2.5 \log VOL$, where VOL is the integral over the elementary surface to fit the star, calculated from $-\infty$ to $+\infty$. In case of linear data this is expected to differ from the magnitude by a constant.

5.4.5 Photometry of the Other Program Frames

The procedure described above is the standard procedure and results in an *independent* reduction of *each* frame. However, there are cases where a different approach is more efficient. Imagine, for instance, that several frames of the same region in the sky had to be reduced but the frames were taken in different seeing conditions, or at telescopes with different pixel matching. It is useful, in this case, to *transfer* the inputs from the frame with the best resolution to the others. The result of this operation is to add information (number of components, for instance) to the frames of poor quality. This procedures obviously applies also to frames taken with different filters.

The operation requires three logical steps:

1. Create a base for transformation of coordinates from frame A to frame B;

2. Transform the coordinates from A to B;

3. Adjust the input (intensity, background, holes and so forth)

These steps are accomplished by the commands CBASE/ROMAFOT, CTRANS/ROMAFOT and ADAPT/ROMAFOT, respectively.

CBASE/ROMAFOT

CBASE/ROMAFOT frame_1 frame_2 [out_tab1] [out_tab2]

This command presents two images at the same time and the user has to use the cursor to mark (a few) objects common to both images. CBASE/ROMAFOT creates two MIDAS tables, defaulted to TRACOO1 and TRACOO2, which will be used by the following command to perform the transformation of coordinates stored in the intermediate table.

CTRANS/ROMAFOT

CTRANS/ROMAFOT int_tab [base_1] [base_2] [pol_deg]

This command uses the two tables generated by CBASE/ROMAFOT and derives the analytical transformation to pass from coordinates on "frame 1" to coordinates on "frame 2". If the transformation is considered satisfactory by the operator (e. g. if the rms is of the order of a few tenths of a pixel or better), the command changes the coordinates on the *intermediate table* in order to use these as inputs in the next program frame.

In practice, after having completed reductions on the first frame, one should copy the intermediate table and transform it. If the transformation produces a rms larger than a few tenths of a pixel, this means that a mismatching exists on the objects selected with CBASE/ROMAFOT. In some cases a high rms could indicate the necessity of an higher polyminal order ("N"), for the analytical transformation, provided that more than

$$\frac{N^2+3N+2}{2}$$

objects are available for the base.

At this point it could be useful to check the transformation with the command ANALYSE/ROMAFOT. In this case the *new* frame and the intermediate table just transformed must be used and the keys *D1, D2... Dk* will allow the display of objects with the old parameters (height, β and so on) on the new image.

With the key *L*, in addition, it is possible to determine the approximate central intensity of the star and the associate sky luminosity. These values are useful to adjust the parameters in input with the command

ADAPT/ROMAFOT

ADAPT/ROMAFOT int_tab [thres] [i_factor] [s_factor] [h_factor] [x_size,y_size]

This is a self-explanatory command. Its use is recommended in order to facilitate the convergency taking into consideration the differences in exposures and seeing conditions between the template and the other program frames.

At this point the user has reached a stage which correspondings to that after executing GROUP/ROMAFOT. In fact, he/she has all the objects organised in *windows* with certain trial values and he/she is ready to execute the command FIT/ROMAFOT.

The user may realise that ADAPT/ROMAFOT does not delete the objects which in the new frame are, for instance, fainter than the photometric threshold: it only flags them. This flag will allow the registration of such objects but with the instrumental magnitude set to "0" in order to keep the sequential correspondence of the objects in different frames. It is certainly wise not to loose such correspondence using the key *"C"* in ANALYSE/ROMAFOT. The key *"T"* should be used instead.

5.4.6 Additional Utilities

DIAPHRAGM/ROMAFOT

`DIAPHRAGM/ROMAFOT frame [regi_tab] [rego_tab] ap_rad`

This command performs aperture photometry at given positions. These positions are read from a registration table created by REGISTER/ROMAFOT. Values of the sky background are read as well.

This command has two fairly different applications. The first application is fast photometry of many objects with the sequence of commands SKY/ROMAFOT, SEARCH/ROMAFOT, DIAPHRAGM/ROMAFOT. The second application is to calibrate a frame transporting standard magnitudes from another frame. Given the importance of a careful determination of the sky background, the previous use of FIT/ROMAFOT is not redundant in this second case. The user is allowed to select the diaphragm over which the integration is performed. Usually a compromise between large apertures (to sample *all* the stellar contribution) and small apertures (to minimise the light from nearby companions) is necessary.

This module creates an output file with the same structure as that created by FIT/ROMAFOT. However, the instrumental magnitude is determined by summing up the pixel intensities above the sky within the diaphragm.

RESIDUAL/ROMAFOT

`RESIDUAL/ROMAFOT in_frame out_frame diff_frame [reg_tab]`

This command produces two images to display the work just done. The first one is the assembly of all the fitted elementary images and the second is the difference between the original frame and the reconstruction produced by ROMAFOT. While the reconstructed image is essentially heuristic, the resulting difference could be useful to visualise maxima not detected by the searching programs. These images can then be examined with FIND/ROMAFOT or SELECT/ROMAFOT.

In order to estimate the internal error introduced by the presence of other images (error for crowding), the following simulation is generally used.

ADDSTAR/ROMAFOT

`ADDSTAR/ROMAFOT in_frame out_frame [reg_tab] [cat_tab] [x_dim,y_dim] [n_sub]`

This command selects one subframe from the original frame and generates an artificial image identical to the primitive but with the quoted window added at given positions.

If this window contains one stellar image, one is able to determine, through a new ROMAFOT session, the ability of the procedure to detect an object and the general reproducibility of the photometry.

A catalogue, similar to that created by `FIND/ROMAFOT`, is generated by `ADDSTAR/ROMAFOT`. This catalogue contains the positions where the subarray has been added and the original instrumental magnitudes in order to allow a ready detection of the differences. Obviously, one single window can be added in several output positions and different windows are generally used in input to sample the behaviour of the errors in luminosity.

The procedure described above results in a multi-reduction of the same frame, because, after having generated the artificial image, the user must go through the commands `SKY/ROMAFOT` → `SEARCH/ROMAFOT` → `GROUP/ROMAFOT` → `FIT/ROMAFOT` → `ANALYSE/ROMAFOT` → `REGISTER/ROMAFOT`. The following command can make the job less time-consuming.

FCLEAN/ROMAFOT

`FCLEAN/ROMAFOT cat_tab inti_tab [into_tab]`

With this command the user can avoid examining all the objects found on the artificial image and, consequently, is able to save a considerable amount of time. `FCLEAN/ROMAFOT` compares the intermediate file created by `GROUP/ROMAFOT` to the catalogue created by `ADDSTAR/ROMAFOT` and selects only the windows which contain an artificial image. This way the number of windows passed to `FIT/ROMAFOT` is drastically reduced and the user can afford a statistically significative simulation by making repeated trials. The final results can be statistically examined with `CHECK/ROMAFOT`.

CHECK/ROMAFOT

`CHECK/ROMAFOT cat_tab reg_tab err_mag`

This command can give an answer to the questions what fraction of artificial frame has been recovered, and to what extent photometry is affected by crowding of frames?

This is accomplished by comparing the instrumental magnitudes and positions (recorded in the catalogue file generated by `ADDSTAR/ROMAFOT`) with that derived through the complete procedure and recorded in the output table. The results are arranged in the form of a histogram for the sake of a synthetical understanding. In addition, `CHECK/ROMAFOT` modifies in the catalogue table the positions setting flag to 1 for the objects that have been recovered, while this flag remains set to 0 for the undetected ones.

5.4.7 Big Pixels

This problem is examined in Buonanno and Iannicola, 1989. In short, ROMAFOT performs the best fit by comparing a given pixel to the value of the PSF at the centre of the pixel. This approximation is valid if the pixel size is small compared with the scale length of the point images. If R is the ratio between the FWHM of the PSF and the pixel size, then a value of $R \sim 1.5$ is the limit of validity of the quoted approximation and, correspondingly, the limit beyond which we enter in the regime of *big pixels*. With the commands below the integral of the PSF over the pixel area is computed via the Gauss - Legendre integration formulae which are characterized by a high precision for a correspondingly small number of subpixels where the PSF must be calculated.

Since the number of subpixels depends on the intensity of the star, on the local gradient of the PSF and so on, two commands are used. The first command prepares a file where the subpixel values are computed according to the different parameters of the problem (aperture of the telescope, exposure time, seeing, required accuracy, magnitude of the star, distance of the pixel from the centre of the star and so forth). The second command performs the non - linear fitting by comparing the integral of the PSF over the subpixel values.

MODEL/ROMAFOT

`MODEL/ROMAFOT [mod_file]`

This command creates a sequence of arrays whose elements are the subpixels required by the Gauss - Legendre formulae to achieve a given precision in computing the integral of the PSF. The precision depends on the user who selects which fraction of the intrinsic noise is acceptable in numerically computing the integral. This data are stored in a file, IW.DAT, and passed to the command for fitting.

MFIT/ROMAFOT

`MFIT/ROMAFOT frame [int_tab] [thres,sky] [sig,sat,tol,iter] [meth[,beta]] [fit_opt] [mean_opt] [pix_file]`

This command is the analogue of FIT/ROMAFOT but in case of big pixels. The difference is that MFIT/ROMAFOT computes an integral and, consequently, the fitting is more time consuming in correspondence of the larger number of subpixels. Its use in case of small pixels is therefore not recommended.

5.5 Command Syntax Summary

Table 5.5 lists the commands for the ROMAFOT package. These commands are activated by setting the ROMAFOT context (by means of the command `SET/CONTEXT ROMAFOT` in a MIDAS session).

C	ommand
ADAPT/ROMAFOT	int_tab [thres] [i_factor] [s_factor] [h_factor] [x_size,y_size]
ADDSTAR/ROMAFOT	in_frame out_frame [reg_tab] [cat_tab] [x_dim,y_dim] [n_sub]
ANALYSE/ROMAFOT	frame [cat_tab] [int_tab] [sigma,sat]
CBASE/ROMAFOT	frame_1 frame_2 [out_tab1_1] [out_tab2]
CHECK/ROMAFOT	cat_tab reg_tab err_mag
CTRANS/ROMAFOT	int_tab [base_1] [base_2] [pol_deg]
DIAPHRAGM/ROMAFOT	frame [regi_tab] [rego_tab] ap_rad
EXAMINE/ROMAFOT	int_tab [hmin,hmax]
FCLEAN/ROMAFOT	cat_tab inti_tab [into_tab]
FIND/ROMAFOT	frame [cat_tab]
FIT/ROMAFOT	frame [int_tab] [thres,sky] [sig,sat,tol,iter]
	[meth[,beta]] [fit_opt] [mean_opt]
GROUP/ROMAFOT	frame [area] [cat_tab] [int_tab] [thres] [wnd_max]
	[end_rad,sta_rad] [wnd_perc]
MFIT/ROMAFOT	frame [int_tab] [thres,sky] [sig,sat,tol,iter]
	[meth[,beta]] [fit_opt] [mean_opt] [mod_file]
MODEL/ROMAFOT	[mod_file]
REGISTER/ROMAFOT	int_tab reg_tab [wnd_opt] [obj_opt]
RESIDUAL/ROMAFOT	in_frame out_frame diff_frame [reg_tab]
SEARCH/ROMAFOT	frame [sky_tab] [cat_tab] [area] [psf_par] [thresh] [height]
SELECT/ROMAFOT	frame [int_tab] [wnd_size]
SKY/ROMAFOT	frame [sky_tab] [area] [nrx,nry]

Table 5.3: ROMAFOT Command List

Bibliography

[1] Newell, E.B., and O'Neil, E.J.: 1974, in *Electronography and Astronomical Observations*, Eds. G.L. Chincarini, P.J. Griboval, and H.J. Smith, University of Texas, p. 153

[2] Van Altena, W.F., Auer, L.H.: 1975, in *Image Processing Techniques in Astronomy*, Eds. C. de Jager and H. Nieuwenhuyzen, Reidel: Dordrecht, p. 411

[3] Butcher, H.: 1977 *Astrophys. J.* **216**, 372.

[4] Herzog, A.D., Illingworth, G.: 1977, *Astrophys. J. Suppl.* **33**, 55

[5] Chiu, L.-T.G.: 1977, *Astron. J.* **82**, 842

[6] Auer L.H., Van Altena, W.F.: 1978, *Astron. J.* **53**, 83

[7] Buonanno R., et al.: 1979, in *International Workshop on Image Processing in Astronomy*, Eds. G. Sedmak, M. Capaccioli, R.J. Allen, Trieste, p. 354

[8] Buonanno, R., et al.: 1983, Astron. Astrophys. **126**, 276

[9] Stetson, P.B.: 1979, Astron. J. **84**, 1056

[10] Stetson, P.B.: 1979, Astron. J. **84**, 1149

[11] Bragaglia, A. et al.: 1986, in *The Optimization of the Use of CCD Detectors in Astronomy*, Eds. J.-P. Baluteau and S. D'Odorico, Garching: ESO pag. 203

[12] Buonanno, R. and Iannicola, G.:1989, *Publ. Astron. Soc. Pacific*, In print

Chapter 6

Long-Slit and 1D Spectra

6.1 Introduction

This chapter describes the long-slit reduction package in a general way. One-dimensional spectra are considered to be a particular case of long-slit spectra and are also handled by the package. More instrument-specific operating instructions may be given in appendices to this MIDAS manual or in the relevant ESO Instrument Handbooks.

The package provides about 50 basic commands to perform the calibration and to display the results, defined in the context `LONG`. A tutorial procedure, `TUTORIAL/LONG`, illustrates how to operate the package. The context `SPEC` is a low-level context including general utility commands required by the different spectroscopy packages `ECHELLE`, `IRSPEC` and `LONG`. These commands are refered to and summarized in this chapter. The graphical user interface `XLong` is a MOTIF-based interface providing an easy access to the commands of the package. This interface is described in the Appendix G. Standard spectral analysis can be performed with the graphical user interface `XAlice` (command `CREATE/GUI ALICE`). The use of general MIDAS commands for spectral analysis is also discussed in this chapter.

The main characteristics of the `LONG` context are derived from its modularity, that allows to perform, in a first step, the image cosmetics and photometric corrections – which are detector dependent – and then, to correct for the geometric distortions. The method gives particular emphasis to an accurate wavelength calibration and correction of all distortions along the whole slit, so that the spatial structure of extended objects can be examined in detail.

Although the method is applicable to different instrumental configurations, we assume, in the following description, that the generic instrument consists of a spectrograph coupled to a CCD detector. In the current version it is assumed that the CCD frame is oriented with the rows in the dispersion direction and the columns along the slit.

Note

This chapter only provides a synopsis of the commands needed for the reduction of long-slit. It is important to realise that it can neither substitute the `HELP` *information available for each command nor be exhaustive, especially not with regard to the usage of general utilities such as the MIDAS Table System, the*

AGL Graphics package, etc. The Appendix G provides a practical approach to the reduction of long-slit spectra.

6.2 Photometric Corrections

6.2.1 Detector Non-Linearity

This section outlines the transformation from raw to radiometrically corrected data. For the purpose of this simple description we make a distinction between linear and non-linear detectors. Data acquired with detectors of the latter category require, at most, bias and dark signal subtraction plus division by a flat field exposure.

Most real detectors have nonlinear Intensity Transfer Functions (ITF), *e.g.* photographic plates, the effects of the dead time at high count rates in photon counting systems and the low light level nonlinearity of Image Dissector Scanners (see M. Rosa, *The Messenger*, **39**, 15, 1985). If the ITF is known analytically, the command `COMPUTE/IMAGE` will be sufficient for the correction of the raw data. For example, the paired pulse overlap (dead-time) correction of photoelectric equipment could be corrected for by `COMPUTE/IMA OUT = IN/(1+TAU*IN)`, where `TAU` is the known time constant of the counting electronics an `IN` must be in units of a count rate rather than total counts. If the ITF is defined in tabular form, the command `ITF/IMAGE` can be used to obtain the ITF correction for each image element (pixel value) by interpolation in an ITF table; this command assumes a uniform transfer function over the image field.

CCDs are generally more nearly linear than are most other detectors used in astronomy. However, especially in particular pixels or regions, CCDs are clearly non-linear and/or suffer deviating signal zero points. Procedures which may be useful for the treatment of such deficiencies are (partly) described in Appendix B.

Problems related to background estimation and more sophisticated flat-field corrections generally are very instrument dependent. Therefore, it is not possible to give one standard recipe here; check the various instrumental appendices for more specific advice.

6.2.2 Removing Cosmic Ray Hits

If two or more spectra have been taken under the same conditions, a very efficient way of removing particle hits from the raw data is to replace pixel values with large deviations from other observations with a local estimator, as done in the command `COMBINE/LONG`.

If multiple images are not available, cosmic ray hits have to be filtered out by a different method. The main requirement is that the image is **not** changed where there are no hits, especially on the area covered by the object under study. This can for instance be done with the command:

```
FILTER/MEDIAN inframe outframe 0,3,0.3 NR subframe
```

where `subframe` defines the region to filter. But you may wish to try other parameters and parameter values. This step is done twice to remove the hits on the sky on each side of the object.

Those cosmic ray hits in the area surrounding the object spectrum, which are particularly disturbing, were removed one by one using `MODIFY/AREA ? ? 1`, and working with the cursor on a zoomed display. Particular care must be exercised here, in order to modify only the few pixels affected by cosmic rays. For point sources and extended sources with very smoothly varying spatial profiles, `FILTER/MEDIAN` may also be tried. However, in order to preserve the instrumental profile the filter width along the dispersion axis *must* be set to 0.

Another possibility is to use the command `FILTER/COSMIC`. The algorithm detects hot pixels from a comparison of the pixel value with the ones of the neighbouring pixels. It rejects every 'feature' that is more strongly peaked than the point spread function.

Regardless of the method used, it is necessary to check the performance of the filter by careful inspection of the difference between raw and filtered image which can be computed as `COMPUTE/IMA TEST = RAW-FILTERED`.

6.2.3 Bias and Dark Subtraction

The dark signal is exposure time dependent. The one appropriate for a given observation has to be derived by interpolation between dark frames of different exposure times. Note that long dark exposures first have to be cleaned of particle hits (see above). Bias and dark signal will usually be determined and subtracted in the same step.

In a CCD with good cosmetic properties, bias and dark signal are the same for all pixels. Then, only a number (obtained from, *e.g.*, `STATISTICS/IMAGE`) should be subtracted in order to prevent the noise in the dark frames to propagate into the reduced spectra. If, because of local variations, this is not possible, some suitable smoothing should still be attempted.

6.2.4 Flat-Fielding

The lamps used for flat fields have an uneven spectral emissivity. Combined with the uneven spectral sensitivity of the spectrograph and detector, this results in flat fields which usually have very different intensities in different spectral regions. Therefore a direct flat-fielding, *i.e.*, division by the original dark subtracted flat-field, produces spectra which are artificially enhanced in some parts and depressed in others. In principle, this should not be a problem because the instrumental response curve should include these variations and permit them to be corrected. However, the response curve is established by integration over spectral intervals of small but finite width. At this stage, strong gradients obviously introduce severe problems. Also the wish, at a later stage to evaluate the statistical significance of features argues strongly against distorting the signal scale in a basically uncontrolled way.

Therefore, it is better first to remove the low spatial frequencies along the dispersion axis (but not perpendicular to it!) from the flat-field, using the command `NORMALIZE/FLAT`. This command first averages the original image along the slit (assumed to be along the Y-axis) and fits the resulting one-dimensional image by a polynomial of specified degree. Then the fitted polynomial grows to a two-dimensional image and divides the original flat-field by this image to obtain a "normalized" flat-field.

In some cases, *e.g.*, EFOSC in its B300 mode which gives a very low intensity flat-field at the blue end, the polynomial fit is not satisfactory and it is advisable to do the sequence manually. Instead of fitting a polynomial one could fit a spline using the command `INTERPOLATE/II` or `NORMALIZE/SPECTRUM`. The latter belongs to the low-level context `SPEC` and is interactively operated on a graphical display of the spectrum.

6.3 Geometric Correction

An accurate two dimensional geometric correction over the entire frame is an important part of the whole reduction process. It strongly affects velocity measurements along the slit and is very critical even for point sources if narrow night sky lines shall be properly subtracted. If the spectrum contains only a point source the user may prefer to extract it from the 2D image (`EXTRACT/LONG`) and then proceed with the normal one-dimensional spectral reduction.

On the other hand, if the spatial information along the slit is of no interest and night sky lines do not have to be corrected for, much time can be saved by properly averaging the signal along the slit (`EXTRACT/AVERAGE`).

The full geometrical correction of long-slit spectra is a transformation from the raw pixel coordinates (X, Y) to the sampling space (λ, s), where λ is the wavelength and s is an angular coordinate in the sky along the slit. Logically, it often makes sense to separate the geometrical transformation into two orthogonal components: the dispersion relation $\lambda = \lambda(X, Y)$, which can be obtained from an arc spectrum with a fully illuminated slit and the distortion along the slit $s = s(X, Y)$, which can be derived from the continuum spectra of point sources. Practically, these two transformations should, whenever possible, be combined into one before rectifying the data, because this saves one non-linear rebinning step, each of which necessarily leading to some loss of information.

As far as the reduction is concerned, the easiest way to achieve this is to observe the comparison lamp used for wavelength calibration through a a pin-hole mask. (Of course, this method can account only for instrumental distortions, not for differential atmospheric refraction, *etc.*)

In the presence of strong distortions along the slit, a 2-D modeling must be attempted and the command `RECTIFY/LONG` could be considered. If distortions along the slit can be neglected or suitably corrected for in a separate step, a 2-D modeling of the dispersion relation is still a valid approach. However, a separate reduction of detector row after detector row along the slit, then, often is a superior alternative. A broad overview of the major options is given in the next subsections; **a detailed comparison and a Cookbook for the usage of the two methods are provided in Appendix G.**

6.3.1 Detecting and Identifying Arc Lines

For optimum results, it is important to use a comparison spectrum with as high a signal-to-noise ratio of the lines as possible. It is therefore advisible to flat-field also the arc spectrum in order to correct for small-scale fluctuations. Another good idea is to filter the frame along the slit using the command FILTER/MEDIAN with a rectangular filter window of one pixel in the dispersion direction and several pixels in the perpendicular direction. For the row-by-row method one could also consider smoothing the spectra along the slit axis which would provide for a stronger coupling between neighbouring rows and thereby yield a solution which is intermediate between the extremes presented by the two pure methods.

- SEARCH/LONG: Finds the positions of reference lines in world coordinates. Positions are by default estimated by the center of fitted Gaussians. Other centering methods are available (Gravity, Maximum) but could result in systematic position errors (See Hensberge & Verschueren, Messenger, **58**, 51). The results are stored in a table called line.tbl. The parameter YWIND corresponds the half-size of the row averaging window applied to adjacent rows of the spectrum for an improvement of the signal to noise ratio. The parameter YSTEP controls the step in rows between successive arc line detections. The value YSTEP=1 corresponds to the default row-by-row method and larger values can be used to get a quicker calibration. The algorithm detects lines whose strength exceeds a certain threshold (parameter THRES) above the local background. The local background results from a median estimate performed on a sliding window which size is controlled by the parameter WIDTH. The command PLOT/SEARCH allows to check the results at this stage. Note that for a two-dimensional spectrum, both options 1D and 2D can be used (See HELP PLOT/SEARCH).
- The command IDENTIFY/LONG allows an initial interactive identification, by wavelength, of some of the detected lines. Spectral line atlas are provided in the instrument operating manuals. The command PLOT/IDENT visualizes the interactive identifications.

6.3.2 Getting the Dispersion Solution

The command CALIBRATE/LONG approximates the dispersion relation for each searched row of the arc spectrum. The algorithm can be activated in different modes, controlled by the parameter WLCMTD:

- The mode IDENT must be used if the lines have been identified interactively using IDENTIFY/LONG.
- The mode GUESS allows a previously saved session to be used (command SAVE/LONG) for the determination of the dispersion relation. The two observation sets must in principle correspond to the same instrumental set-up. Limited shifts (up to 5 pixels) are taken into account by a cross-correlation algorithm. The name of the reference session must be indicated by the parameter GUESS.
- It is also noteworthy to indicate the presence of a mode LINEAR introduced in the package for the purpose of on-line calibration. This mode is still **in evaluation phase** and allows the results of the command ESTIMATE/DISPERSION to be taken into account in the calibration process.

The command CALIBRATE/LONG provides the following results:

- The coefficients of the dispersion relation for each searched row of the two-dimensional spectrum are computed as a series of polynomials $\lambda = p_y(x)$. The results are written in the table coerbr.tbl.
- The starting, final and average step wavelength of the spectrum are estimated and written in the keywords REBSTRT, REBEND, REBSTP used by the commands REBIN/LONG and RECTIFY/LONG.
- A two-dimensional dispersion relation of the form $\lambda = \lambda(X, Y)$ is estimated if the parameter TWODOPT is set to YES. This bivariate dispersion relation is necessary to use the command RECTIFY/LONG. The resulting dispersion coefficients are stored in the keyword KEYLONGD.

The command CALIBRATE/TWICE performs a two-pass determination of the dispersion relation. In a first pass, the lines are identified by a standard CALIBRATE/LONG. Only the lines which have consistently identified at all rows are selected for the second pass, which then performs a new calibration on a stable set of arc lines. If after selection a good spectral coverage of the arc spectrum is secured, this method provides very stable estimates of the dispersion relation.

The command PLOT/CALIBRATE visualizes the lines found by the calibration process. The dispersion curve and the lines that were used to determine it are presented by PLOT/DELTA. Residuals to the dispersion curve are plotted by PLOT/RESIDUAL. For two-dimensional spectra, the command PLOT/DISTORTION can be used to check the stability of the dispersion relation along the slit.

The iterative identification loop consists of estimating the wavelength of all lines in the arc spectrum and associate them to laboratory wavelengths to refine the estimates of the dispersion relation. The line identification criterion will associate a computed wavelength λ_c to the nearest catalog wavelength λ_{cat} if the residual:

$$\delta\lambda = |\lambda_c - \lambda_{cat}|$$

is small compared to the distance of the next neighbours in both the arc spectrum and the catalog:

$$\delta\lambda < min(\delta\lambda_{cat}, \delta\lambda_c) * \alpha$$

where $\delta\lambda_{cat}$ is the distance to the next neighbour in the line catalog, $\delta\lambda_c$ the distance to the next neighbour in the arc spectrum and *alpha* the tolerance parameter. Optimal values of α are in the range $0 < \alpha < 0.5$. The tolerance value is controlled by the parameter ALPHA.

Lines are identified in a first pass without consideration of the rms of the residual values by an iterative loop controlled by the parameter WLCNITER. The residuals for each line are then checked in order to reject outliers which residual is above the value specified by the final tolerance parameter TOL. The degree of the polynomials is controlled by the parameter DCX and the iterative loop is stopped if residuals are found to be larger than MAXDEV.

6.3.3 Distortion Along the Slit

The distortion along the slit $s = s(X, Y)$ can be modelled using an image containing one or several spectra of (ideally: point-) sources with well defined continuum. The command `SEARCH/LONG` can be used to generate a table with the positions of the spectra at several wavelengths. (Contrary to the general conventions used throughout this chapter, for this particular application the dispersion direction must be parallel to the 'Y'-axis!) The distortion is then modelled by a two-dimensional polynomial fitted to these positions (using, *e.g.*, `REGRESSION/TABLE`). If the resulting coefficients are stored in the keyword `COEFY*, (*=I,R,D)` and combined with suitable (if necessary: dummy) coefficients for the dispersion resolution in the keyword `KEYLONG*, (*=I,R,D)`, the command `RECTIFY/LONG` can be applied.

Note that these steps are not implemented as a convenient-to-use high-level command procedure.

6.3.4 Resampling the Data

In the standard row-by-row option, the dispersion coefficients are kept in the table (`coerbr.tbl`). The rebinning to constant step in wavelength is accomplished, row by row, with the command `REBIN/LONG`.

If the 2-D option described above for the solution of the dispersion relation is followed (`TWODOPT=YES`), the polynomial coefficients in both directions are stored in the keywords `KEYLONGD` and `COEFYD`, respectively. This information is then used by the command `RECTIFY/LONG` to resample the image in the (λ, s) space.

Data resampling can be avoided by the command `APPLY/DISPERSION` which generates a table with the columns `:WAVE` and `:FLUX`, each row corresponding to the central wavelength and flux of a CCD detector pixel.

6.4 Sky Subtraction

As stated before, sky subtraction can be a very critical step in the reduction of long slit spectra with the main problem being the curvature of the lines along the slit (due to both misalignment of CCD and spectrum and residual optical distortions).

Although one may intuitively tend to subtract the sky spectrum still in pixel space in order to avoid the problems inherent to non-linear rebinning, experience shows that a proper wavelength calibration can remove the curvature of the sky lines to a high degree of accuracy (one should aim for ± 0.1 pixels rms).

The command `SKYFIT/LONG` makes a polynomial fit to the sky in two windows above and below the object spectrum, either with one single function for the full length of the spectrum (`mode = 0`) or with one function for every column (`mode = 1`). `Mode 0` is recommended for the spectral regions where the sky is faint, because usually there is not enough signal to achieve a meaningful fit for every column. The same is not true for the bright sky lines, where `mode 1` helps dealing with the variable line width and with residual line curvature.

For this reason, it often is best to prepare two sky spectra first: `sky0` fitted obtained in `mode 0` and with a polynomial of degree 0-2 fitted to windows with "clear" sky, and `sky1` derived with `mode 1` and multiple polynomials of degree 2-4 on windows going as close as possible to the object. The final sky spectrum, (`sky`), to be subtracted from the object is obtained from a combination of the two sky spectra and essentially consists of `sky0` which only for the bright sky lines has been replaced with `sky1`. Such a combination can be prepared by mean of the command `REPLACE/IMAGE` (e.g., `REPLACE/IMA sky0 sky 120.,>=sky1`).

6.5 Flux Calibration

The commands `EXTINCT/LONG` followed by `RESPONSE/FILTER` or by `INTEGRATE/LONG` and `RESPONSE/LONG` permit a one-dimensional response curve `response.bdf` to be obtained from the extracted (*e.g.*, via `EXTRACT/LONG` or `EXTRACT/AVERAGE`) and wavelength calibrated spectrum of a standard star.

The command `RESPONSE/FILTER` divides the standard star spectrum by the flux table values and uses median and smooth filterings (parameters `FILTMED` and `FILTSMO`) to smooth the instrumental response function.

The command `INTEGRATE/LONG` also performs a division of the standard star spectrum by the flux table but generates an table of response values at a wavelength step equal to the one of the flux table. The command `RESPONSE/LONG` interpolates these values using a polynomial or spline interpolation scheme.

This response curve can be applied to a one- or two-dimensional extracted, wavelength calibrated and extinction corrected (`EXTINCTION/LONG`) spectrum by the command `CALIBRATE/FLUX`.

Verification commands include `PLOT/FLUX` to visualize the standard star reference flux table and `PLOT/RESPONSE` to vizualize the final instrumental response function.

6.5.1 Flux Calibration and Extinction Correction

To calibrate the chromatic response, observations of a standard star (preferably more than one) are needed, for which the absolute fluxes are known. The spectral response curve of the instrument can then be determined with the command `RESPONSE/LONG`, and absolute fluxes for the objects of interest are obtained with `CALIBRATE/FLUX`. The extinction correction must previously be done in a separate step with the command `EXTINCTION/LONG`.

An alternative procedure, if no standard star spectrum is available, is the normalisation of the continuum as described below.

6.5.2 Airmass Calculation

The commands in the previous Section require the airmass as input parameter. Some instrument/telescope combinations provide raw data files with the proper values of right ascension, declination, siderial time, geographical latitude, duration of measurement and eventually even "mean" airmass. In most cases it will however be necessary to compute an appropriate airmass using the `COMPUTE/AIRMASS`. Refer to `HELP COMPUTE/AIRMASS` for the details of the image descriptors which are needed. Otherwise, the required information must be provided by the user on the command line. It is important to keep in mind that "mean airmass" and "mean atmospheric extinction correction" are different from the values at mid-exposure, especially for larger zenith distances. This is so because the airmass depends non-linearly on zenith distance ($\sec z$) and extinction corrections depend non-linearly on airmass ($10^{-(0.4\times \mathrm{airm}\times \mathrm{ELAW(mag)})}$). For reasonable combinations of exposure time and zenith distance, the weighted mean airmass supplied by `COMPUTE/AIRMASS` should be appropriate.

6.6 Spectral Analysis

6.6.1 Rebinning and Interpolation

Raw spectra are usually not sampled at constant wavelength or frequency steps, and sometimes even with gaps in between the bins. At some stage the independent variable will have to be converted into linear or non-linear functions of wavelength or frequency units and gaps will have to be filled with interpolated values. Frequent cases are: wavelength calibration, redshift correction, $\log(F_\lambda)$ versus $\log(\lambda)$ presentation, and the comparison of narrow-band filter spectrophotometry with scanner data. Related commands are `REBIN/LONG`, already described, `REBIN/LINEAR` for linear rebinning, i.e. scale and offset change, `REBIN/II (IT,TI,TT)` to do nonlinear rebin conserving flux (see below) and `CONVERT/TABLE` to interpolate table data into image data.

Note that in our implementation we make a conceptual difference between straightforward interpolation and rebinning. `REBIN/II (IT, TI, TT)` redistributes intensity from one sampling domain into another. There is no interpolation across undefined gaps and no extrapolation at the extremities of the input data. If you need these, you will have to manipulate the input data first (generating non-existent information !). `REBIN/II (IT, TI, TT)` conserves flux locally and globally.

6.6.2 Normalization and Fitting

A frequently used procedure, alternative to the correction for the chromatic response, is to normalise the continuum to unity by dividing the observed spectrum by a smooth approximation of its continuum. This approximation can be obtained either interactively with the graphic cursor, or from a table (command `NORMALIZE/SPECTRUM`) or by dividing the raw data by itself after filtering or smoothing. Median filtering and running average algorithms are well suited for this purpose (command `FILTER`). A spline fit can be made to the points defined interactively as well as to the filtered data (command `CONVERT/TABLE`). The MIDAS Fitting Package permits to go further and perform a more advanced modelling of the continuum.

6.6.3 Convolution and Deconvolution

Estimating the instrumental point spread function and correcting the observed spectra for the instrumental profile by deconvolution is a delicate subject. Here, the available tools are only briefly mentioned. The point spread function (PSF) can be estimated from the observations (*e.g.*, from the profile of suitable lines in the comparison, night sky, or interstellar medium spectrum), possibly using the fitting package if a noise-free, analytical approximation is desired. Once this is done, the command `DECONVOLVE/IMAGE` can be applied to deconvolve the observed data. Conversely, the convolution of, *e.g.*, a synthetic spectrum with the PSF can be done with the command `CONVOLVE`.

6.6.4 Other Useful Commands

Interactive continuum determination can be done with the command `NORMALIZE/SPECTRUM` as mentioned above. There are commands to measure the position of spectral features. `CENTER/MOMENT` determines positions from moments whereas `CENTER/GAUSS` fits a Gaussian profile; for very sparsely sampled point spread functions, `CENTER/UGAUSS` is preferred. Optionally, all positions can be stored in a working table file. Measurements of line strengths and equivalent widths can be obtained with the command `INTEGRATE/LINE`. Unwanted spectral features (*e.g.*, spikes due to particle events) can be interactively removed with `MODIFY/GCURSOR`. The commands `GET/GCURSOR` and `CONVERT/TABLE` provide means to generate artificial images from cursor positions.

The fitting package permits the determination of line parameters and additional modelling of the data.

6.7 Auxiliary Data

Wavelength calibration and absolute flux determination procedures require some auxiliary data. Some tables are provided by the system, but the user can modify or include new information using the available table commands (Vol. A, Chapter 5). In addition to columns of interest to the user, user-supplied tables *must* have the columns and column labels required by the various MIDAS commands they are to be used with.

For the purpose of the reduction of spectral data, there are three basic table structures:

a) Line identification tables, used in the wavelength calibration step. One column must contain the laboratory (if the reference frame is to be used) wavelengths in the units desired for the calibrated spectrum. A sample table with comparison lines of the spectrum He-Ar is available in `MID_ARC:hear.tbl` it contains lines in the range from 3600 Å to 9300 Å and is suitable for data with dispersions between about 5 and 10 Å/mm.

b) Flux tables, used for flux calibration and rectification of spectra. These tables contain at least three columns defining for each entry central wavelength, bin width and flux, with labels `:WAVE`, `:BIN_W`, `:FLUX_W`. When composing your own tables, take care that the wavelength unit is the same for all three quantities and agrees with the one used for the wavelength calibration. Some sample flux tables are available in the directory `MID_STANDARD`. If you are still preparing your observations, note that the data available for the various stars is very inhomogeneous whereas the quality of the calibrations depends very sensitively on the number and spacing of entries within the spectral range to be observed.

c) Sample atmospheric and interstellar extinction laws are contained in directory `MID_EXTINCTION`. All interstellar laws are normalised to $A_v/E(B-V) = 3.1$ at 5500 Å, and interpolated and rebinned to a constant step in wavelength (10 Å or nm). Note that the wavelength coverage is very different for various data sets.

Filename	Ref.	Filename	Ref.	Filename	Ref.
BD2015	(2)	GD190	(3)	HR4963	(4)
HR5573	(4)	HR6710	(4)	HR8414	(4)
HR8518	(4)	HR8747	(4)	HZ14	(3)
HZ4	(3)	HZ7	(3)	L745X4	(3)
L870X2	(3)	L930X8	(3)	L970X3	(3)
LB227	(3)	LDS235	(3)	LDS749	(3)
LT3218	(1)	LT7987	(1)	W485A	(3)

(1) Stone, Baldwin, 1983, *M.N.R.A.S.*, **204**, 347
(2) Stone, 1977, *Ap. J.*, **218**, 767
(3) Oke, 1974, *Ap. J. Suppl.*, **27**, 21
(4) Breger, 1976, *Ap. J. Suppl.*, **32**, 7
Flux units are 10^{-16}ergs/s/cm^2/ Å for refs 1, 2 and 3
and 10^{-13}ergs/s/cm^2/ Å for ref 4

Table 6.1: Standard Stars for Absolute Flux Calibration in system area MID_STANDARD.

Filename	Description	column number	Ref.
INSTEXAN	Interstellar: Galaxy(Å)	2	(1)
INSTEXAN	Interstellar: LMC (Å)	3	(2)
INSTEXAN	Interstellar: SMC (Å)	4	(3)
INSTEXAN	Interstellar: LMC (Å)	5	(4)
INSTEXAN	Interstellar: Galaxy fit (Å)	6	(5)
INSTEXAN	Interstellar: Galaxy fit (Å)	7	(6)
INSTEXAN	Interstellar: LMC fit (Å)	8	(6)
ATMOEXAN	Atmospheric (Å)	2	(7)
INSTEXNM	Interstellar (nm) †	-	(1-6)
ATMOEXNM	Atmospheric (nm) ‡	2	(7)

(1) Savage, Mathis, 1979, *An.Rev.Astr.Ap.*, **17**, 73
(2) Nandy *et al*, 1981, *MNRAS*, **196**, 955
(3) Prevot *et al*, 1984, *Astron.Astrophys.*, **132**, 389
(4) Koornneef, Code, 1981, *Ap. J.*, **247**, 860
(5) Seaton, 1979, *M.N.R.A.S.*, **187**, 73P
(6) Howarth, 1983, *M.N.R.A.S.*, **203**, 301
(7) Tüg, 1977, *Messenger*, **11**
†The same as INSTEXAN but wavelengths in nanometers
‡The same as ATMOEXNM but wavelengths in nanometers

Table 6.2: Extinction Tables in directory MID_EXTINCTION

6.8 Command Summary

This section summarizes the commands of the contexts LONG and SPEC as well as general MIDAS commands that can be used for spectral analysis. These latter commands are described in Vol. A, Chapter 5 (MIDAS Tables) and Chapter 8 (Fitting of Data).

- Context LONG	
APPLY/DISPERSION	in out [y] [coef]
BATCH/LONG	
CALIBRATE/FLUX	in out [resp]
CALIBRATE/LONG	[tol] [deg] [mtd] [guess]
CALIBRATE/TWICE	
CLEAN/LONG	
COMBINE/LONG	cat out [mtd]
EDIT/FLUX	[resp]
ERASE/LONG	
ESTIMATE/DISPERS	wdisp wcent [ystart] [line] [cat]
EXTINCTION/LONG	in out [scale] [table] [col]
EXTRACT/AVERAGE	in out [obj] [sky] [mtd]
EXTRACT/LONG	in out [sky] [obj] [order,niter] [ron,g,sigma]
GCOORD/LONG	[number] [outtab]
GRAPH/LONG	[size] [position] [id]
HELP/LONG	[keyword]
IDENT/LONG	[wlc] [ystart] [lintab] [tol]
INITIALIZE/LONG	[session]
INTEGRATE/LONG	std [flux] [resp]
LINADD/LONG	in w,bin [y] [mtd] [line] [out]
LOAD/LONG	image [scale_x,[scale_y]]
MAKE/DISPLAY	
NORMALIZE/FLAT	in out [bias] [deg] [fit] [visu]
PLOT/CALIBRATE	[mode]
PLOT/DELTA	[mode]
PLOT/DISTORTION	wave [delta] [mode]
PLOT/FLUX	[fluxtab]
PLOT/IDENT	[wlc] [line] [x] [id] [wave]
PLOT/RESIDUAL	[y] [table]
PLOT/RESPONSE	[resp]

Table 6.3: Commands of the context LONG

- Context LONG	
PLOT/SEARCH	[mode] [table]
PLOT/SPECTRUM	table
PREPARE/LONG	in [out] [limits]
REBIN/LONG	in out [start,end,step] [mtd] [table]
RECTIFY/LONG	in out [reference] [nrep] [deconvol_flag] [line]
REDUCE/INIT	partab
REDUCE/LONG	input
REDUCE/SAVE	partab
RESPONSE/FILTER	std [flux] [resp]
RESPONSE/LONG	[plot] [fit] [deg] [smo] [table] [image] [visu]
SAVE/LONG	[session]
SEARCH/LONG	[in] [thres] [width] [yaver] [step] [mtd] [mode]
SELECT/LINE	
SET/LONG	key=value [...]
SHOW/LONG	[section]
SKYFIT/LONG	input output [sky] [degree] [mode] [g,r,t] [radius]
TUTORIAL/LONG	
VERIFY/LONG	file mode
XIDENT/LONG	[wlc] [ystart] [lintab] [tol]

Table 6.4: Commands of the context LONG (continued)

- Context SPEC	
CORRELATE/LINE	table_1 table_2 [pixel] [cntr,tol,rg,st] [pos,ref,wgt] [ref_value] [outima]
EXTINCTION/SPECTRUM	inframe outframe scale [table] [col]
FILTER/RIPPLE	frame outframe period [start,end]
MERGE/SPECTRUM	spec1 spec2 out [interval] [mode] [var1] [var2]
NORMALIZE/SPECTRUM	inframe outframe [mode] [table] [batch_flag]
OVERPLOT/IDENT	[table] [xpos] [ident] [ypos]
PLOT/RESIDUAL	[table]
SEARCH/LINE	frame w,t[,nscan] [table] [meth] [type]

Table 6.5: Commands of the context SPEC

- Spectral Analysis	
CENTER/method	GCURSOR table EMISSION/ABSORPTION
COMPUTE/FIT	output = function[(refima)]
COMPUTE/FUNCTION	output = function[(refima)]
CONVERT/TABLE	image = table indep dep refimage method
CONVOLVE	input output psf
CREATE/GUI ALICE	
DECONVOLVE	input output psf
FIT/IMAGE	niter,chisq,relax image [function]
GET/GCURSOR	table
INTEGRATE/LINE	image [y0] [x0,x1] [nc,degree] [type]
MODIFY/GCURSOR	image [y0] [x0,x1] [nc,degree] [type]
REBIN/II	input output
REBIN/LINEAR	input output
STATISTICS/IMAGE	image

Table 6.6: Spectral Analysis Commands

6.9 Parameters

For the storage of control parameters and results, the context `LONG` uses a number of special keywords. They are intialised by the command `SET/CONTEXT LONG`, their values can at any time be listed by typing `SHOW/LONG`. The following table provides a brief description of the purpose of the `LONG` keywords:

Parameter	Description
ALPHA	Rejection parameter for lines matching [0,0.5]
AVDISP	Average dispersion per pixel
BETA	Non-linearity in mode LINEAR
BIAS	Bias image or constant
BIASOPT	Bias Correction (YES/NO)
COERBR	Table of coefficients for RBR
COMET	Combination method (AVERAGE/MEDIAN)
COORFIL	Name of coords table of GCOORD/LONG
COROPT	Computes correlation
CORVISU	Plots correlation peak
DARK	Dark image or constant
DARKOPT	Dark Correction (YES/NO)
DCX	fit degree of the dispersion coeff.
DISPCOE	Dispersion coefficients
EXTAB	Extinction Table
EXTMTD	Extraction method (AVERAGE, LINEAR)
EXTOPT	Extinction Correction Option (YES/NO)
FDEG	Flat fitting degree
FFIT	Flat fitted function
FILTMED	Radius of median filter
FILTSMO	Radius of smoothing filter
FITD	Degree of fit
FITD	fit degree of the dispersion coeff.
FITYP	Type of fit (POLY, SPLINE)
FLAT	Flat-Field Image
FLATOPT	Flat Correction (YES/NO)
FLUXTAB	Flux Table of the standard star
FVISU	Visualisation flag (YES/NO)
GAIN	Gain (e-/ADU)
GUESS	Guess session name
IMIN	lower limit from LINCAT
INPNUMB	Input generic name
INPUTF	Input generic name
INSTRUME	instrument
LINCAT	line catalogue
LINTAB	Table of line identifications
LOWSKY	Lower, upper row number of lower sky
MAXDEV	Maximum deviation (pixels)
NITER	Number of iterations
NPIX	size of the raw images in pixels
OBJECT	Lower, upper row number of object spectrum
ORDER	Order for optimal extraction

Table 6.7: Keywords Used in Context `LONG`

Parameter	Description
OUTNUMB	Output starting number
OUTPUTF	Output generic name
PLOTYP	Type of plot (RATIO, MAGNITUDE)
RADIUS	Radius for cosmics rejection
REBEND	Final wavelength for rebinning
REBMTD	Rebinning method (LINEAR, QUADRATIC, SPLINE)
REBOPT	Rebin Option (YES/NO)
REBSTP	Wavelength step for rebinning
REBSTRT	Starting wavelength for rebinning
RESPLOT	Plot flag for response computation
RESPONSE	Response Image
RESPOPT	Response Correction Option (YES/NO)
RESPTAB	Intermediate Response Table
RON	Read-Out-Noise (ADU)
ROTOPT	Rotation Option (YES/NO)
ROTSTART	Y-start after rotation
ROTSTEP	Y-step after rotation
SEAMTD	Search centering method (GAUSS, GRAV, MAXI)
SESSION	Session name
SHIFT	Shift in pixels
SIGMA	Threshold for rejection of cosmics (std dev.)
SKYMOD	Mode of fitting
SKYORD	Orderfor sky fit
SMOOTH	Smoothing factor for spline fitting
START	start points of the raw image.
STD	Standard Star Spectrum
STEP	start points of the raw image.
THRES	Threshold for line detection (above local median)
TOL	tolerance in Angstroms for wavelength ident.
TRIM	Trim window (x1,y1,x2,y2) in pixels
TRIMOPT	Trim Option (YES/NO)
TWODOPT	Computes bivariate polynomial option.
UPPSKY	Lower, upper row number of upper sky
WCENTER	Central wavelength
WIDTH	Window size in X for line detection (pixels)
WLC	wavelength calibration image
WLCMTD	Wavelength calibration method (IDENT,GUESS)
WLCNITER	Minimum, Maximum number of iterations
WRANG	wavelength range to take from LINCAT
YSTART	Starting row for the calibration (pixel value)
YSTEP	Step in Y for line searching (pixels)
YWIDTH	Window size in Y for line detection (pixels)

Table 6.8: Keywords Used in Context `LONG` (continued)

6.10 Example

As an example of the use of the commands described above, we here include the tutorial procedure, executed as `TUTORIAL/LONG`.

The input images are `wlc`, the wavelength calibration frame, and `obj`, the object. The catalogue of laboratory wavelengths used is stored in the table `lincat`.

```
INIT/LONG
GRAPH/LONG
MAKE/DISPLAY
!
WRITE/OUT Copy test images
-DELETE lndemo_*.*
-COPY MID_TEST:emhear.bdf   lndemo_wlch.bdf
-COPY MID_TEST:emth.bdf     lndemo_wlcth.bdf
-COPY MID_TEST:emstd.bdf    lndemo_wstd.bdf
-COPY MID_TEST:emmi0042.bdf lndemo_bias1.bdf
-COPY MID_TEST:emmi0043.bdf lndemo_bias2.bdf
-COPY MID_TEST:emmi0044.bdf lndemo_bias3.bdf
-COPY MID_TEST:emmi0045.bdf lndemo_bias4.bdf
-COPY MID_TEST:emmi0046.bdf lndemo_flat1.bdf
-COPY MID_TEST:emmi0047.bdf lndemo_flat2.bdf
-COPY MID_TEST:emmi0048.bdf lndemo_flat3.bdf
-COPY MID_TEST:emmi0049.bdf lndemo_flat4.bdf
-COPY MID_TEST:thorium.tbl  lndemo_thorium.tbl
-COPY MID_TEST:hear.tbl     lndemo_hear.tbl
-COPY MID_TEST:l745.tbl     lndemo_l745.tbl
-COPY MID_TEST:atmoexan.tbl lndemo_atmo.tbl
!
WRITE/OUT "This tutorial shows how to calibrate long slit spectra"
WRITE/OUT "The package assumes wavelengths increasing from"
WRITE/OUT "left to rigth."
WRITE/OUT "It is assumed that the images have been already"
WRITE/OUT "rotated, corrected for pixel to pixel variation"
WRITE/OUT "and the dark current has been subtracted."
WRITE/OUT "Input data are:"
WRITE/OUT "wlc.bdf       - wavelength calibration image"
WRITE/OUT "obj.bdf       - object image"
WRITE/OUT "lincat.tbl    - line catalogue"
!
WRITE/OUT  "Combining flat and dark images"
!
LOAD   lndemo_flat1
CREATE/ICAT bias lndemo_bias*.bdf
COMBINE/LONG   bias  lnbias  MEDIAN
STAT/IMA       lnbias
!
CREATE/ICAT flat lndemo_flat*.bdf
SET/LONG TRIM=20,60,520,457
PREPARE/LONG   flat.cat lndemo_ft
```

```
CREATE/ICAT flat lndemo_ft*.bdf
COMBINE/LONG   flat  lnff    AVERAGE
NORMALIZE/FLAT lnff  lnflat  190.
!
CREATE/ICAT  lndemocat  lndemo_w*.bdf
READ/ICAT    lndemocat
LOAD         lndemo_wlch
WRITE/OUT "Extracting useful part of spectra with command PREPARE/LONG"
SET/LONG  TRIM = 0,60,0,457
PREPARE/LONG  lndemocat.cat lndemo
!
WLC:

SET/GRAPH PMODE=1 XAXIS=AUTO YAXIS=AUTO
SET/LONG WLC=lndemo1 LINCAT=lndemo_hear YWIDTH=10 THRES=30.
SET/LONG YSTEP=10 WIDTH=8 TWODOPT=YES DCX=2,1
!
SESSDISP = "NO "
SHOW/LONG wlc
!
WRITE/OUT Search lines:
WRITE/DESCR {WLC}  STEP/D/2/1  -2.
SEARCH/LONG                 ! search calibration lines
PLOT/SEARCH
!
WRITE/OUT "Identify some of the brightest lines:"
WRITE/OUT
WRITE/OUT " X   = 379.30     922.50 "
WRITE/OUT " WAV = 5015.680   5606.733"
WAIT 2
IDENTIFY/LONG               ! interactive line identification
SET/LONG  WLCMTD=IDENT  TOL=0.3
CALIBRATE/TWICE             ! wavelength calibration
PLOT/IDENT                  ! display initial identifications
!
WRITE/OUT Compute the dispersion coefficients by fitting a 2-D polynomial
WRITE/OUT to the whole array
PLOT/CALIBRATE              ! display all identifications
PLOT/RESIDUAL
PLOT/DISTORTION 5015.680
!
SAVE/LONG ses1
WRITE/OUT "Now calibrating another arc spectrum in GUESS mode"
SET/LONG  WLCMTD=GUESS GUESS=ses1 WLC=lndemo2 LINCAT=lndemo_thorium
SET/LONG  WIDTH=4  THRES=3.  TOL=0.1   ALPHA=0.2
LOAD {wlc}
SEARCH/LONG
CALIBRATE/LONG
!
```

```
WRITE/OUT "Now demonstrating the three possible ways to apply the"
WRITE/OUT "dispersion relation : "
WRITE/OUT " - APPLY/DISPERSION involves no rebinning and outputs a table."
WRITE/OUT "     Input must be a 1D spectrum or a row of a long-slit spectrum"
WRITE/OUT " - REBIN/LONG       rebins row by row, taking coefficients from coerbr.tbl"
WRITE/OUT " - RECTIFY/LONG     applies the 2D polynomial dispersion relation"
WRITE/OUT "Note: Rebin can be applied before or after extraction"
!
!INIT/LONG ses1
!
APPLY/DISPERSION {wlc}  wlct  @100
PLOT/SPECTRUM    wlct
!
REBIN/LONG {wlc} wlcrb
LOAD       wlcrb
PLOT       wlcrb  @100
!
RECTIFY/LONG {wlc}  wlc2
LOAD         wlc2
PLOT         wlc2  @100
!
WRITE/OUT "Session is now saved, initialized, and loaded from session tables"
SAVE/LONG mysess
INIT/LONG
SESSDISP = "NO "
SHOW/LONG
INIT/LONG mysess
SESSDISP = "NO "
SHOW/LONG
!
WRITE/OUT "Now extracting a spectrum with two possible methods:"
WRITE/OUT "  - Simple rows average with EXTRACT/AVERAGE"
WRITE/OUT "  - Optimal extraction  with EXTRACT/LONG"
LOAD/IMA         lndemo3
SET/LONG         REBSTR=4600. REBEND=5800. REBSTP=2.00
REBIN/LONG       lndemo3   ext8
SET/LONG         LOWSKY = 189,198  UPPSKY = 204,215
SET/LONG         GAIN=2. RON=5. THRES=3. RADIUS=2
SKYFIT/LONG      ext8    stdsky
LOAD             stdsky
COMPUTE/IMAGE    ext7 = ext8 - stdsky
SET/LONG         OBJECT = 199,203
EXTRACT/AVERAGE  ext7   stda
PLOT             stda
EXTRACT/LONG     ext8   stde  stdsky
PLOT             stde
!
WRITE/OUT  "Now computing instrumental response"
!
```

```
SET/LONG FLUXTAB=lndemo_l745  EXTAB=lndemo_atmo
PLOT/FLUX
EXTINCTION/LONG  stde stdext
RESPONSE/FILTER  stdext
INTEGRATE/LONG   stdext
RESPONSE/LONG fit=SPLINE
PLOT/RESPONSE
CALIBRATE/FLUX   stdext stdcor
CUTS             stdcor   100.,500.
PLOT             stdcor
```

Chapter 7

Echelle Spectra

This Chapter provides the basic information necessary to understand the echelle package implemented in MIDAS. It includes the description of the reduction method and the system implementation, without reference to a particular instrument. A detailed description of the operation and relevant parameters for the different instruments supported are described in the corresponding Appendix D. Tutorial examples are available in the system (command TUTORIAL/ECHELLE) and are also included in the Appendix D so that users without previous experience in MIDAS could become familiar with the calibration procedures.

The package follows a very flexible scheme, where most of the reduction steps include several algorithms which can be dynamically chosen and some of the steps can be executed optionally. Experienced users could modify the scheme to adapt it to their data configuration. The methods described here are generally sufficient to cover a wide range of echelle formats. The ESO instruments supported by the package are CASPEC (Cassegrain Echelle Spectrograph), EFOSC(1+2), ECHELLEC, and EMMI.

The package provides a set of about 30 first-level basic commands to perform the reduction, most of them having the qualifier ECHELLE. These commands are structured in five main MIDAS procedures to perform complete steps of reduction. In Section 7.1 we describe the algorithms used in the echelle reduction, i.e. the algorithms to find the position of the echelle orders, order extraction procedures, wavelength calibration and instrument response correction. In Section 7.10 we include a brief outline of the different data formats involved in the reduction and a summary of the commands. Finally, the session parameters are detailed in Section 7.11

Note

This chapter only provides a synopsis of the commands needed for the reduction of echelle. It is important to realise that it can neither substitute the HELP *information available for each command nor be exhaustive, especially not with regard to the usage of general utilities such as the* MIDAS *Table System, the* AGL *Graphics package, etc. Reduction steps which are not specific to echelle spectra are described in more detail in Chapter L. The Appendix D provides a practical approach to echelle reduction.*

7.1 Echelle Reduction Method

7.1.1 Input Data and Preprocessing

The information involved in a reduction session consists of user data and system tables. User data is a set of echelle images, observed with the same instrument configuration, including a wavelength calibration image (WLC), a flat field image (FLAT) and astronomical objects OBJ. Optionally, this set will include standard stars (STD) to be used in the absolute or relative flux calibration, and dark images (DARK). Catalogues with comparison lines and absolute fluxes for standard stars are available in the system as MIDAS tables.

Before starting the actual reduction some preprocessing of the data is required to correct for standard detector effects as follows:

- Rotation of input frames.
  After this rotation, the dispersion direction of the echelle orders will be horizontal, with wavelengths increasing from left to right and spectral order numbers decreasing from bottom to top of the image. As always in MIDAS, the origin is the pixel $(1, 1)$, located in the lower left corner of the image.
- Updating START and STEP descriptors.
  Descriptors START and STEP must be set to 1.,1. for all images processed. Session keyword CCDBIN must be set to the original binning factor along x- and y-axis. Image rotation and descriptors update are performed by the command ROTATE/ECHELLE.
- Cleaning of bad columns.
  First, bad columns – bad rows after the rotation – can be removed with the command COMPUTE/ROW. The cleaning of bad columns is required for FLAT images where the variation of the intensity due to these columns can affect the automatic detection of the orders.
- Cleaning of hot pixels.
  Hot pixels can be eliminated by filtering the images. In case the observation has been splitted in several exposures and more than one image is available with the same information, the images can be averaged with the command AVERAGE/WINDOW; this command can, optionally, interpolate pixel values with large deviations from the average value. Removal of hot pixels is required for DARK images and is recommended for OBJ exposures. General methods to clean bidimensional spectra are described in Chapter L (Removing Cosmic Ray Hits). The command FILTER/ECHELLE, adapted to echelle spectra is described in Section 7.3
- Subtraction of dark current from FLAT, OBJ and STD frames. The dark level is estimated from a series of DARK exposures of short duration which are averaged to reduce the effect of the read–out noise of the CCD and to eliminate hot pixels as described before. If preflashing is necessary, a set of preflashed DARK exposures should be obtained in a similar manner. It is advisable to obtain a set of DARK images with similar exposure times as the object and standard star frames, or to scale the dark level to the observed exposure.

- Checking exposure times in OBJ and STD frames. For images generated by ESO instruments, the exposure time (in seconds) is stored in the descriptor O_TIME(7). If necessary this descriptor can be created as O_TIME/D/1/7 with the command WRITE/DESCRIPTOR.

7.1.2 Retrieving demonstration and calibration data

Calibration tables are required to provide reference values of wavelength for Th-Ar arc lamp lines, atmospheric extinction or standard star fluxes.

A certain number of tables are distributed on request in complement to the Midas releases. These tables are also available on anonymous ftp at the host **ftphost.hq.eso.org** (IP number 134.171.40.2). The files to be retrieved are located in the directory /midaspub/calib and are named README.calib and calib.tar.Z. Command SHOW/TABLE can be used to visualize the column name and physical units of the tables. Demonstration data required to execute the tutorial procedure TUTORIAL/ECHELLE are also located on this ftp server in the directory /midaspub/demo as echelle.tar.Z. FTP access is also provided on the World Wide Web URL:

http://http.hq.eso.org/midas-info/midas.html

The calibration directory contains other information such as characteristic curves for ESO filters and CCD detectors, which can be visualized with the Graphical User Interface XFilter (command CREATE/GUI FILTER).

7.1.3 General Description

The first problem in the reduction of echelle spectra is, of course, the solution of the dispersion relation. That is the mapping between the space (λ, m) wavelength, spectral order and the space (x, y) sample x, line y in the raw image. This relation gives the position of the orders on the raw image, and defines the wavelength scale of the extracted spectrum. The mapping is performed in two steps:

- A first operation (order definition), gives the position of the orders in the raw image. In figure 7.1, this operation corresponds to the step “Find Order Position”. The required input is an order reference frame (usually FLAT or STD) and the output is a set of polynomial coefficients. These coefficients are an input of the step “Extract Orders”.

- A second operation (wavelength calibration) defines the wavelength scale of the extracted spectrum. The successive steps of this operation are shown in the second column of figure 7.1. The output is a set of dispersion coefficients required by the step “Sample in Wavelength”.

Sections 7.2 and 7.6 describe the solution of this mapping.

The second step in the reduction, described in Section 7.4, is to estimate the image background. The background depends mainly on the characteristics of the detector, but includes the additional components of the scattered light in the optics and spectrograph. This operation corresponds to the step “Subtract Background” in fig. 7.1.

FLAT	WLC	STD	OBJECT
⇓	⇓	⇓	⇓
FIND ORDER POSITIONS	EXTRACT ORDERS	*SUBTRACT BACKGROUND*	SUBTRACT BACKGROUND
⇓	⇓	⇓	⇓
SUBTRACT BACKGROUND	IDENTIFY LINES	*FLAT FIELD CORRECTION*	*FLAT FIELD CORRECTION*
⇓	⇓	⇓	⇓
DEFINE BLAZE	COMPUTE DISP.COEFFS.	*EXTRACT ORDERS*	EXTRACT ORDERS
		⇓	⇓
		SAMPLE IN WAVELENGTHS	SAMPLE IN WAVELENGTHS
		⇓	⇓
		COMPUTE RESPONSE	*MULTIPLY BY RESPONSE* or *FIT BLAZE*
			⇓
			MERGE ORDERS

Figure 7.1: Echelle Reduction Scheme

A particular problem in the CCD–detector used by the two echelle instruments is the appearence of interference fringes produced within the silicon, which can be especially important in the long wavelength range of the instrument. By processing the flat-field (first column of fig. 7.1), correction frames are prepared and used for the standard star and the object reduction. A method to correct for this effect is described in Section 7.7.

After the corrections for all these effects, the information in the spectral orders is extracted using the methods described in Section 7.5. The extracted flux, used in conjunction with the dispersion relation, gives the photometric profiles of the spectral orders. Two instrumental effects are still present in these profiles: first, due to the blaze effect of the echelle grating, the efficiency of the spectrograph changes along each order; second, the efficiency of the whole instrument is not uniform with wavelength. In Section 7.8 we describe how to correct both effects, to normalize the fluxes and, if the input data includes calibration stars, to convert the fluxes into absolute units.

Note

Taking a standard star exposure (STD) is a recommended observation strategy which can make easier the order definition in the blue part of the spectrum as well as the correction of individual orders for the variations of grating efficiency (blaze function).

The steps summarised above comprise the `STANDARD` reduction. Alternatively, it is possible to correct the variation in sensitivity along the spectral orders using a suitable model for the blaze function as described in Section 7.8.2. Figure 7.1 displays the process scheme in a typical reduction session; slanted fonts indicate optional operations. In the rest of this Section the algorithms used in each step of the reduction are described.

7.2 Order Definition

The dispersion relation is defined by the following equations:

$$\begin{aligned} y &= f_1(x, m) \\ \lambda &= f_2(x, m) \end{aligned} \tag{7.1}$$

The first of the equations 7.1 defines the position of the spectral orders, m, in the raw image, while the second equation gives, for each order, the dispersion relation in one dimension. The mapping between the spaces (λ, m) and (x, y) is separated into two different equations; the first one will be discussed in this Section, while the description of the second equation will be postponed to Section 7.6.

The function f_1 is approximated by a polynomial of the form

$$y = f_1(x, m) \approx \sum_{j=0}^{J} \sum_{i=0}^{I} a_{ij} x^i m^j \tag{7.2}$$

where the coefficients a_{ij} are computed using least squares techniques on a grid (x_k, y_k), i.e. sample number and line number of points located within the spectral orders of the image. These points in the grid are found automatically by an order–following algorithm, normally using the FLAT or STD image.

- A first guess of the position of the orders is found on a trace perpendicular to the dispersion direction done in the middle of the flat field image, in this way we define the set of points (x_0, y_{0m}), m being the relative order number.

- For each order, the order–following algorithm finds the series of points located on the order at $x_n = x_0 + n \times \Delta x$ for points on the right half of the order, and at $x_{-n} = x_0 - n \times \Delta x$ for points on the left half of the order, $n = 1, 2, \ldots$ integer and Δx is the step of the grid.

This set of points forms the basic grid with the geometric positions of the orders. Typical values of the standard deviation of the residuals of this approximation are about 0.3 to 0.1 pixel.

It is worth mentioning here that the order following algorithm finds the center of the orders by taking the middle point with respect to the edges of the orders. The edges of the orders are detected automatically by thresholding the order profiles, perpendicular to the dispersion direction; the level of the threshold is a function of the signal in the order. The command DEFINE/ECHELLE performs the automatic order detection.

An alternative method is available, based on the Hough transform to perform the order detection and involving a tracing algorithm able to estimate an optimal threshold for each order independently. The order definition is performed as follows:

- A preprocessing of the frame is performed, including a median filtering (radx,y=2,1) to remove hot pixels and bad rows from the image. Then the background value is measured in the central area of the image and subtracted. This preprocessing assumes that the defaults are small enough to be corrected by a simple median filtering and that the interorder background is basically constant all over the image. If the above conditions are not respected, the frame must be processed by the user. The echelle command BACKGROUND/SMOOTH enables performance of a background correction at this early stage of the calibration.

- A first guess of the position and the slope of the orders is found by processing the Hough transform of a subset of columns of the input image. The order detection by Hough transform is described in (Ballester, 1994).

- For each order, an initial threshold is estimated by measuring the pixel values in the middle of the order. The order following algorithm finds the series of points located on the order at regular steps on the grid, as describd above. The threshold is optimised in order to follow the order on the longest possible distance. If the trace of the order is lost, the algorithm extrapolates linearly the positions and attempts to skip the gap.

- For each position, the center of the order is defined as the first moment of the pixel values above the threshold:

$$y_{center} = \frac{\sum_{y=y_{min}}^{y_{max}} y * (I_y - threshold)}{\sum_{y=y_{min}}^{y_{max}} y}$$

  This algorithm is implemented in the command `DEFINE/HOUGH`. The algorithm can run in a fully automatic mode (no parameters are required apart from the name of the input frame). It is also possible to set the following parameters to enforce a given solution:

  - numbers of orders to be detected.
  - half-width of orders
  - threshold

A practical decription of the way to use this algorithm and to optimise the parameters is described in the Appendix D

7.3 Removal of particle hits

Cosmic ray events pose a serious problem in long CCD exposures. Their removal is a rather delicate step, because in high-contrast images (such as well-exposed echelle spectra) there is always a danger of damaging the scientific contents of the frame. Particle hits can be removed from scientific exposures by splitting the exposure and comparing spectra of the same target obtained under the same instrumental configuration. Offsets of the target resulting from the positioning of the target on the entrance slit of the spectrograph and variations of exposure time must be accounted for. Commands `AVERAGE/IMAGE` and `AVERAGE/WEIGHT` offer number of options to compare the images and reject particle hits. In the case of echelle spectra of sources with very little variation of the spectral information along the slit, one can also exploit the knowledge provided by the order definition as to where in the frame the relevant data is located.

The removal of unwanted spikes, above an otherwise featureless background such as the inter-order space of echelle spectra, is done most easily with a median filter. Therefore, in a first step a median filter is applied to the entire frame. This then enables the true background to be determined as described in Section 'Background Definition' of this chapter. The subtraction of the background calculated completes the second step, and, as far as the inter-order space is concerned, the final result is reached already. The removal of cosmics from the object spectrum forms the third step and is restricted to the regions covered by the spectral orders in the background-corrected, but otherwise raw frame. This step comprises the following operations performed within a sliding (along and separately for each spectral order) window of user-specified width:

- For all x_i of the window, normalize the spatial profile to the total flux in the associated slice (i.e. all y_j of the order considered).

- Form the 'true' spatial profile as the median over the individual profiles at all x_i in the window.

- For all pixels (x_c, y_j) in the central slice, c, of the window compare their re-normalised flux, $f_{c,j}$, with the properly scaled contents m_j of pixel j in the median profile. If the difference exceeds the expected statistical error of $f_{i,j}$ (calculated from the number of photons detected and the readout noise) by a user-specified factor, replace $f_{c,j}$ with m_j.

If the threshold for substitution by the median is set properly ($\sim 4\ \sigma$) and the spectral information within the spatial profile does not change (point sources are best), this procedure does not redistribute the flux or dilute the point spread function.

These three steps are the backbone of command `FILTER/ECHELLE`. The final output frame is the merger of the median-filtered inter-order domains with the spectral orders after having been subjected to step three. Note that the background has been subtracted already. A keyword `BACKGROUND` with contents `SUBTRACTED` is appended to the frame as a flag to subsequent high-level procedures so as not to have to go through the very time consuming step of the background modeling again. Delete that descriptor or change its contents if re-modeling of the residual background is desired.

7.4 Background Definition

The estimation of the background is one of the critical points in the reduction of echelle spectra for two reasons. On one side, a correct estimate of the background level is necessary to compute the true flux of the object spectrum; on the other side, a wrong estimate of the background in either the flat field image (`FLAT`), the object (`OBJ`) or (optionally) the standard star (`STD`), will severely affect the accuracy with which instrumental effects – such as the blaze – can be corrected for. The background in an echelle image consists of:

- a constant offset introduced by the electronics (bias),

- an optional constant pedestal due to the pre–flashing of the CCD,

- the dark current,

- general scattered light,

- diffuse light in the interorder space coming from adjacent orders.

- sky background spectrum

The first three components are removed during the preprocessing as described in Appendix D. Correction for the general scattered light and diffuse light background is presented in the three following sections. Correction for the sky background is presented in Section 7.4.4.

The remaining background due to scattered light is estimated from points located in the interorder space. These locations are defined when performing the order definition (command DEFINE/ECHELLE or DEFINE/HOUGH) and can be displayed by command LOAD/ECHELLE. The order definition is used to create a table back.tbl containing the background positions. Unscanned areas of the CCD can be avoided in the background estimate by using the command SCAN/ECHELLE. If bright features (e.g. sky lines) are located at the same location as background points, these locations must be unselected from table back.tbl using SELECT/BACKGROUND. Selection of the method is possible by assignment of a value to the echelle keyword BKGMTD and the background estimate, subtraction and update of the descriptor BACKGROUND is performed by the command SUBTRACT/BACKGROUND

7.4.1 Bivariate polynomial interpolation

Background points are used to approximate the observed background by a polynomial of two variables, sample and line numbers, as:

$$B(x,y) \approx \sum \sum b_{ij} x^i y^j \tag{7.3}$$

The background of flat field images is usually well modelled by a 2D polynomial of degrees 3 and 4 in variables sample and line respectively. The agreement of the model is typically better than 1% of the background level. For object exposures the signal–to–noise ratio is normally much lower, as is the actual background level. A polynomial of lower degree, for example linear in both dimensions or a constant background should be enough. Because small errors in the determination of the background are carried through the whole rest of the reduction and are even amplified at the edges of the orders, care should be taken in the background fitting.

If no DARK or BIAS frames are available, the background definition might be slightly less accurate because the modelling procedure has to take into account these contributions as well. In some cases the degree of the polynomial has to be increased. As a rule of thumb, one should try to fit the background with a polynomial of the lowest possible degree.

This method gives good results when the main contribution to the background is due to global scattered light.

7.4.2 Smoothing spline interpolation

An alternative method performs the interpolation of interorder background using smoothing spline polynomials. Spline interpolation consists of the approximation of a function by means of series of polynomials over adjacent intervals with continuous derivatives at the end-point of the intervals. Smoothing spline interpolation enables to control the variance of the residuals over the data set, as follows:

$$\delta = \sum_{i=1}^{m} (y_i - \hat{y}_i)^2$$

where y_i is the i^{th} observed value and $\hat{y}_i$ the i^{th} interpolated value is the sum of the squared residuals and the smoothing spline algorithm will try to fit a solution such as:

$$\delta \leq S * \alpha$$

where S is the smoothing factor and $\alpha = 0.001$ is the tolerance.

One must retain two particular values of S:

- S = 0. The interpolation pass through every observation value.
- S very large. The interpolation consists of the one-piece polynomial interpolation.

The solution is estimated by an iterative process. Smoothing spline interpolation is designed to smooth data sets which are mildly contaminated with isolated errors. Convergence is not always secured for this class of algorithms, which on the other hand enables to control the residuals. The median of pixel values in a window surrounding the background reference position is computed before spline interpolation. The size of the window (session keyword `BKGRAD`) is defined along the orders and along the columns of the raw spectrum.

7.4.3 Background estimate by filtering

The command `BACKGROUND/SMOOTH` enables to define the background without previous order definition. This feature is useful for example to correct for the background the image required for the order definition. This command involves an iterative algorithm which performs the following operations:

- the input frame is heavily smoothed in the direction perpendicular to the orders.
- the original frame is divided by the smoothed one and all pixels which ratio is greater than one are replaced by the smoothed value.
- the corrected frame becomes the new input frame and the two previous steps are repeated until a satisfying solution is obtained.

A more elaborated scheme is required if the contribution to the background from adjacent orders is important, this occurs when the distance between orders is small. All above methods are implemented in command `BACKGROUND/ECHELLE` and selected by the echelle keyword `BKGMTD`.

7.4.4 Sky background definition

The previous methods allow to correct for the interorder scattered light. The sky background component can be measured in spectra taken with an extended slit using the two commands `DEFINE/SKY` and `EXTRACT/SKY`. The command `DEFINE/SKY` allows to define the offset limits of up to two sky windows, usually on both sides of the spectrum. The command `EXTRACT/ECHELLE` performs an average extraction of the sky background using these offsets, optionally fiters for cosmics, and provides an extracted spectrum in a format similar to the one resulting from `EXTRACT/ECHELLE`. The two extracted images can be subtracted for sky background correction. It can be noted that the background measured by the command `EXTRACT/SKY` includes also the interorder scattered light components and makes in principle unnecessary the correction described in the previous Section.

7.5 Order Extraction

Individual echelle orders are extracted by adding the pixel values over a numerical slit running along the orders, with the position of the slit center defined by equation 7.2. The width of the slit is one pixel and its length, as well as an optional offset to shift the slit prependicular to the dispersion direction, are defined by the user.

The pixel values in the numerical slit are found by linear interpolation of the values in the image. The extracted flux number is the weighted average of these interpolated pixel values. There are three types of weights (w) depending on the selected option:

- option `LINEAR`: $w = 1$,
- option `AVERAGE`: $w = 1/L$ where L is the length of the slit,
- option `OPTIMAL`: weights optimising the signal to noise ratio of the extracted spectrum. This algorithm is based on (Mukai, 1990)

There are several effects to consider when defining the length of the extraction slit. If the length is too small, the orders are only partially extracted and they present a periodic variation due to the inclination of the orders with respect to the lines in the image. On the other side, if the slit is too large, the extracted flux will include noisy pixels from the flatfielded background, when the flat field correction is applied.

The command `EXTRACT/ECHELLE` is used for the order extraction. The method is selected using echelle keyword `EXTMTD`.

7.6 Wavelength Calibration

7.6.1 General Description

A preliminary step to the wavelength calibration consists of extracting the orders of the `WLC` image which can then be used to determine the dispersion relation in two steps:

- The calibration lines are detected on the extracted orders by means of a simple thresholding algorithm. The center of the line is estimated by its center of gravity (`GRAVITY` method) or by a gaussian fit to the line profile (`GAUSSIAN` method). This is done with the command `SEARCH/ECHELLE`.
- A few lines are identified interactively on the 2D image display and a set of global dispersion coefficients are derived by comparing the identified lines with the line catalogue available in the system. This global model for the dispersion is a function of the wavelength and the spectral order number. Finally, dispersion coefficients

for each order are computed using the global coefficients as a first approximation. A polynomial of degree 2 or 3 is sufficient to obtain, for each order, a good approximation of the wavelength scale.

The command `IDENTIFY/ECHELLE` involves the echelle relation and requires the identification of two lines in overlapped regions of adjacent orders (method `PAIR`). The calibration can as well be performed for spectra which orders are not overlapped, this time requiring a minimum of four identifications (method `ANGLE`). Both methods are based on the echelle relation and therefore are not applicable if the disperser is not an echelle grating, as it is the case for `EFOSC` which involves a grism disperser. The method `TWO-D` allows to start directly the calibration with a two-dimensional fitting polynomial and requires more initial identifications. In case of several observations with the same, or near the same instrumental configuration, it is possible to use the global dispersion model from a previous calibration. The method `GUESS` implements this mode of operation. Two additional methods `RESTART` and `ORDER` are available. The selection of the method is performed by assigning a value to the echelle keyword `WLCMTD`.

Solutions are computed either for each independent order (`WLCOPT=1D`) or using a global bivariate polynomial (`WLCOPT=2D`).

7.6.2 The Echelle Relation

The wavelength calibration involves a physical equation, the echelle relation and regression analysis to achieve estimates of the dispersion relation. Provided that the echelle dispersion is performed with a grating, any echelle spectrum can be calibrated usually with four lines used as pre-identifications and a catalog of laboratory wavelengths associated to the calibration lamp. The achieved accuracy is usually in the range 0.2 - 0.02 pixel. Accuracy can be improved by selecting lines of a sufficient signal-to-noise ratio and using a line catalog sorted for blends for the specific spectral resolution of the instrument.

The echelle relation derives from the grating dispersion relation :

$$\sin i + \sin \theta = k.m.\lambda$$

with k the grating constant, m the order number, and λ the wavelength. The cross-disperser displaces successive orders vertically with respect to one another. For a given position x on the frame, we have :

$$m.\lambda = cste(x) \text{ (Echelle Relation)}$$

The acurracy of this relation is limited by optical aberrations and optical misalignments, which make it only useful to initialise the calibration process by reducing the number of identifications necessary to determine this one-dimensional relation, expressed as a polynomial of low degree N like:

$$\lambda(m, x) = \frac{1}{m}.\sum_{i=0}^{N} a_{mi}.x^i$$

The two major limits of accuracy of the echelle relation are:

- **Optical aberrations:** the echelle relation does not include the effect of optical aberrations which will displace the lines in the frame and then become a source of unaccuracy when attempting to estimate the echelle relation parameters from a calibration frame. This contributor however will be partially removed by using an appropriate model to fit the echelle relation, like a polynomial of sufficient degree.
- **Optical misalignments:** Optical misalignments occur between echelle grating and cross-disperser between this latter and detector. The effective misalignment angle can be up to a few degrees (usually less than 3). Over many hundreds pixels, the misalignment error amounts to systematic errors of many pixels, far beyond the seeked accuracy. it is therefore necessary to correct for any rotation of the detector.

7.6.3 Estimating the angle of rotation

As a consequence of the cross-disperser dispersion relation, a given line repeated in two overlapped regions of the spectrum must be found at the same y position if spectrum and detector are perfectly aligned.

In the following, we will call a pair two occurences of the same line in the overlapped parts of two adjacent orders.

The above condition provides a geometrical way to estimate the angle, as: $\alpha = \Delta y/\Delta x$, with Δy the difference in y position of the two occurences of the pairs and Δx the difference of their x positions. This method requires overlaps between orders and is used in the calibration method PAIR.

Another approach consists of estimating the rotation angle as the parameter for which regression analysis provides the smallest residual for a given set of observations.

- $x_r = x \cos \alpha + y \sin \alpha$
- $m.\lambda = \sum_{i=0}^{N} a_i.x_r^i$
- α such as the standard-error is minimized

The determination of the optimum is performed analytically and is involved in the calibration method ANGLE. This one-dimensional representation of the dispersion relation is used as a starting point in the loop and is replaced by a bivariate solution $m.\lambda = f(x, m)$ as soon as a sufficient number of identifications has been performed.

7.6.4 Identification loop

The identification criterion combines estimate of the wavelength of a line and an estimate of the error as well as a list laboratory wavelengths to determine and guarantee the identification of a given line. One could involve the additional knowledge of line strengths. But in practice this information of limited use, in reason of variations of relative line intensities caused by impurities, lamp ageing, pressure variations, a.s.o. Accordingly, the identification criterion is:

$$\lambda_c \equiv \lambda_{cat} \text{ if } \exists! \ \lambda_{cat} \ / \ \| \lambda_c - \lambda_{cat} \| < \delta\lambda$$

with λ_c, computed estimated wavelength, λ_{cat} the catalog wavelength and $\delta\lambda$, a majorant of the error of the computed wavelength, taken as the distance of the closest neighbor in the line catalog or in the arc spectrum divided by a coefficient α which value does not exceed 1 (this parameter is controlled by the session keyword `WLCLOOP(2)`).

7.6.5 Resampling and checking the results

The extracted orders in the pixel domain can be resampled into wavelengths with the command `REBIN/ECHELLE`. Several quality checks are provided by the command `IDENT/ECH`. Residuals are visualized with the command `PLOT/RESIDUAL`. A method to verify the wavelength calibration consists of extracting and resampling the wavelength calibration frame and displaying the resampled image with a large scaling factor on the Y-axis. The variation of wavelength coverage from one order to the next should be smooth. Different regions of the resampled image and in particular the overlaps can be verified with the command `PLOT/SPECTRUM`.

7.7 Flat Field Correction

The main disadvantage of the thin CCD–chips used currently as detectors in some instruments is the generation of interference fringes, that is, intensity fluctuations in the spectra which can be as high as 30% at $\lambda > 6000$Å (York, 1981). These fringes arise from interferences within the silicon for long wavelengths, while for shorter wavelengths they can be due to the interfaces silicon–glass on the back side of the chip. This effect is constant for a given setting and it can be effectively corrected by dividing the object image by a flat–field exposure taken with the same instrument configuration. Before the actual division is carried out, the background levels, both in the object image and flat–field, are subtracted and the flat–field is normalised.

The flat–field correction is done with the command `COMPUTE/IMAGE`, this command divides the background subtracted `OBJ` by the normalised flat–field as computed by the command `FLAT/ECHELLE`.

7.8 Instrument Response Correction

7.8.1 Using a Standard Star

The extracted orders, together with the dispersion relation, define the observed flux as a function of the wavelength for each order:

$$F = F_m(\lambda) \tag{7.4}$$

This flux has to be corrected for two effects in order to get absolute fluxes: first, for the echelle blaze effect, and second, for the chromatic response of the instrument. For a given configuration, the blaze effect is a function of the position in the order, while the instrument response is, essentially, a function of the wavelength.

The solution adopted in the reduction, using the standard star, is to correct for both the blaze effect and the instrument response simultaneously. This is done by comparing a standard star, observed with the same configuration as the object, to a table of absolute fluxes. The standard star is reduced exactly as the object and then correction factors are calculated by comparing the flux values in the table to the observed counts sampled at the same wavelength intervals as the fluxes in the table. The resulting response is normalised to an exposure time of one second. There is no effect due to differences between the flatfield of the object and the one corresponding to the standard star given that the flatfields are normalized (see Section 7.7).

If, as usually is the case, OBJ and STD were observed through different airmasses, the spectra have to be corrected for extinction using command `EXTINCTION/SPECTRUM`. More information about this command is available in Vol.2, Chapter L (Spectral Data). The command `RESPONSE/ECHELLE` is used to compute the instrument response as described here. Southern spectrophotometric standards have been published by Hamuy and al. (1992, 1994).

7.8.2 Fitting the Blaze Function

A different approach is also available at this stage, as an alternative to the method described in the last Section. It consists of a correction for the blaze function by using a suitable model of the blaze effect introduced by the echelle grating. In this approach, no correction for the chromatic response of the instrument is applied. It is noteworthy however to say that the standard star correction is a much more efficient way to perform the instrumental response correction. The model assumes a plane grating, used in near–Littrow mode. The blaze function R at wavelength λ is approximated by

$$R(\lambda) = \frac{\sin^2 \pi\alpha X}{(\pi\alpha X)^2} \tag{7.5}$$

where α is a grating 'constant' with value between 0.5 and 1, and $X = m(1 - \lambda_c(m)/\lambda)$, in which m is the order number, and $\lambda_c(m)$ is the central wavelength of order m. Both parameters are related through the grating 'constant' k by $k = m\lambda_c(m)$. This correction is done with the command `RIPPLE/ECHELLE`; the command includes three methods to compute the parameters k and α:

The first one, method `SINC`, is a modification of the method suggested by Ahmad, 1981, *NASA IUE Newsletter*, **14**, 129. This algorithm approximates the blaze function by a sinc square and finds the function parameters by a *non–linear least squares fit to the order profile*. The method is suitable for objects without strong emission or absorption features and can be used to get a first estimation of the blaze parameters.

The second method, named `OVER`, is based on Barker, 1984, *Astron.J.*, **89**, 899. This method uses the overlapping region of adjacent orders to estimate, in a few iterations, the parameter k of the blaze function which is, as before, assumed to be a sinc square. The method works well, provided that orders are overlapping and that there is a very good estimation of the parameter α, assumed to be a contant.

The third method, FIT, is an extension of the previous one. It uses, as before, the overlapping region of the adjacent orders but has the advantage of assuming that both parameters k and α can vary. The method minimises the difference of the corrected orders in each of the overlapping intervals.

7.9 Order Merging

Finally, the extracted orders, sampled at constant wavelength steps and corrected for the blaze effect, can be merged into a one dimensional spectrum which is suitable for further analysis. The merging algorithm computes a weighted average in the overlapping region of adjacent orders. The normalised weight is a linear ramp changing from 0 to 1 in the overlapping region for each of the adjacent orders. Therefore, in the overlapping region of two consecutive orders, the ramp decreases linearly for the 'blue' order while it increases linearly for the 'red' order. This gives less weight to points near the edges of the orders. It is also possible to extract individual orders and store them in different spectral files for further analysis. The command MERGE/ECHELLE performs the merging of the orders.

7.10 Implementation

In this Section we describe the different data formats related to the echelle reduction. Both images and tabular information are used to store intermediate results of the commands in the reduction procedure. We also include a summary of the echelle commands and a detailed description of the parameters used in a reduction session. In the command description we use upper case letters for fixed parts of the command, while names in lower case are variable parameters. Optional parameters are enclosed in square brackets.

7.10.1 The Session Manager

Input data observed with the same configuration of the instrument and the parameters needed for the reduction define a session. Session parameters are listed by the command SHOW/ECHELLE. It is recommended to use this command to control the actual status before executing any further reduction step. Current parameters are saved with the command SAVE/ECHELLE name, where name is the session name. This command will be used whenever you want to interrupt a session which then can be restarted at any other time. Current parameter values are set to the default value with the command INIT/ECHELLE, or set to the values of a previously saved session with INIT/ECHELLE name.

Relevant session parameters can be defined for each command in the usual way:

```
command/qualifier parameters
```

or can be defined in explicit form as

```
SET/ECHELLE param=value [param=value ...]
```

where `param` is the parameter name and `value` is the assigned value. The assigned values will be maintained until you save them for later reference. Current parameter values are re–assigned either by `INIT/ECHELLE` or by another `SET/ECHELLE` command.

Please note that the session concept is part of the high level command structure (level 1 and above). If level 0 commands are used one has to specify explicitely ALL the parameters required by the level 0 commands, despite the fact that some of them may already have been set via `SET/ECHELLE`.

7.10.2 Image Formats

One and two dimensional images are produced by echelle commands. These images are standard MIDAS frames, sampled in the following spaces:

- pixel–pixel space: corresponds to 2D raw images, with the sampling units `PIXEL` in both dimensions.
- pixel–order space: corresponds to 2D extracted images, produced by the command `EXTRACT/ECHELLE`. Unit in the x–axis is `PIXEL`, unit in the y–axis is `ORDER`, sequential echelle order number.
- wavelength–order space: corresponds to 2D extracted images, with uniform sampling steps in wavelength, produced by the command `REBIN/ECHELLE`. Unit in the x–axis is Ångstrøm , unit in the y–axis is `ORDER`, sequential echelle order number. These images *are not standard*, in the sense that the descriptors `START(1)` and `NPIX(1)` are meaningless. The starting position for each order is defined in the descriptor `WSTART`, and the number of pixels per order is given by `NPTOT`, both descriptors are arrays of `NPIX(2)` elements. This deviation from the standard is due, obviously, to the different spectral range of each echelle order.
- wavelength space: corresponds to 1D images, sampled at constant wavelength steps. These images are produced by the command `MERGE/ECHELLE`. Unit in the x–axis is Ångstrøm .

7.10.3 Table Formats

MIDAS table files are used to store relevant information in tabular form. Four types of tables are considered. The first two, tables order.tbl and line.tbl, are operational tables generated by the user during the session. The other two types of tables are provided by the system to calibrate the data in wavelength and flux.

order.tbl — Contains echelle orders. This table contains the coefficients a_{ij} in relation 7.2 stored as descriptors. The table is created by the command `DEFINE/ECHELLE` or `DEFINE/HOUGH`

back.tbl — Contains information related to the background reference positions. This table is created by the command `PREPARE/BACKGROUND` which is called by `DEFINE/ECHELLE` and `DEFINE/HOUGH`.

line.tbl — Contains information related to the position of the calibration lines. This table contains the dispersion coefficients for each spectral order, stored as descriptors after the line identification process. It is generated by the command `SEARCH/ECHELLE` and extended and updated by the command `IDENTIFY/ECHELLE`.

Line Catalogue — Contains the wavelength calibration lines. There are several line catalogues available in instrument related system area, like `MID_CASPEC` and `MID_EFOSC`.

Standard Stars — Contains the fluxes of standard stars in different units. These tables are copied automatically from the area `MID_STANDARD` into the user work space with the command `SET/ECHELLE FLUXTAB=name`

Table order.tbl		
Label	**Unit**	**Description**
ORDER	–	sequential order number
X	PIXEL	sample (x–axis) position of the order
Y	PIXEL	line (y–axis) position of the order
YFIT	PIXEL	fitted line (y–axis) position of the order
RESIDUAL	PIXEL	residual (Y-YFIT)

Table back.tbl		
Label	**Unit**	**Description**
ORDER	–	sequential order number
X	PIXEL	sample (x–axis) position of the order
YBKG	PIXEL	line position of the background above the order
BKG	DN	value of the background

Table line.tbl		
Label	**Unit**	**Description**
X	PIXEL	sample (x–axis) position of the line
Y	PIXEL	relative order number
PEAK	DN	estimated line maximum
YNEW	PIXEL	line (y–axis) position of the line
ORDER	–	spectral order number
IDENT	ANGSTROM	manual identification
WAVEC	ANGSTROM	computed wavelength
RESIDUAL	ANGSTROM	residual (WAVEC-WAVE)

Line Catalogue		
Label	**Unit**	**Description**
WAVE	ANGSTROM	wavelength

Standard Star Atlas		
Label	**Unit**	**Description**
MAGNITUDE	-	magnitude
WAVE	ANGSTROM	wavelength
BIN_W	ANGSTROM	bin width
FLUX_W	ERGS/S/CM/CM/ANG	flux

Table 7.1: Auxiliary Tables

7.10.4 MIDAS Commands

A full description of the echelle commands is given in the Appendix D. We include here a brief summary for quick reference. The echelle package is hierarchically organised from low–level commands (level 0) to high–level commands (level 3). The organization is a result of the following classification:

- Level 0.
  Apart from the level 0 command `LOAD/ECHELLE`, the user will make seldom use of level 0 commands. All parameters of these commands must be provided on the command line. The user may use these commands to benefit from options non available from higher level commands.

- Level 1.
  Level 1 commands perform the individual steps of the reduction. Parameters may be provided on the command line (thus updating the echelle keywords) or will be read from the echelle keywords (set by command `SET/ECHELLE`). Level 1 command enable to check individual steps of the reduction.

- Level 2.
  Level 2 commands perform high level steps of the reduction. The level 2 set of commands is self–consistent and enables to perform a complete reduction, with the optional use of verification lower level commands `PLOT/ECHELLE, PLOT/IDENT, LOAD/IDENT, PLOT/RESIDUAL, LOAD/ECHELLE`. Apart from method keywords and input/output images, no parameter value is required on the command line, all keywords being updated by `INIT/ECHELLE` and `SET/ECHELLE` commands.

- Level 3.
  The higher level 3 includes only two very general commands. The `SET/CONTEXT` initializes echelle keyword values and `TUTORIAL/ECHELLE` provides a demonstration of the package.

SET/CONTEXT echelle
TUTORIAL/ECHELLE

Table 7.2: Echelle Level–3 Commands

Echelle reduction commands	
CALIBRATE/ECHELLE	[defmtd] [wlcmtd]
FLAT/ECHELLE	[flat] [correct] [blazeff]
FILTER/ECHELLE	input output
PREPARE/WINDOW	catalog flatbkg lhcuts
RESPONSE/ECHELLE	[std] [fluxtab] [response]
REPEAT/ECHELLE	[scalx,scaly] [response]
REDUCE/ECHELLE	input output [bkcor]
ROTATE/ECHELLE	catalog root-name [mode]
SELECT/BACKGROUND	[all]
SCAN/ECHELLE	frame [scan-par]
Session manager commands	
HELP/ECHELLE	[param]
INIT/ECHELLE	[name]
SAVE/ECHELLE	name
SET/ECHELLE	par=value
SHOW/ECHELLE	

Table 7.3: Echelle Level–2 Commands

BACKGROUND/SMOOTH	input output [radx,rady] [niter] [visu]
BACKGROUND/ECHELLE	in out [radx,rady,step] [degree] [smooth] [method]
DEFINE/ECHELLE	[ordref] [width1,thres1,slope] [defmtd] [defpol]
DEFINE/HOUGH	[ordref] [nbord] [hwid] [hough_par] [thresh] [degx,degy] [hot_thres,step]
DEFINE/SKY	ima [nsky] [possky] [half-width]
EXTRACT/ECHELLE	input output slit[,offset] [method]
EXTRACT/SKY	in out [mode]
LOAD/IDENTIFICATIONS	
MERGE/ECHELLE	input output [params] [method]
IDENTIFY/ECHELLE	[wlc] [lincat] [dc] [tol] [wlcloop] [wlcmtd] [guess]
PLOT/ECHELLE	frame [ord1,ord2] [printer]
PLOT/IDENT	frame [ord1,ord2] [printer]
PLOT/RESIDUALS	[ord1,ord2]
PLOT/SPECTRUM	in [start,end]
REBIN/ECHELLE	input output [sample]
RIPPLE/ECHELLE	input output [params] [method] [option]
SEARCH/ECHELLE	input [width2,thres2]
SUBTRACT/BACKGR	input backgr output [bkgmtd] [bkgvisu]

Table 7.4: Echelle Level–1 Commands

AVERAGE/TABLE	frame table columns outcol rad (No on--line help)
EXTRACT/ORDER	in out slit,angle,offset meth table coeffs [ord1,ord2] (No on--line help)
EXTRACT/OPTIMAL	
HOUGH/ECHELLE	input [scan] [step,nbtr] [nbord] [flags] [hwid] [thres]
LOAD/ECHELLE	
PREPARE/BACKGROUND	input [step] [init] [back_tab] [order_tab] [descr]
REGRESSION/ECHELLE	[defpol] [niter] [absres] [kappa]
SEARCH/ORDER	[ordref] [w,t,s] [outtab] [defmtd]
VERIFY/ECHELLE	file [type]

Table 7.5: Echelle Level–0 Commands

7.11 Session Parameters

Parameters: CALIBRATE command (See also next table)	
Parameter	**Description**
INSTR	Defines the spectrograph (e.g. CASPEC, EFOSC).
GRATING	Defines the grating or instrument configuration.
CCD	Defines the CCD.
CCDBIN	Binning factor along X and Y. This parameter has impact on the wavelength calibration.
SCAN	Limits of the scanned area of the CCD. This parameter has impact on order definition (method Hough) and background estimates.
ORDREF	Order reference frame
DEFMTD	Order definition method (STD, COM, HOUGH).
DEFPOL	Degree of the bivariate polynomial used in the definition of the orders, see Section 7.2.
WIDTH1	Defines the width in pixels of the orders perpendicular to the dispersion direction. It is used to detect the orders as described in Section 7.2 (methods STD, COM). Typical values are between 2 and 15, depending on the observing slit.
THRES1	Defines the threshold to detect orders on the order reference image as described in Section 7.2 (methods STD, COM).
SLOPE	Mean slope of the orders in pixel space (methods STD, COM).

Table 7.6: Command parameters (1)

Parameters: CALIBRATE command (Continued)	
Parameter	**Description**
NBORDI	Number of orders to be detected (method HOUGH).
WIDTHI	This parameter is used to remove the traces of the orders in Hough space during the detection. Its minimum value is the half-width in pixels of the orders perpendicular to the dispersion direction. Its maximum value is the interorder distance minus the half-width of the orders (method HOUGH).
THRESI	Defines the threshold for the order detection (equivalent to THRES1)
WLC	Raw wavelength calibration image.
LINCAT	System table with comparison lines.
EXTMTD	Extraction method (AVERAGE, LINEAR, OPTIMAL).
SLIT	Width of the extraction slit in pixels.
OFFSET	Offset of the extraction slit. The offset is given in pixels, positive values are above the center and negative offsets are below the center of the orders as shown in the image monitor (command LOAD/ECHELLE).
SEAMTD	Searching method (GAUSSIAN, GRAVITY). This parameter controls the definition of the center of the calibration lines.
WIDTH2	Defines the width of the calibration lines in pixels.
THRES2	Defines the threshold above the local background to detect comparison lines. The actual value should be chosen to detect about 10 to 20 lines per order.
WLCMTD	Wavelength calibration method (PAIR, ANGLE, TWO-D, GUESS, RESTART, ORDER). This parameter controls the execution of IDENTIFY/ECHELLE.
WLCOPT	Dispersion relation option (1D, 2D)
GUESS	Optional reference to a previous session with the same instrument configuration, to be used as a guess for the dispersion coefficients (required if WLCMTD=GUESS).
WLCVISU	Visualisation of intermediate results (YES/NO).
DC	Is the degree of the polynomial used in the dispersion relation, see Section 7.6.
TOL	Defines a tolerance window for the last step of single-order computation of the dispersion relations.
WLCLOOP	Three parameters to control the iterative identification of lines by improvement of a global solution. • WLCLOOP(1) = Minimum initial accuracy of the solution (pixels) • WLCLOOP(2) = Tolerance on the distance of the nearest neighbours. A value of e.g. 0.2 means that the nearest neighbour must be found at least at 5 times (=1/0.2) the residual to confirm the identification. Optimal values of this parameter are in the range 0. to 0.5. • WLCLOOP(3) = Maximal allowed accuracy of the loop.
WLCNITER	Two parameters for the minimum and maximum number of iterations of the identification loop.

Table 7.7: Command parameters (2)

Parameters: FLAT command	
Parameter	**Description**
FFOPT	Flat-Fielding option (YES/NO). If the value NO is assigned to this paramter, no Flat-Field correction is performed
FLAT	Raw flat field image.
CORRECT	Flat field correction image.
BLAZE	Smoothed flat field used in the normalisation of the flat field.
BKGMTD	Background estimation method (POLY, SPLINE, SMOOTH).
BKGVISU	Visualisation of intermediate results (YES/NO).
BKGSTEP	Step in pixels of the grid of background reference positions.
BKGPOL	Degree of the bivariate polynomial for background fiting (method POLY).
BKGRAD	Radius (radx, rady) in pixels of the window for local estimate of the background. The window is parallel to the orders (radx is measured along the orders, rady is parallel to the columns). (method SPLINE)
BKGDEG	Degree of spline polynomials (method SPLINE).
BKGSMO	Smoothing factor for smoothing spline interpolation. See Section 7.4 (method SPLINE).
EXTMTD	(See definition in Table 7.7)
SLIT	(See definition in Table 7.7)
OFFSET	(See definition in Table 7.7)
SAMPLE	Defines the wavelength step of the spectrum sampled in wavelengths. It can either be a real number indicating the step in units of Ångstrøm or a reference file corresponding to an image sampled in the space order–wavelength.
Parameters: FILTER command	
Parameter	**Description**
BKGxxx	All background parameters : see definitions above.
MEDFILT	widx,widy,no_iter. See help of command FILTER/ECHELLE.
CRFILT	radx,rady,mthresh.
CCDFILT	read-out noise (in e-), inverse gain factor (e-/ADU), threshold (in units of standard deviation).

Table 7.8: Command parameters (3)

Parameters: `RESPONSE` command	
Parameter	**Description**
STD	Raw Standard star, in `STANDARD` reduction mode.
RESPONSE	Computed instrument response in `STANDARD` reduction. If the name `NULL` is assigned to this parameter no response correction is applied in the `REDUCE` command. Instead, the blaze function is fitted.
FLUXTAB	Table with absolute fluxes for the Standard star, in `STANDARD` reduction mode. The tables are in the area `MID_STANDARD`
FILTMED	Radius in x and y, threshold, as in command `FILTER/MEDIAN` used to smooth the response function.
FILTSMO	Radius in x and y, threshold, as in command `FILTER/SMOOTH` used to smooth the response function.
PIXNUL	Number of pixels to set to zero at the edges of each order.
BKGxxx	All background parameters : see definitions in Table 7.8)
FFOPT	Flat-Fielding option (`YES/NO`)
CORRECT	Flat field correction image (required if `FFOPT=YES`).
BLAZE	Smoothed flat field (required if `FFOPT=YES`).
EXTMTD	(See definition in Table 7.7)
SLIT	(See definition in Table 7.7)
OFFSET	(See definition in Table 7.7)
SAMPLE	(See definition in Table 7.7)
Parameters: `REDUCE` command	
Parameter	**Description**
BKGxxx	All background parameters : see definitions in Table 7.8)
FFOPT	Flat-Fielding option (`YES/NO`)
CORRECT	Flat field correction image (required if `FFOPT=YES`).
BLAZE	Smoothed flat field (required if `FFOPT=YES`).
EXTMTD	(See definition in Table 7.7)
SLIT	(See definition in Table 7.7)
OFFSET	(See definition in Table 7.7)
SAMPLE	(See definition in Table 7.7)
RESPOPT	Option for the response corection (`YES/NO`).
RESPMTD	Method for the instrumental response correction (`STD, IUE`).
RESPONSE	Name of the response image as produced by `RESPONSE/ECHELLE` (required if RESPMTD=STD).
RIPMTD	Defines the ripple correction algorithm as `SINC` for the method fitting the blaze function, described by Ahmad(1981), and as `OVER` for the method using the overlapping region of the orders, proposed by Barker (1984) (required if `RESPMTD=IUE`).
RIPK	Defines the initial guess for the parameter k in Equation 7.5. The default value is updated during the wavelength calibration.
ALPHA	Defines the initial guess for the parameter α in Equation 7.5.
MGOPT	Merging option (`YES/NO`).
MRGMTD	Merging method (`AVERAGE, NOAPPEND`).
DELTA	Interval (in wavelength units) to be skipped at the edges of each order.

Table 7.9: Command parameters (4)

Bibliography

[1] Ahmad, 1981, Echelle Ripple Function Determination, NASA IUE Newsletter, 14, 129.

[2] Ballester P., 1992, Reduction of Echelle Spectra With MIDAS, Proceedings of the 4^{th} `ESO/ST-ECF` Data Analysis Workshop, pp. 177-188

[3] Ballester P., 1994, Hough Transform for Robust Regression and Automated Detection, Astron. Astrophys., **286**, 1011-1018.

[4] Barker: 1984, Ripple Correction of High-Dispersion IUE Spectra: Blazing Echelle, Astron.J., 89, 899.

[5] Hamuy M. and al., 1992, Southern Spectrophotometric Standards I, PASP **104** 533-552

[6] Hamuy M. and al., 1994, Southern Spectrophotometric Standards II, PASP **106** 566-589

[7] Mukai K., 1990, Optimal Extraction of Cross-Dispersed Spectra, PASP, **102**:183-189

[8] Pojmanski, 1991, How To Remove Easily the Unpleasant Column Pattern, Proceedings of the 3^{rd} `ESO/ST-ECF` Data Analysis Workshop, pp. 101-105

[9] York D.G. and al., 1981, *Proceedings of the SPIE*, **290**, 202).

Chapter 8

Inter-stellar/galactic Absorption Line Modelling

8.1 Introduction

The program described here is a general code for modelling *interstellar or intergalactic absorption* features on an initial polynomial continuum which may also contain emission lines. For each absorption line the input parameters are : atomic transition, column density, thermal width and position (velocity shift).

The output spectrum is computed at a given instrumental resolution and can therefore be used for a direct comparison with observations (provided that the lines are resolved). In this case the step is completely interactive with no possibility for a "least square approach".

8.1.1 Principle of the Program

1. Creates a 1D image containing the continuum and possible emission lines.
2. Induces absorption features on this image.
3. Possibly, comparison of the resulting image with observations. According to the result, the user can then modify some of the input parameters and repeat the operation until the agreement is found to be satisfactory.

The package contains:

4 commands:	`CREATE/PSF`	:	creation of the instrumental response
	`COMPUTE/EMI`	:	computation of the initial unabsorbed spectrum
	`COMPUTE/ABS`	:	main program, computation of the theoretical absorption spectrum.
	`TUTORIAL/CLOUD`	:	on line tutorial
3 tables:	`EMI`	:	contains the emission line parameters
	`ABSP`	:	contains the atomic data

	ABSC	:	contains the absorption line parameters (cloud model)
4 keywords	CLDDIM	:)	
	CLDZ	:)	monitor several parameters or options
	CLDCT	:)	common to the 3 commands
	CLDOP	:)	

8.2 Astrophysical Context

The process of absorption line formation in the intersteller medium has been described in detail by several authors (e.g. Strömgren 1948, Spitzer 1978). Here we shall give only the basic formulae and the variables to which the user has access in order to make the notations used in the program precise.

8.2.1 Basic Equations

Optical Depth

After having crossed an absorbing cloud, the intensity of a source, $I_o(\lambda)$, is received by the observer as $I(\lambda) = I_o(\lambda)e^{-\tau}$, where τ is the optical depth of the cloud.

Let's connect τ to the physical parameters.

$$\tau = Na(\lambda)$$

N : column density
$a(\lambda)$: line absorption coefficient

$$a(\lambda) = a_o\phi_\lambda$$

ϕ_λ : broadening function
a_o $= \frac{\lambda^4}{8\pi c}\frac{g_k}{g_l}a_{kl}$
l : lower level of the atomic transition
k : upper level of the atomic transition
λ_{lk} : rest wavelength of the transition
g_l : statistical weight of the lower level
g_k : statistical weight of the upper level
a_{kl} : spontaneous transition probability

$$a_{kl} = f_{lk}\ \frac{g_l}{g_k}\ \frac{1}{\lambda_{lk}^2}\ \frac{8\pi^2 e^2}{m_e c}$$

f_{lk} = upward oscillator strength

Broadening Function

Broadening is due both to the natural width of the transition and to the velocity spread of the absorbing atoms along the line of sight.

- In the ideal case of atoms at rest.

$$\phi_\lambda(v=o) = \frac{1}{\pi} \; \frac{\delta_\lambda(\lambda)}{[\delta k(\lambda)]^2 + (\lambda - \lambda_{lk})^2}$$

$$\delta k(\lambda) = \frac{\lambda^2}{4\pi c} \sum_{E_r<E_k} a_{kr}$$

$$= \frac{\lambda^2}{4\pi c} \; SAKL$$

- Let $\psi(v)dv$ be the normalized distribution of atoms with velocity between v and $v + \mathrm{d}v$, then:

$$\phi_\lambda = \frac{1}{\pi} \int_{-\infty}^{+\infty} \frac{\delta_k(\lambda)}{[\delta_k(\lambda)]^2 + [\lambda - \lambda_{lk}(1+\frac{v}{c})]^2} \; \psi(v)dv \qquad (8.1)$$

- In the program the velocity is assumed to be gaussian, thus:

$$\psi(v) = \frac{1}{\sqrt{\pi}} \; \frac{1}{b} \; \exp \left[- \; \left(\frac{v - v_o}{b} \right) \right]^2$$

$$b = \sqrt{\frac{2kT}{m}}$$

v_o = velocity of the cloud relative to the observer.

This full expression (1) is denoted as a "Maxwell + damping wing" or "Voigtian" profile in the program.

In the case of low column density ($\tau < 1$)

τ can be approximated to:

τ = NS ϕ_λ

$$\phi_\lambda = \frac{\lambda_{lk}}{\sqrt{\pi} b} e^{-(w/b)^2} \qquad (8.2)$$

$$\frac{w}{c} = \frac{\nu - \nu_{lk}}{\nu_{lk}}$$

$$S = \frac{\pi e^2}{m_e c} f_{lk}$$

This simplified expression (2) is denoted as a "Maxwellian" profile in the program.
Finally if the line of sight crosses N clouds then, the resulting optical depth is:

$$\tau = \sum_{i=1}^{N} \tau_i$$

In cases where the source has a (cosmological) velocity.
Let z be the redshift of the source.
An absorption is measured in the spectrum at $\lambda = \lambda_a$ corresponding to a rest wavelength λ_o.
This yields for the redshift of the cloud:

$$Za = \frac{\lambda_a}{\lambda_o} - 1$$

The velocity of the cloud relative to the source is:

$$v_{rel} = c \frac{R^2 - 1}{R^2 + 1}$$

$$R = \frac{1 + Z}{1 + Z_a}$$

In practice the program computes the absorption profile in the cloud reference frame ($v_o = 0$) and shifts the result into the observer's rest frame $\left[\lambda \to \frac{\lambda}{1+Za}\right]$

8.2.2 Summary of the parameters handled by the user:

<u>INPUT</u>

- Z source.
- The user is supposed to have determined the continuum and the emission line characteristics.
- Atomic data, for each transition
  - SAKL = $\sum a_{kl}$
  - λ_{kl}
  - f_{kl}
  - g_l
  - g_k
- For each absorption line

  - N
  - b
  - P (λ) : position Ångstroms of the observed absorption feature.

OUTPUT

- Z_a
- v_{rel}

8.3 Typical Run

8.3.1 Preparation

Command:

```
SET/CONTEXT CLOUD
```

8.3.2 Initialisation of the Keywords

These keywords are relative to the CLOUD context and can be handled by the usual commands (READ/KEY.....).

They are required to run the programs:

```
CREATE/PSF  COMPUTE/ABS  COMPUTE/EMI
```

They define the output image dimensions, the coefficients of the polynomial continuum, the redshift of the source, and the type of absorption profile (Maxwellian or Voigtian). Their format is described in section Auxiliary Data.

Ex. command:

```
CLDDIM(1) = 3500   CLDDIM(2) = 0.5   CLDDIM(3) = 200
```

Sets the synthetic image dimensions to:

Start point 3500 Å, step size 0.5 Å, number of pixels 200

```
CLDZ(1) = 2.48
```

Sets the redshift of the source to 2.48

`CLDCT(1) = 1`

Defines a flat continuum

`CLDOP(1) = 1`

Indicates that the absorption coefficients will be computed according to the Voigt Formula.

8.3.3 Creation of the Instrumental Function

The command `CREATE/PSF` creates a 1D image of a normalized gaussian centered in 0. Command:

`CREATE/PSF NAMEP FWMM`

NAMEP	:	output image
FWMM	:	full width at half maximum of the instrumental response (Å)

The stepsize is specified by the keyword CLDDIM.

This image is to be used by the command `COMPUTE/ABS`; any other (non gaussian) `PSF` can be used provided it is centered in 0, normalised and has the same step size as the synthetic image.

8.3.4 Creation of the Input Emission Spectrum

1. The coefficients of the continuum are to be entered in keyword CLDCT.

2. Edition of the table containing the emission line parameters. The lines are assumed to be gaussian. A template can be found in `MID_CLOUD:EMI.TBL`

   Command:

   `EDIT/TAB EMI.`

   The actual format is described in section Auxiliary Data.

3. Computation of the spectrum. The command `COMPUTE/EMI` creates a 1D image, containing emission lines and continuum.

   Command:

   `COMPUTE/EMI SYNEMI EMI`

SYNEMI	:	output spectra.
EMI	:	input table containing the emission line parameter. if no table name is given, the output spectra will contain only the continuum as defined by keyword CLDCT.

8.3.5 Edition of the Table Containing the Atomic Parameters

All of the atomic data listed in 13.2.1. are needed by the program and are to be entered via this table.

A template can be found in `MID_CLOUD: ABSP.TBL`. The quoted parameters are from Morton and Smith (1973) and according to the author the wavelengths are:

$$\text{vaccum for } \lambda < 2000 \text{ Å}, \quad \text{air for } \lambda > 2000 \text{ Å}$$

Command:

```
EDIT/TAB ABSP
```

The actual format of this table is described in section Auxiliary Data.

8.3.6 Edition of the Table Containing the Cloud Model

This table is used both as an input and output by the main program. The table content is:

INPUT

Parameters relative to the absorption lines

For each of them requests are:
column density, b parameter, observed central position (Å), atomic transition.

OUTPUT

In output, the program will write in this table, for each absorption line, the velocity of

the corresponding cloud relative to the source and its redshift (relative to the observer). A template of this table can be found in `MID_CLOUD:ABSC.TBL`

Command:

```
EDIT/TAB   ABSC
```

The actual format is described in section Auxiliary Data.

8.3.7 Computation of the Output Absorption Spectrum

This is the main step achieved by the command:

```
COMPUTE/ABS   SYNEMI   SYNABS   ABSC   ABSP   NAMEP
```

SYNEMI	:	input emission spectrum
SYNABS	:	output spectrum with synthetic absorption lines
ABSC	:	table containing the cloud model
ABSP	:	table containing the atomic data

NAMEP : image containing the instrumental response
Warning : this image must have the same step size as SYNEMI

Then the program:

1. Types:
   PSF: FWHM a ANGSTROM

   STEP: b
   Where a and b are read from the description of the PSF; only as a check for the user.

2. Computes:
   The synthetic spectra according to the parameters of table ABSC.

   The output spectra are:

$$\begin{cases} SYNABS1 & \text{if option Voigtian profiles} \\ or & \\ SYNABS0 & \text{if option Maxwellian profiles} \end{cases}$$

   and

$$\begin{cases} SYNABS1C & \text{convolution of above spectra} \\ or & \\ SYNABS0C & \end{cases}$$

8.4 Auxiliary Data

8.4.1 Description of the Keywords

1. Synthetic image dimensions
   CLDDIM(1) = 'START'
   CLDDIM(2) = 'STEP'
   CLDDIM(3) = 'NPIX'

2. Redshift of the Source
   CLDZ(1) = 'Z'

3. Coefficients for the continuum
   assumed to be a polynom of degree ≤ 5
   CLDCT/R/1/6 :
   continuum = $\sum_{i=1}^{6}$ CLDCT (i) $\bullet$ λ^{i-1}

4. Option for the calculation of the absorption coefficients
   CLDOP(1) = "value"
   the value "0" will give Maxwellian absorption line profiles
   the value "1" will give Voigtian absorption line profiles

	PROGRAMS		
KEYWORDS	COMPUTE/PSF	COMPUTE/EMI	COMPUTE/ABS
CLDDIM	Yes	Yes	
CLDZ			Yes
CLDCT		Yes	
CLDOP			Yes

Table 8.1: Use of the Keywords

8.4.2 Format of the table for atomic parameters (ABSP.TBL).

Column		Name	
1		: TRANSITION	Commentary – name of the transition
2		: MUL	Multiplicity of the line (2 max)
3	[4]	: WL(1), [:WL(2)]	Wavelength(s) of the transition (Å)
5	[6]	: GL(1), [:GL(2)]	a_l value(s)
7	[8]	: GK(1), [:GK(2)]	a_k value(s)
9	[10]	: SAKL(1), [:SAKL(2)]	$\sum a_{kv}$ = SAKL
11	[12]	: FLK(1), [:FLK(2)]	F_{lk} value(s)

For convenience in table handling only values for single lines or doublets (most of the observed cases) can be entered. If higher multiplicities are actually needed small changes have to be made in the table and program format. Possibility for 50 different atomic transitions.

N.B. : in case of doublets, columns 3, 5, 7, 9, and 11 refer to the reddest component, and columns 4, 6, 8, 10 and 12 to the bluest.

8.4.3 Format of the table for the emission line model (EMI.TBL)

The lines are assumed to be gaussian:

$$I(\lambda) = HMAX * EXP\left[-Ln2\left(\frac{2(\lambda - POS)^2}{FWHM}\right)^2\right]$$

Column	Name	
1	IDENT	
2	HMAX	(Flux unit)
3	POS	(Ångstrom)
4	FWHM	(Ångstrom)

Possibility for 50 emission lines.

8.4.4 Format of the table for the absorption line model (ABSC.TBL)

INPUT

Column	Name	
1	: IDENT	Identification of the transition
2	: NH	Column density in nb (atoms)/cm^2
3	: POS	Position of the absorption in Å In the case of a doublet, indicate only the position of the reddest component; the other will be set automatically.
4	: BVAL	b parameter $\left(\sqrt{\frac{2kT}{m}}\right)$ in km/s
5	: TRANS	This number indicates the line number where the corresponding atomic transition is identified in table ABSP.

OUTPUT

6	: VELOCITY	velocity of the cloud with respect to the source.
7	: ZA	redshift of the cloud, relative to the observer.

Possibility for 50 absorption lines.

N.B.

- All the data should be entered in the table as real. [except : IDENT (character)]
- The unit for wavelength is Å (c.f. coefficient for the continuum and the emission on absorption lines).

8.5 Dimensions of the Output Images

1. PSF
   - STEP : given by the user (keyword CLDDIM)
   - START : -4 • FWHM
   - NPIX : -2 • START/STEP
2. SPECTRA
   - SYNEMI
     - STEP : keyword CLDDIM
     - START : keyword CLDDIM
     - NPIX : keyword CLDDIM
   - SYNABS
     - STEP :)
     - START :) The same as for SYNEMI, copied directly from its descriptor.
     - NPIX :)

N.B. NPIX (SYNEMI,...) in any case must be greater than NPIX (PSF) but is still limited up to 5000.

8.6 References

Morton D.C. and Smith W.H., 1973, Ap. J. Suppl. Vol. 26 p. 333.
Spitzer L.Jr., 1978 Physical Processes in the intersteller medium, Ed Wiley.
Strömgren B., 1948 Ap J Vol. 108 p. 242.

8.7 Example

A concrete application is displayed step by step by the command: `TUTORIAL/CLOUD`.

The data used for this example are CaII absorption features observed in the spectrum of SN 1986g in NGC 5128 (observer: S. d'Odorico, instrument: CASPEC).

The two images (components K and H) have been pre-reduced so that the continuum is normalised to 1. Sampling is 0.05Å per pixel and the measured instrumental resolution, 17km/s.

The final cloud model consists of 12 absorption doublets. The corresponding image includes both components K and H and has been convolved by a gaussian PSF of 0.22Å (17km/s) FWHM. (The results are consistent with those derived by the ALAS software).

8.8 Acknowledgements

The program ALAS, developed by M. Pettini, was a source of inspiration during the design of this package. We are indebted to S. D'Odorico for providing the data used in the example and would like to thank D. Baade for a number of useful suggestions.

Chapter 9

Test Data

9.1 Introduction

Modern solid-state detectors produce data which differ in two main aspects from classical photographic images: (a) a much higher S/N (signal to noise ratio) and (b) the rigid spatial digitisation of the incoming flux. The importance of (b) becomes evident if one considers that for most astronomical instrument/detector combinations the sampling of the point–spread function is hardly adequate and usually just fulfills the Nyquist criterion which states that there should be at least two detector elements to a resolution element. Thus, in good data, the information content of each pixel is highly significant, and one must ensure that this information is not lost or degraded during the data reduction process.

While the user wants to be assured of this, it appears unrealistic to expect that complicated algorithms can always be documented to the sub–pixel level. With the accelerating exchange and export of software in astronomy, inquiries to the people who have written or are maintaining the software are often difficult or impractical. The only solution is testing. This must be done on some suitable test images so that the results can immediately be checked with, e.g., a pocket calculator. This evidently requires images with some simple geometrical patterns.

The primary use of such images is in assessing accuracy of operations involving transformations of images.

The generation of such test images is a non–trivial task, both on the conceptualisation and on the implementation levels because the test images must be independently conceived of the software items which will be subsequently used on them, and the numerical precision of the test images (we have aimed at 1 in 10^4) must exceed the S/N of typical observations by an order of magnitude. We therefore offer a small set of programs for the generation of such test images and, in cases where the required execution time is prohibitive, the frames themselves. The objective of the package is to allow tests of all routines which do not maintain the identity of a pixel such as rebinning, rotation and filtering. Fitting can also be considered. The usage of the programs and images that we provide as a reference test package should also facilitate documentation of possible application software malfunctions and their subsequent repair. We provide in detail below the images constructed, with the precisions obtained, and mention implementation details where relevant.

It is clearly not possible to provide every image configuration of interest but it may be noted that the images which are provided may be combined in any desired manner. This refers in particular to the rotating or flipping of the patterns described below, or the adding of various backgrounds (also provided) to these images. Images with spatial frequency spectra which come close to those of real data can be obtained by filtering.

The commands described in the following sections are activated by setting the GEOTEST *context*, by using the command `SET/CONTEXT GEOTEST` in a MIDAS session.

9.2 2–Dimensional Images

There are six two–dimensional images available, — these are provided instead of the relevant programs to create them on account of intensive computation time requirements. They may be copied into one's directory from the MIDAS directory containing the source code of the GEOTEST context. All images are MIDAS Bulk Data Frames of dimensions 128×128, with the patterns approximately centered. All frames are of STEP 1 and START 1, in both dimensions. All patterns have infinitely sharp edges. The coordinates of a pixel pertain to its center, *not* to any extremity; this convention is rigorously followed.

9.2.1 Patterns

Images BARS and OFFBARS, below, are of SCALE = 3; BARS30 and BARS60 have SCALE = 4; and RINGS and OFFRINGS have SCALE = 2. All of these images have CUTS defined as 0, 100. Reference may be made to Figs. 16.1 and 16.2 for a reproduction of these images, produced using the SET/SPLIT feature of MIDAS.

BARS.BDF consists, firstly, of three rectangular bars; each of length 25 pixels; of width 1, 4, and 9 pixels; horizontally positioned; and flush with pixel boundaries. The uniform flux density in each pattern is 100 units per pixel. The three rectangular patterns are also vertically positioned, again flush with pixel boundaries. Note that because of this alignment of images and pixel grids and because of the dimensions of the images being in integer numbers of pixels — which is not likely to occur in reality — no effect of the spatial discretisation (i.e. of the necessarily finite pixel size) is visible.

In order to illustrate the case of a point spread function which in one dimension is extremely undersampled, a delta function is also included (the lower left hand pattern in BARS). It is defined as: $f(x, y) = 100$ where $y = y_0$ and $x_{min} \leq x \leq x_{max}$, for given y_0, x_{min} and x_{max}. Note that the one–pixel wide pattern, flush with pixel boundaries, cannot be distinguished from the delta function.

OFFBARS.BDF has the same patterns as BARS, but offset to a small amount in x and y relative to pixel boundaries. In the horizontal (x) direction, the offset is 0.4 of a pixel side length, and in the vertical (y) direction it is 0.2 of a pixel (both in the positive direction).

Figure 9.1: Images BARS, OFFBARS, BARS30 and BARS60.

This yields for the rectangular patterns flux values of 32.0, 40.0, and 8.0 in the top left corner, along the boundary, and in the top right corner respectively; corresponding figures for the lower part of each pattern are: 48.0, 60.0, and 12.0; finally the flux along the left and right boundaries are 80.0 and 20.0. The comparison with BARS shows the effect of discrete steps quite clearly.

The delta function in OFFBARS is defined on a different interval of the x–axis. Its constant flux is apportioned to the pixels it runs through in the horizontal direction, but in the y (vertical axis) direction it has infinitesimal extension. In other words, shifting a delta function perpendicularly to the axis to which it is parallel has no effect while this is not true of the 1-pixel wide bar.

Note that flux conservation is at all times verified, both locally (single pixel) and globally.

BARS30.BDF rotates the horizontal patterns of BARS through -30 degrees.

In all cases the center of rotation is the pixel in the lower left hand corner of the rectangular patterns. These pixel coordinates are 10, 20 and 34 pixel units distant from the delta function whose left hand pixel has coordinates (42,52).

The delta function is again taken as a line of given starting point, of given length, of given angle to the horizontal, and of infinitesimal width. Note the difference between this and the 1–pixel wide bar.

In order not to depend on any interpolation algorithms etc., creation of these patterns was carried out by subdividing pixels, rotating the subpixels, and accumulating flux contributions of these subpixels. For a subdivision of 299×299 the elapsed time

Figure 9.2: Images RINGS and OFFRINGS.

may be a clearer indicator of reconstruction time: BARS30 took approximately 30 hours on a VAX 11/785 machine.

An estimate of precision was obtained in the following way. For the central portion of the larger rectangles, flux of 100 ought to obtained. If pixel subdivision is coarse, values greater than or less than 100 will be obtained. Hence, the sampling of pixels where the value 100 is expected allows an estimate of accuracy, and should additionally vary proportionally to n^2 where pixel subdivision is $n \times n$. (The latter statement is based on empirically noting that a small constant number of subpixels — usually 1 — were "scattered" in being projected. This n^2 precision dependence was empirically verified.) For $n = 299$, accuracy defined in this way was determined as 0.003 units, i.e. in no pixel the residual from 100 units is larger than this value.

BARS60.BDF was constructed in an analogous manner, with the same precision, as BARS30. In this case, the angle of rotation was +60 degrees. The coordinates of the left hand pixel of the delta function are again (42,52), and the other patterns have lower left hand pixels distant by 10, 20 and 34 pixel units.

RINGS.BDF relates to centered, circular patterns. The patterns here are a delta function (infinitely thin) at radius 10 units, and rings of thicknesses 1, 5 and 10 units at inner surface radii 22, 33 and 48 steps. Uniform flux density of 100 units/pixel is used. Additionally flux 100 is located in the pixel at the center of the set of concentric rings. Note again the difference between the delta function and the ring of thickness 1: the former distributes flux (100 units) to any pixel which is overlapped by a mathematically–defined circular curve; while the ring of thickness 1 distributes flux to pixels proportionally to their overlap with the ring.

Total flux in the delta function was defined as $2\pi r \times 100$, and subdivision was carried out in intervals of 0.025 degree.

Technically, the approach adopted in constructing these patterns was to subdivide pixels, rotate (or otherwise project) the small subpixels, and accumulate the flux (cf. BARS30 and BARS60). This allows the precision of the annuli to be assessed by noting that the approximation improves as the subpixels get smaller, and that the typical flux excess or deficit in one output pixel amounts to the flux content of one subpixel. Per output pixel, this gives a maximum error of $\frac{100}{n^2}$ (where n^2 subpixels correspond to the original pixel with flux 100), and a mean error of half this value (where we assume uniform statistical fluctuation; a histogram analysis of the frames created confirms this). For $n = 101$, we therefore obtain a maximum error estimate of less than 0.01 units.

RINGS is perfectly symmetric under rotation through multiples of 90 degrees.

OFFRINGS.BDF is analogous to RINGS, except offset from the center of a pixel by (0.3,0.25) units. Note the effect of this offset on the flux distributed uniformly in a square of pixel dimensions at the center; and note also that shifting the annuli as done in OFFRINGS necessarily destroys the symmetry of the annuli. In other respects (time required for construction, precision) OFFRINGS presents similar characteristics to RINGS.

9.2.2 Backgrounds

The following commands may be executed within the GEOTEST context (activated with the `SET/CONTEXT GEOTEST` command).

CREATE/RAMP produces square two–dimensional images of user–specified dimensions, with mean flux per pixel of 100 units, of user–specified slope of the plane, and with the plane itself sloping at a position angle. The latter can vary between 0 and 90 degrees.

For large slopes, the flux assigned to some pixels will be negative (— this being dependent on the position angle).

CREATE/WAVE produces an image with sinusoidal modulations along the horizontal axis. Again, the image is two–dimensional, of user–specified dimensions, and with user–specified amplitude and period. The *sin* function is always evaluated at the center of the pixel. The mean flux per period is 0 units.

9.3 1–Dimensional Images ("Spectra")

As with the previous commands, the following are implementable within the GEOTEST context. Again, all frames are of STEP 1 and START 1; and the coordinates of a pixel pertain to its center, *not* to any extremity.

CREATE/SPC1 generates a 1–dimensional image with a sinusoidal pattern of mean flux 100 units and with an optional slope. (This is analogous to the WAVE 2-dimensional case.) Input parameters are: slope (default = 0.0), period of sinewave (default is 8.5 pixel units), amplitude of sinewave (default, half peak–to–peak, is 10.0), and the phase of sinewave (default is 0.0). Additionally, the number of pixels in the artificial image can be specified, and is defaulted to 128.

CREATE/SPC2 produces a 1–dimensional image with an arbitrary periodic pattern whose period (an integer number between 1 and 20, both included) and the flux values of each pixel can be specified by the user. (Thus, for a user–specified period of 13, 13 values are obtained which define the 1–dimensional image pixel values.)

Two options are available for inputting these numbers: either the keyword `RIPVALS` is defined and initialised by the user prior to executing `CREATE/SPC2`; or by default a period of 10 is assumed, together with the values 4, 5, 6, 3, 2, 7, 1, 4, 5, 0.

If a negative period or a period of value larger than 20 is specified, a message indicates to the user that a period of 10 or a period of value *mod*(period,20), respectively, is employed.

Three additional parameters are user–specifiable: the slope of the wave pattern generated (default 0.0); the phase of the pattern (an integer number $n \leq$ period meaning that the first pixel of the image will have the flux of the nth element of `RIPVAL`; defaulted to 1); and the dimension of the 1–dimensional image (default 128 pixels).

CREATE/SPC3 produces an artificial spectrum with lines whose location are defined by a MIDAS table. The table `LINES` (used as default) provides an example of approximately randomly distributed lines with some relatively broad gaps and also some close pairs or groups. Note that the locations must be given as integer numbers between 1 and 220 (both included).

The table `LINES.TBL` must be copied into one's own directory for use. It is available in the MIDAS directory containing the source code of the GEOTEST context.

The spectrum generated consists of 660 pixels and repeats the pattern defined by the line positions three times. In the first 220 pixels, the line locations are used as read from the table. In the second 220 pixels, the lines are relocated on the edges of pixels (by subtracting 0.5 pixel units from the initial line positions). In the third set of 220 pixels, a user–specifiable offset is added in order to reposition the lines relative to pixel centers (the default is 0.645).

Should a line extend beyond the boundary of any of the 220–wide segments, then it is truncated at the boundary.

Two types of point spread functions (PSF) are available, namely boxes (default) and Gaussians. The full width of a box (default 1.0) or the full width at half maximum (FWHM) of a Gaussian (default 1.0) are user suppliable. If, for a box-like PSF, the width given is less than 0.001, then a delta function is used instead. For Gaussians, a FWHM below 0.2 pixel is not acceptable. In all cases, the *peak value of the PSF* is normalised to 1 unit. (Note that this implies that the actual appearance of the PSF may be quite different due to the effects of discretisation due to a finite step size.) Gaussians are evaluated every $1/100^{th}$ of a pixel step size, (this is necessary in the case when most of the Gaussian is within a pixel), and truncated at 5 standard deviations from the mean.

9.4 Noise

Real data always comes with some noise, and it is important that the software does not worsen the noise spectrum (in pixel as well as in data space) more than necessarily. With the command CREATE/RANDOM_IMAGE, images with various random–number distributions can be generated. For most purposes it will be sufficient to just add such an image (using COMPUTE/IMAGE) to the frame with the test pattern. This would simulate the case where there is only detector read–out noise. If for some reason the simulation of photon noise is more desirable, this case can be approximated by scaling (again using COMPUTE/IMAGE) the amplitude of a suitable noise distribution (output of CREATE/RANDOM_IMAGE) with the square root of the pattern before it is added to the pattern frame. Evidently, all kinds of combinations of these two cases can be used.

If the STMODEL context is available (activated by `SET/CONTEXT MODEL` in a MIDAS session), then the command `HPOIS` may also be used for the simulation of noise.

9.5 Other Images

Fractal geometry has excited much interest in recent years. Generating the type of intricate pattern given by a Mandelbrot curve is a computation–intensive task, perhaps requiring hours of CPU time. A command is included for experimentation, `CREATE/ART_IMAGE`, with which one can create Mandelbrot and Julia curves. See relevant help files for details.

9.6 Command Syntax Summary

Table 9.1 lists the commands for creating artificial images described in this chapter. These commands are available in the GEOTEST *context*.

Command	
CREATE/RAMP	`outimage [slope] [angle] [dimension]`
CREATE/WAVE	`outimage [amplitude] [period] [dimension]`
CREATE/SPC1	`outimage [slope] [amplitude] [period] [phase] [dimension]`
CREATE/SPC2	`outimage [period] [slope] [amplitude] [phase] [dimension]`
CREATE/SPC3	`outimage [psf-option] [centring] [table] [boxwidth-or-fwhm]`
CREATE/ART_IMAGE	`outimage [dims] [starts,steps] [func_type] [coefs]`

Table 9.1: Geometrical Test Image Commands

Chapter 10

Multivariate Analysis Methods

10.1 Introduction

Compared to the past, ever larger amounts of data are being collected in astronomy, and the rate will continue to accelerate in the next decades. It is therefore necessary to work on bigger samples if full advantage is to be taken of all accessible information. It is also necessary to derive as much information as possible from the diversity of the data, rather than restricting attention to subsets of it.

One way to work effectively on large samples is to apply, and if necessary to develop, suitable statistical methods. Multivariate data analysis methods are not intended to replace physical analysis: these should be seen as complementary, and statistical methods can effectively be used to run a rough preliminary investigation, to sort out ideas, to put a new ("objective" or "independent") light on a problem, or to point out aspects which would not come out in a classical approach. Physical analysis is necessary subsequently to refine and interpret the results.

The multivariate methods implemented all operate on MIDAS tables. Such tables cross objects (e.g. ultraviolet spectra, spiral galaxies, or stellar chemical abundances) with variables or parameters.

Among widely-used multivariate methods :

1. Principal Components Analysis
2. Hierarchical Cluster Analysis
3. Non–Hierarchical Clustering, or Partitioning
4. Minimal Spanning Tree
5. Discriminant Analysis
6. Correspondence Analysis

We will look briefly at each of these in turn. Since comprehensive ancillary documentation is available (see references), we will not dwell on background material here. The wide–ranging applications for which multivariate statistical methods have been employed in astronomy also cannot be catalogued here (again, see the references given). In the following sections we will concentrate on various practical aspects of the methods. Note that the MVA *context* is required to activate the relevant commands (by using the command `SET/CONTEXT MVA` in a MIDAS session).

10.2 Principal Components Analysis

Among the objectives of Principal Components Analysis are the following.

1. dimensionality reduction;
2. the determining of linear combinations of variables;
3. feature selection: the choosing of the most useful variables;
4. visualisation of multidimensional data;
5. identification of underlying variables;
6. identification of groups of objects or of outliers.

The tasks required of the analyst to carry these out are as follows:

1. In case of a table of dimensions $n \times m$, each of the n rows or objects can be regarded as an m–dimensional vector. Finding a set of $m' < m$ principal axes allows the objects to be adequately characterised on a smaller number of (artificial) variables. This is advantageous as a prelude to further analysis as the $m - m'$ dimensions may often be ignored as constituting noise; and, secondly, for storage economy (sufficient information from the initial table is now represented in a table with $m' < m$ columns). Reduction of dimensionality is practicable if the first m' new axes account for approximately 75 % or more of the variance. There is no set threshold, — the analyst must judge. The cumulative percentage of variance explained by the principal axes is consulted in order to make this choice.

2. If the eigenvalue is zero, the variance of projections on the associated eigenvector is zero. Hence the eigenvector is reduced to a point. If this point is additionally the origin (i.e. the data is centred), then this allows linear combinations between the variables to be found. In fact, we can go a good deal further: by analysing second–order variables, defined from the given variables, quadratic dependencies can be straightforwardly sought. This means, for example, that in analysing three variables, y_1, y_2, and y_3, we would also input the variables y_1^2, y_2^2, y_3^2, y_1y_2, y_1y_3, and y_2y_3. If the linear combination

$$y_1 = c_1 y_2^2 + c_2 y_1 y_2$$

exists, then we would find it. Similarly we could feed in the logarithms or other functions of variables.

3. In feature selection we want to simplify the task of characterising each object by a set of attributes. Linear combinations among attributes must be found; highly correlated attributes (i.e. closely located attributes in the new space) allow some attributes to be removed from consideration; and the proximity of attributes to the new axes indicate the more relevant and important attributes.

4. In order to provide a convenient representation of multidimensional data, planar plots are necessary. An important consideration is the adequacy of the planar representation: the percentage variance explained by the pair of axes defining the plane must be looked at here.

5. PCA is often motivated by the search for latent variables. Often it is relatively easy to label the highest or second highest components, but it becomes increasingly difficult as less relevant axes are examined. The objects with the highest loadings or projections on the axes (i.e. those which are placed towards the extremities of the axes) are usually worth examining: the axis may be characterisable as a spectrum running from a small number of objects with high positive loadings to those with high negative loadings.

6. A visual inspection of a planar plot indicates which objects are grouped together, thus indicating that they belong to the same family or result from the same process. Anomalous objects can also be detected, and in some cases it might be of interest to redo the analysis with these excluded because of the perturbation they introduce.

10.3 Cluster Analysis

The routines implemented are `CLUSTER` which has 8 options for hierarchical clustering and `PARTITION` which carries out non–hierarchical clustering. We will look at the hierarchical options available first.

The automatic classification of the n row–objects of an n by m table generally produces output in one of two forms: the assignments to clusters found for the n objects; or a series of clusterings of the n objects, from the initial situation when each object may be considered a singleton cluster to the other extreme when all objects belong to one cluster. The former is non–hierarchical clustering or partitioning.

The latter is hierarchical clustering. Brief consideration will show that a sequence of $n-1$ agglomerations are needed to successively merge the two closest objects and/or clusters at each stage, so that we have a set of n (singleton) clusters, $n-1$ clusters, ..., 2 clusters, 1 cluster. This is usually represented by a hierarchic tree or a *dendrogram*, and a "slice" through the dendrogram defines a partition of the objects. Unfortunately, no rigid guideline can be indicated for deriving such a partition from a dendrogram except that large increases in cluster criterion values (which scale the dendrogram) can indicate a partition of interest.

In carrying out the sequence of agglomerations, various criteria are feasible for defining the newly–constituted cluster:

The minimum variance criterion (method `MVAR`) constructs clusters which are of minimal variance internally (i.e. compact) and maximal variance externally (i.e. isolated). It is useful for synoptic clustering, and for all clustering work where another method cannot be explicitly justified.

The minimum variance hierarchy: All options, with the exception of `MNVR`, construct a set of Euclidean distances from the input set of n vectors. Thus the internal storage required is large. Option `MNVR` allows a minimum variance hierarchy (identical to option `MVAR`) to be obtained, without requiring storage of distances. Computational time is slightly higher than the latter option.

The single link method (method `SLNK`) often gives a very skew or "chained" hierarchy. It is therefore not useful for summarising data, but may indicate very anomalous or outlying objects, — these will be among the last to be agglomerated in the hierarchy.

The complete link method (method `CLNK`) often does not differ unduly from the minimum variance method, but its restrictive criterion is not suitable if the data is noisy.

The average link method (method `ALNK`) is a reasonable compromise between the (lax) single link method and the (rigid) complete link criterion: all of these methods may be of interest if a graph representation of the results of the clustering is desired.

The weighted average link method (method `WLNK`) does not take the relative sizes of clusters into account in agglomerating them. This, and the two following methods, are included for completeness and for consistency with other software packages, but are not recommended for general use.

The median method (method `MEDN`) replaces a cluster, on agglomeration, with the median value. It is not guaranteed that these criterion values will vary monotonically, and this may present difficulty with the interpretation of the dendrogram representation.

The centroid method (method `CNTR`) replaces a cluster, on agglomeration, with the centroid value. As in the case of the last option, reversals or inversions in the hierarchy are possible.

The Minimal Spanning Tree, which is closely related to the single link method, has been used in such applications as interferogram analysis and in galaxy clustering studies. It is useful as a detector of outlying data points (i.e. anomalous objects).

Routine `PARTITION` operates in one two options. For both, a partition of minimum variance, given the number of clusters, is sought. Two iterative refinement algorithms (minimum distance or the exchange method) constitute the options available.

10.4 Discriminant Analysis

Discriminant Analysis may be used for two objectives: either we want to *assess* the adequacy of classification, given the group memberships of the objects under study; or we wish to *assign* objects to one of a number of (known) groups of objects. Discriminant Analysis may thus have a descriptive or a predictive objective.

In both cases, some group assignments must be known before carrying out the Discriminant Analysis. Such group assignments, or labelling, may be arrived at in any way. Hence Discriminant Analysis can be employed as a useful complement to Cluster Analysis (in order to judge the results of the latter) or Principal Components Analysis. Alternatively, in star–galaxy separation, for instance, using digitised images, the analyst may define group (stars, galaxies) membership visually for a conveniently small *training set* or *design set*.

Methods implemented in this area are Multiple Discriminant Analysis, Fisher's Linear Discriminant Analysis, and K-Nearest Neighbours Discriminant Analysis.

Multiple Discriminant Analysis (MDA) is also termed Discriminant Factor Analysis and Canonical Discriminant Analysis. It adopts a similar perspective to PCA: the rows of the data matrix to be examined constitute points in a multidimensional space, as also do the group mean vectors. Discriminating axes are determined in this space, in such a way that optimal separation of the predefined groups is attained. As with PCA, the problem becomes mathematically the eigenreduction of a real, symmetric matrix. The eigenvalues represent the discriminating power of the associated eigenvectors. The n_Y groups lie in a space of dimension at most $n_Y - 1$. This will be the number of discriminant axes or factors obtainable in the most common practical case when $n > m > n_Y$ (where n is the number of rows, and m the number of columns of the input data matrix).

Linear Discriminant Analysis is the 2-group case of MDA. It optimally separates two groups, using the *Mahalanobis metric* or *generalized distance*. It also gives the same linear separating decision surface as Bayesian maximum likelihood discrimination in the case of equal class covariance matrices.

K-NNs Discriminant Analysis : Non–parametric (distribution–free) methods dispense with the need for assumptions regarding the probability density function. They have become very popular especially in the image processing area. The K-NNs method assigns an object of unknown affiliation to the group to which the majority of its K nearest neighbours belongs.

There is no best discrimination method. A few remarks concerning the advantages and disadvantages of the methods studied are as follows.

- Analytical simplicity or computational reasons may lead to initial consideration of linear discriminant analysis or the NN–rule.

- Linear discrimination is the most widely used in practice. Often the 2-group method is used repeatedly for the analysis of pairs of multigroup data (yielding $\frac{k(k-1)}{2}$ decision surfaces for k groups).

- To estimate the parameters required in quadratic discrimination more computation and data is required than in the case of linear discrimination. If there is not a great difference in the group covariance matrices, then the latter will perform as well as quadratic discrimination.

- The k–NN rule is simply defined and implemented, especially if there is insufficient data to adequately define sample means and covariance matrices.

- MDA is most appropriately used for *feature selection*. As in the case of PCA, we may want to focus on the variables used in order to investigate the differences between groups; to create synthetic variables which improve the grouping ability of the data; to arrive at a similar objective by discarding irrelevant variables; or to determine the most parsimonious variables for graphical representational purposes.

10.5 Correspondence Analysis

Correspondence Analysis may be described as a PCA in a different metric (the χ^2 metric replaces the usual Euclidean metric). Mathematically, it differs from PCA also in that points in multidimensional space are considered to have a mass (or weight) associated with them, at their given locations. The percentage *inertia* explained by axes takes the place of the percentage variance of PCA, — and in the former case the values can be so small that such a figure of merit assumes less importance than in the case of PCA. Correspondence Analysis is a technique in which it is a good deal more difficult to interpret results, but it considerably expands the scope of a PCA–type analysis in its ability to handle a wide range of data.

While PCA is particularly suitable for quantitative data, CA is recommendable for the following types of input data, which will subsequently be looked at more closely: frequencies, contingency tables, probabilities, categorical data, and mixed qualitative/categorical data.

In the case of *frequencies* (i.e. the ij^{th} table entry indicates the frequency of occurrence of attribute j for object i) the row and column "profiles" are of interest. That is to say, the relative magnitudes are of importance. Use of a weighted Euclidean distance, termed the χ^2 distance, gives a zero distance for example to the following 5–coordinate vectors which have identical *profiles* of values: (2,7,0,3,1) and (8,28,0,12,4). Probability type values can be constructed here by dividing each value in the vectors by the sum of the respective vector values.

A particular type of frequency of occurrence data is the *contingency table*, — a table crossing (usually, two) sets of characteristics of the population under study. As an example, an $n \times m$ contingency table might give frequencies of the existence of n different metals in stars of m different ages. CA allows the study of the two sets of variables which constitute the rows and columns of the contingency table. In its usual variant, PCA would privilege either the rows or the columns by standardizing: if, however, we are dealing with a contingency table, both rows and columns are equally interesting. The "standardizing" inherent in CA (a consequence of the χ^2 distance) treats rows and columns in an identical manner. One byproduct is that the row and column projections in the new space may both be plotted on the same output graphic presentations (— the lack of an analogous direct relationship between row projections and column projections in PCA precludes doing this in the latter technique).

Categorical data may be coded by the "scoring" of 1 (presence) or 0 (absence) for each of the possible categories. Such coding leads to *complete disjunctive coding*. CA of an array of such complete disjunctive data is referred to as Multiple Correspondence Analysis (MCA) (and in fact such a coding of categorical data is, in fact, closely related to contingency table type data).

Dealing with a complex astronomical catalogue may well give rise in practice to a mixture of quantitative (real valued) and qualitative data. One possibility for the analysis of such data is to "discretize" the quantitative values, and treat them thereafter as categorical. In this way a set of variables — many more than the initially given set of variables — which is homogenous, is analysed.

10.6 Related Table Commands

A tutorial command (`TUTORIAL/MVA`) may be used to see the very straightforward way in which the methods described here are used. Help commands may also be used for the required syntax.

A few MIDAS Table commands, of particular interest for the use of multivariate routines, are mentioned here for convenience. Note that the multivariate routines use all columns and rows in the input tables, and hence tables must first be "rearranged", as desired, before analysis. MIDAS subsequently offers a powerful environment for plotting results.

To show the characteristics of table mytab, to read its values to the screen, to print it out on a good quality device, to delete column 4 of mytab, and to construct table mytab2 from mytab, with three columns (PC1, PC2 and PC3) in the former:

```
SHOW/TAB mytab
READ/TAB mytab
PRINT/TAB mytab
DELE/TAB mytab #4
PROJ/TAB mytab mytab2 :PC1 :PC2 :PC3
```

To select group 2 members, defined in column 4, from table MYTAB; then plot coordinates defined in columns 1 and 2; then select group 3 members; overlay the coordinates of these points on the initial plot with a different symbol; and finally to get a good quality hard–copy representation:

```
ASSIGN/PLOT LASER NOSPOOL
SELECT/TAB mytab #4.EQ.2
SET/PLOT STYPE=2
PLOT/TAB mytab #1 #2
```

```
SELE/TAB mytab ALL
SELE/TAB mytab #4.EQ.3
SET/PLOT STYPE=6
OVER/TAB mytab #1 #2
SEND/PLOT LASER
```

Since the scales of group 3 members, may differ from the scale of group 2 members in the above, it may be advisable to issue a `SET/PLOT XAXIS= YAXIS=` command, with attendant minima and maxima for the axis, before plotting.

For further details of parameters to these commands, and for other relevant commands, the Tables chapter in this manual should be consulted.

Also in this Tables chapter are some routines for regression.

10.7 References

1. A. Heck, F. Murtagh, D. Ponz, "The increasing importance of statistical methods in astronomy", *The Messenger*, No. 41, 22–25, Sept. 1985.

2. An inventory of astronomical and general references, under headings related to major multivariate methods, is available. It is described in F. Murtagh and A. Heck, "An annotated bibliographical catalogue of multivariate statistical methods and of their astronomical applications", *Astronomy and Astrophysics Supplement Series* **68** 113–115, 1987.

3. A text–book fully describing the algorithms implemented here and with many examples is: F. Murtagh and A. Heck, *Multivariate Data Analysis*, Kluwer (Astrophysics and Space Science Library), Dordrecht, 1987.

10.8 Command Syntax Summary

Table 10.8 lists commands for Multivariate Data Analysis. These commands are activated by setting the MVA *context* (by means of the command `SET/CONTEXT MVA` in a MIDAS session).

Command	
PCA/TABLE	intable outable [analysis option]
	[row/column analysis] [no. cols. in outable]
	[eigenvectors outable]
CLUSTER/TABLE	intable outable [method]
PARTITION/TABLE	intable outable [number of classes]
	[algorithm] [min. cardinality] [seed value]
CORRESPONDENCE/TABLE	intable outable [row/column analysis]
	[no. cols. in outable] [eigenvectors outable]
LDA/TABLE	intable outable
MDA/TABLE	intable outable [eigenvectors outable]
KNN/TABLE	training-set-table no.-of-group-1-members
	test-set-table no.-of-NNs

Table 10.1: Multivariate Data Analysis Commands

Chapter 11

DAOPHOT II: The Next Generation

This manual is only available as separate file in postscript format, and you can download it on the ESO-Documentation WWW page:
http://www.eso.org/projects/esomidas/midas-document.html

The DAOPHOT II manual is intended as a guide to the use of the digital stellar photometry reduction program DAOPHOT II: The Next Generation. Brief descriptions of the major routines are provided to aid the user in the use of the program, and to serve as an introduction to the algorithms for programmers called upon to make modifications. A more complete understanding of the methods used can best be obtained by direct examination of the source code and the comment statements embedded therein.

DAOPHOT Classic was originally constructed within the framework of a computer program given to Linda Stryker and the Dominion Astrophysical Observatory by Jeremy Mould (Caltech). The aperture-photometry portion of the current program is still somewhat influenced by algorithms and subroutines developed at Kitt Peak National Observatory, by way of the POORMAN code developed by Jeremy Mould and Keith Shortridge at Caltech. POORMAN was first made to run on the DAO VAX by Linda Stryker, and was modified and generalized by Ed Olszewski. Over the years, Peter Stetson has replaced all of the subroutines with others of his own writing. In DAOPHOT II: The Next Generation all of the major algorithms and actual FORTRAN coding are done by Stetson, with the exception of those Figaro, IRAF, or MIDAS routines which are used for the image input and output. Stetson is solely responsible for bugs and algorithm-related problems. If any such problems are found, please contact Peter Stetson; for major problems, hard-copy documentation of the circumstances and nature of the difficulty would be highly desirable.

***** **NOTE** *****

This manual has been modified to reflect the new VMS/Unix compatible version of DAOPHOT II: The Next Generation. If you are still running DAOPHOT Classic you will find many places where this manual is inappropriate. The last appendix (Appendix V) shows how to enable the DAOPHOT software within MIDAS and to convert DAOPHOT tables to MIDAS tables and vice versa.

Chapter 12

Time Series Analysis

12.1 Introduction

In spite of enormous amounts of existing software and literature on time series analysis (TSA), only comparatively little work has been done concerning those time series which are most often encountered in astronomy, namely unevenly sampled series. Hence, our first reason for yet another TSA package is to provide tools to handle such series. The second reason is to supply astronomers with modern tools enabling a more complete statistical evaluation of the results than practicized so far.

The interests of galactic and extragalactic astronomers in time series analysis differ markedly. The former are particularly interested in the analysis of genuinely periodic variations, while the latter are more often concerned with stochastic phenomena. We have implemented tools to satisfy the basic needs of both types of users, with emphasis on periodic variations. This chapter provides general guidelines for the user of the MIDAS TSA package. Detailed technical information is provided separately for each command of the package in both Volume A of the MIDAS User Guide and the interactive MIDAS `HELP` utility.

The present chapter is not meant to be a substitute for a textbook on time series analysis. We recommend the publications by Deeming (1976) and Bloomfield (1976) for a general introduction to TSA and works by Brandt (1970) and Eadie *et al.* (1971) for the statistical background. Relevant statistical tables are listed in Brandt (1970) and Abramovitz & Stegun (1972), and a code for the computation of the probability functions is provided by Press et al. (1986). However, we do explain the concepts used in the description of our software and point out their statistical context (Sect. 2). We will discuss Fourier analysis as the crucial example of TSA (Sect3). Sect. 4 contains the general description of the MIDAS TSA package and its commands. A summary of the commands and their syntaxes is listed in Sect. 5. Examples of some typical problems and their treatment within MIDAS are given in Sect. 6.

Note that the present TSA package is not in any way related to the former TSA package first implemented in the January 88 release of MIDAS. The ancestor of this new package originally was a stand-alone package developed at Warsaw Observatory and called

ULA. It was expanded and converted to MIDAS at the European Southern Observatory in Garching.

12.2 Basic principles of time series analysis

The practical problems in the analysis of time series concern

1. the detection of a signal against noise,
2. the estimation of the parameters characterizing the signal, and
3. the presentation of the results.

Presentation and/or evaluation of the results involves a function of the time series called statistics. Detection of a signal, e.g. a period, and evaluation of its properties, e.g. error and significance, then usually mean finding and characterizing a related signal in the associated statistics which is normally based on some model of the signal.

12.2.1 Signals and their models

Signals can be classified broadly into *deterministic* and *stochastic signals.* A deterministic signal, e.g. a *periodic signal,* can be predicted for arbitrary spaces of time. For a stochastic signal, no such prediction can be made beyond a certain time interval, called the *correlation length* l_{corr}. For any finite time series the classification into these two categories is ambiguous so that methods suitable for both stochastic and periodic signals could be applied to any time series with some success (e.g. quasiperiodic oscillations, Sect. 12.2.6).

Usually processes in the source of the signal (e.g. the nucleus of an active galaxy) and/or observational errors introduce a random component into the series, called noise. The analysis of such series usually aims at removing the noise and fitting a *model* to the remaining component of the series. Suitable models can be obtained by shifting a known series by some *time lag*, l, or by repeating fragments of it with some *frequency*, ν. Accordingly, we are speaking of an analysis in the *time* and *frequency domain.* In these domains the correlation length l_{corr} and oscillation frequency ν_o, respectively, have particularly simple meanings. It is transparent that the stochastic signals are analysed more comfortably in the time domain and periodic signals in the frequency domain.

Models usually depend on several parameters. Fitting of the model to the signal means choosing the best set of these parameters. Customarily, the observed series, $X^{(o)}$, is split into the *modeled series* $X^{(m)}$ and the *residuals* of the observations with respect to the model, $X^{(r)} \equiv X^{(o)} - X^{(m)}$.

12.2.2 Signal detection

Paradoxically, signal detection is concerned with fitting models to supposedly random series similarly to mathematical proofs by *reductio ad absurdum* of the antithesis. That is, the hypothesis (antithesis) H_o is made, that the observed series $X^{(o)}$ has properties of a pure noise series, $N^{(o)}$. Then, a model is fitted and series $X^{(m)}$ and $X^{(r)}$ are obtained. If the quality of the model fits to the observations $X^{(o)}$ does not significantly differ from the quality of a fit to pure noise $N^{(o)}$, then H_o is true and we say that $X^{(o)}$ contains no signal but noise. In the opposite case of model fitting $X^{(o)}$ significantly better than $N^{(o)}$, we reject H_o and say that the model signal was detected in $X^{(o)}$. The difference is significant (at some level) if it is not likely (at this level) to occur between two different realizations of the noise $N^{(o)}$.

The quality of the fit is evaluated using a function S of the series $X^{(o)}$, $X^{(m)}$, and $X^{(r)}$. A function of random variables, such as $S(X^{(o)})$, is a random variable itself and is called a *statistic.* A random variable S is characterized by its probability distribution function. Following H_o we use the distribution of S for pure noise signal $N^{(o)}$, $N^{(m)}$ and $N^{(r)}$, to be denoted $p_N(S)$ or simply $p(S)$. Precisely, we shall use the *cumulative probability distribution* function which for a given critical value of the statistic $S = S_o$ supplies the probability $p(S_o)$ for the observed S to fall on one side of the S_o.

The observed value of the statistic and its probability distribution, $S(X^{(o)})$ and $p(S)$ respectively, are used to obtain the probability $p(S(X))$ of H_o being true. That is, if p turns out small, $p < \alpha$, H_o is improbable and $X^{(o)}$ has no properties of $N^{(o)}$. Then we say that the model signal has been detected at the confidence level α. The smaller α is, the more convincing (significant) is the detection. The special realization of a random series which consists of independent variables of common (gaussian) distribution is called (gaussian) white noise. We assume here that the noise $N^{(o)}$ is white noise. Note that in the signal detection process, frequency ν and lag l are considered independent variables and do not count as parameters.

Summarizing, the basis for the determination of the properties of a an observed time series is a test statistic, S, with known probability distribution for (white) noise, $p(S)$.

Let $N^{(o)}$ consist of n_o random variables and let a given model have n_m parameters. Then the modeled series $N^{(m)}$ corresponds to a combination of n_m random variables and the residual series $N^{(r)}$ corresponds to a combination of $n_r = n_o - n_m$ random variables. The proof rests on the observation that orthogonal transformations convert vectors of independent variables into vectors of independent variables. Let us consider an approximately linear model with matrix $\mathcal{M}$ so that $N^{(m)} = \mathcal{M} \circ P$, where P is a vector of n_m parameters. Then $N^{(m)}$ spans a vector space with no more than n_m orthogonal vectors (dimensions). The numbers n_o, n_m and n_r are called the numbers of degrees of freedom of the observations, the model fit, and the residuals, respectively.

12.2.3 Test statistics

The test statistic used for detection is a special case of a function of random variables. Testing the hypothesis H_o using the statistics S is a standard statistical procedure. Important examples of the test statistics are signal variances

$$Var_j \equiv Var[X^{(j)}] = \frac{1}{n_j} \sum_{k=1}^{n_o} \left(x_k^{(j)} \right)^2 \quad j = o, m, r \tag{12.1}$$

For white noise (H_o is true), the distribution of Var_j is a $\chi^2(n_j)$ distribution. It is remarkable that, then, Var_m and Var_r are statistically independent. Note further the inequality $Var_m \geq Var_o \geq Var_r$ which is due to the extra variance, Var_m, in the model signal with respect to the variance, Var_o, of the observations. The equality in these relations holds for pure noise.

The larger the variance of the model series Var_m compared to the residual variance Var_r is, the more significant is the detection or the better is the current parameter estimate, for problems (1) and (2) respectively (Sect. 12.2). Usually, the test statistics S in TSA measure a ratio of two variances. They differ according to the models assumed and the combination of the variances chosen. Since models depend on frequency ν (or time lag l), so do the variances Var_m and Var_r and test statistics S.

The statistics we recommend for use in the frequency domain are the ones introduced by Scargle and the Analysis of Variance (AOV) statistics. These statistics are used in the MIDAS commands `SCARGLE/TSA`, `ORT/TSA` and `AOV/TSA` (Sect. 12.4.6). The SCARGLE/TSA command uses a pure sine model, the ORT/TSA uses Fourier series and the AOV/TSA uses a step function (phase binning). In the time domain, we recommend to use the $Var_r \equiv \chi^2$ statistic with the `COVAR/TSA` and `DELAY/TSA` commands (Sect. 12.4.7). Both `COVAR/TSA` and `DELAY/TSA` are based on a second series of observations which is used for the model. `COVAR/TSA` and `DELAY/TSA` differ in the method used for the interpolation of the series: the former deploys a step function (binning) while the latter relies on an analytical approximation of the autocorrelation function (ACF, Sect. 12.3.2) as a more elaborate approach. Among many other statistics we mention the one by Lafler & Kinman (1965), phase dispersion minimization (PDM) also known as the Whittaker & Robinson statistic (Stellingwerf, 1978), string length (Dvoretsky, 1983), and statistic introduced by Renson (1983).

In the limit of $n_r \rightarrow 0$ ($n_m \rightarrow n_o$) the sums of squares and degrees of freedom converge and so does the variance $Var_r \rightarrow Var_o$ ($Var_m \rightarrow Var_o$). Since $Var_m \geq Var_o$, increasing the number of parameters of a model n_m to n_o implies a decrease of Var_m and a corresponding decrease in the significance of the detection. Therefore, we do not recommend to use models (e.g. long Fourier series, fine phase binning, string length and Renson statistics) with more parameters than are really required for the detection of the feature in question.

In the above limits, Var_o and Var_r (Var_m) become perfectly correlated. Since all statistics named above except AOV use Var_o at least implicitly, their probability distribution may, because of this correlation, differ considerably from what is generally supposed in the literature (Schwarzenberg-Czerny, 1989). However, the correlation vanishes in the asymptotic limit $n_o \rightarrow \infty$ for χ^2, Scargle and Whitteker & Robinson statistics, so that they yield correct results for sufficiently large data sets. Please note that the problem of correlation aggravates for observations with high signal-to-noise ratio, $S/N \rightarrow \infty$, as $Var_m \rightarrow Var_o$, so that the statistics mentioned as using these variances become rather insensitive.

12.2.4 Corrections to the probability distribution

In principle, it is possible to compute a value of the statistic $S(\nu_1)$ for a single frequency ν_1 and to test its consistency with a random signal (H_o). The common procedure of inspecting the whole periodogram for a detected signal corresponds to the N-fold repetition of the single test for a set of trial frequencies, $\nu_n, n = 1, \ldots, N$. The probability of the whole periodogram being consistent with H_o is $1-(1-p)^N \to Np$ for $p \to 0$. The factor N means that there is an increased probability of accepting a given value of the statistic as consistent with a random signal. Therefore, increasing the number of trial frequencies decreases the sensitivity for the detection of a significant signal and accordingly is called the penalty factor for multiple trials or for the frequency bandwidth used. The true number of independent frequencies, N_t, remains generally unknown. It is usually less than the number of resolved frequencies $N_r = \Delta\nu\Delta t$ (Sect. 12.3.1) because of aliasing and still less than the number of computed frequencies N_c, because of oversampling: $N_t \leq N_r \leq N_c$. For a practical and conservative estimate, we recommend to use N_r as the number of trial frequencies, N.

According to the standard null hypothesis, H_o, the noise is white noise. This is not the case in many practical cases. For instance, often the noise is a stochastic process with a certain correlation length $l_{corr} > 0$, so that on average n_{corr} consecutive observations are correlated. Such noise corresponds to white noise passed through a low pass filter which cuts off all frequencies above $1/l_{corr}$. Such correlation is not usually taken into account by standard test statistics. The effect of this correlation is to reduce the effective number of observations by a factor n_{corr} (Schwarzenberg-Czerny, 1989). This has to be accounted for by scaling both the statistics S and the number of its degrees of freedom n_j by factors depending on n_{corr}.

In the test statistic, a continuum level which is inconsistent with the expected value of the statistic $E\{S\}$ may indicate the presence of such a correlation between consecutive data points. A practical recipe to measure the correlation is to compute the residual time series (e.g. with the `SINEFIT/TSA` command) and to look for its correlation length with `COVAR/TSA` command. The effect of the correlation in the parameter estimation is an underestimation of the uncertainties of the parameters; the true variances of the parameters are a factor n_{corr} larger than computed.

In the command individual descriptions, we often refer to probability distributions of specific statistics. For the properties of these individual distributions see e.g. Eadie *et. al.* (1971), Brandt (1970), and Abramovitz & Stegun (1972). The two latter references contain tables. For a computer code for the computation of the cumulative probabilities see Press *et. al.* (1986).

12.2.5 Power of test statistics

The question which method of the detection of features in the test statistic is the most sensitive, is of considerable importance for all practical TSA applications. It directly translates into the comparison of the power of different statistics. Let Y be a signal of some physical meaning, i.e. different from the pure noise. A power β of the test statistic is the probability that H_o is accepted for the given Y: $\beta = 1 - p_Y(S)$. In other words, this is the probability of the false rejection of the no-noise signal Y as pure noise. Generally, the relative power of various statistics depends on the type of the signal Y. An important practical consequence of this fact is that there is no such thing as the universally best method for time series analysis. A method that is good for one type of signals may be poor for another one. An extensive list of signal types and recommended methods is clearly out of scope of the present manual. However, in order to guide the user's intuition we quote below the results of a comparison of the powers of various test statistics for two types of oscillations, both of small amplitude and observed at more or less even intervals (Schwarzenberg-Czerny, 1989):

Let us first consider a sinusoidal oscillation with an amplitude not exceeding the noise. Then, all statistics based on models with 3 parameters have similar values of the test power: power spectrum, Scargle statistics, $\chi^2(3)$ fit of a sinusoid, ORT(1), AOV(3) and corrected PDM(3) statistics. (Please note that we depart here from the conventional notation by indicating in brackets the number n_m of the degrees of freedom of the model - e.g. the number of series terms or phase bins - instead of the number of the residual degrees of freedom n_r.) Statistics with more than 3 parameters, e.g. ORT($n/2+1$), AOV(n), PDM(n) and $\chi^2(n)$ with $n > 3$ and an extended Fourier series have less power. Our final choice is guided by the availability of the analytical probability distribution of the test statistics. Summing up, we recommend tu use for the detection of sinusoidal and other smooth oscillations of small amplitude the statistics with a coarse phase resolution, e.g. ORT(1), Scargle and AOV(3), .

For a narrow gaussian pulse or eclipse of width w repeating with period P the most powerful statistics are these with the matching resolution: ORT($P/2w$), AOV(P/w) and χ^2(P/w). Power spectrum, Scargle, ORT(1), AOV(3), $\chi^2(3)$, ORT(n/2), AOV(n) and $\chi^2(n)$, $n \gg P/w$ all have less power. Note the equivalence of the $\chi^2(3)$ and Scargle's statistic (Lomb, 1976, Scargle, 1982) and the near-equivalence of the power spectrum and Scargle's statistics in the case of nearly uniformly sampled observations. Considering both test power and computational convenience we recommend for signals with sharp features, e.g. narrow pulses or eclipses, to use the ORT and AOV with the resolution matched to the width of these features.

12.2.6 Time domain analysis

As noted above, periodic signals are best analysed in the frequency domain while stochastic signals are usually more profitably analysed in the time domain. The analysis in the time domain often involves the comparison of two different signals while in the frequency domain analyses usually concern only one signal. The expectation value of the covariance function of uncorrelated signals is zero. The expected value of the autocorrelation function (E{ACF}, Sect. 12.3.2) of white noise also is zero everywhere except for 1 at zero lag. The expected ACF of a stochastic signal of correlation length l vanishes outside a range $\pm l$ about the lags. The ACF of a deterministic function does not vanish at infinity. In particular the ACF of a function with period P has the same value, P. Signals of intermediate or mixed type with an ACF which has several maxima spaced evenly by l and a correlation length $L \gg l$ is called a *quasiperiodic oscillation.* Its power is significantly above the noise in the $1/l \pm 1/L$ range of frequencies and its correlation length L is called the *coherence length.*

12.2.7 Presentation and inspection of results

A simple way to graphically present the results of a TSA is to plot the test statistics S against its parameter ν or l, depending on whether the analysis was performed in the frequency or time domain. Plots in the frequency domain are called periodograms. In them, oscillations are revealed by the presence of spectral lines. However, some (often many) lines are spurious and simply arise from random fluctuations of the signal. By means of the confidence level α and the probability distribution of S one can find the critical value S_{crit} for significant features. Examples of statistics used in the time domain are covariance and correlation functions. The correlation of a signal with itself or with another signal produces maxima in these functions at particular lags. Detection of genuine lags then consists of testing the significance of such maxima.

12.2.8 Parameter estimation

In this context, ν (or, in the time domain, l) are no longer independent variables. They are treated like any of the other parameters: i.e. are assumed to be random variables to be estimated from the observations by fitting a model. Parameter estimation in the frequency domain is best done by fitting models using χ^2 statistics (least squares). The MIDAS TSA package contains just one such model, namely Fourier series (`SINEFIT/TSA`). However, note that with its non-linear least-squares fitting package, MIDAS offers very versatile, dedicated tools for model fitting (see Chapter 8 in Vol. 8 of the MIDAS User Guide).

In the time domain, the most important parameters to be estimated from the data are the correlation length of and time lag between the input signals. This measurement can be done with the command `WIDTH/TSA`. The correlation length can be obtained as the width of the line centered at zero lag. The time lag can be measured as the center of the corresponding line in the ACF.

12.3 Fourier analysis: The sine model

12.3.1 Fourier transforms

Transformations which take functions, e.g. x, y as arguments and return functions as results are called operators. The direct and inverse Fourier transform, $\mathcal{F}^{\pm 1}$, and the convolution, $*$, are operators defined in the following way:

$$\mathcal{F}^{\pm 1}[x](\nu) \quad = C_{\pm} \int_{-\infty}^{+\infty} e^{\pm 2\pi i t \nu} x(t)dt \quad = \frac{1}{n_o^{\frac{1\pm 1}{2}}} \sum_{k=1}^{n_o} x_k e^{\pm 2\pi i t_k \nu} \tag{12.2}$$

$$[x * y](l) \quad = C \int_{-\infty}^{+\infty} x(t)y(l-t)dt \quad = \frac{1}{n_o} \sum_{k=1}^{n_o} x_k y_{l-k}, \tag{12.3}$$

where square brackets, [ ], indicate the order of the operators and round brackets, (), indicate the arguments of the input and output functions. Without loss of generality we consider here functions with zero mean value. Note that because of the finite and infinite correlation length of stochastic and periodic series, respectively, no unique normalization C applies in the continuous case.

The discrete operators $\mathcal{F}^{-1}$ and $*$ are well defined only for observations and frequencies which are spaced evenly by δt and $\delta\nu = 1/\Delta t$, respectively, and span ranges Δt and $\Delta\nu = 1/\delta t$. Then and only then $\mathcal{F}^{\pm}1$ reduces to orthogonal matrices. It follows directly from Eq. (12.2) that we implicitly assume that the observations and their transforms are periodic with the periods Δt and $\Delta\nu$, respectively. The assumption is of consequence only for data strings which are short compared to the investigated periods or coherence lengths or for a sampling which is coarse compared to these two quantities. Such situations should be avoided also in the general case of unevenly sampled observations.

The following properties of $\mathcal{F}$ and $*$ are noteworthy:

$$\mathcal{F}[x+y] = \mathcal{F}[x] + \mathcal{F}[y] \tag{12.4}$$

$$\mathcal{F}[x*y] = \mathcal{F}[x]\mathcal{F}[y] \tag{12.5}$$

$$\mathcal{F}e^{-2\pi i\nu_o t} = \delta_{\nu_o}(\nu) \tag{12.6}$$

where δ_x denotes the Dirac symbol: $\int \delta_x f(y)dy = f(x)$. In the discrete case, δ_x assumes the value n_o for x and 0 elsewhere.

12.3.2 The power spectrum and covariance statistics

Let us define power spectrum, covariance and autocovariance statistics P, Cov and ACF:

$$P[x](\nu) = |\mathcal{F}x|^2 \tag{12.7}$$

$$Cov[x,y](l) = x(t) * y(-t) \tag{12.8}$$

$$ACF[x](l) = Cov[x,x](l) \tag{12.9}$$

The power spectrum is special among the periodograms in that it is the square of a linear operator and reveals the important correspondence between frequency and time domain analyses:

$$P[x](\nu) = \mathcal{F}[ACF[x](l)](\nu), \tag{12.10}$$

by virtue of Eq. (12.5).

Let us consider which linear operators or matrices convert series of independent random variables into series of independent variables. For the discrete, evenly sampled observations the ACF is computed as the scalar product of vectors obtained by circularly permutating the data of the series. For a series of independent random variables, e.g. white noise, the vectors are orthogonal. It is known from linear algebra that only orthogonal matrices preserve orthogonality. So, only in the special case of evenly spaced discrete observations and frequencies (Sect. 12.3.1) are $\mathcal{F}[x]$ (and $P[x]$) independent for each frequency. In the next subsection we discuss the case of dependent and correlated values of $P[x]$.

12.3.3 Sampling patterns

The effect of a certain sampling pattern in the frequency analysis is particularly transparent for the power spectrum. Let s be the sampling function taking on the value 1 at the (unevenly spaced) times of the observations observation and 0 elsewhere. The power spectrum of the sampling function

$$W(\nu) \quad = \quad |\mathcal{F}s|^2 \tag{12.11}$$

is an ordinary, non-random function called the spectral window function. The discrete observations are the product of s and the model function f: $x = sf$ so that their transform is a convolution of transforms: $\mathcal{F}x = [\mathcal{F}s] * [\mathcal{F}f] \equiv S * F$, where $S \equiv \mathcal{F}s$ and $F = \mathcal{F}f$. For $f = A\cos 2\pi\lambda t \equiv A(e^{+2\pi\lambda t} + e^{-2\pi\lambda t})/2$ and $F = \mathcal{F}f = A(\delta_{+\nu} + \delta_{-\nu})/2$ we obtain the result $\mathcal{F}x = A(S(\nu - \lambda) + S(\nu + \lambda))/2$. Because of the linearity of $\mathcal{F}$ our result extends to any combination of frequencies. Taking the square modulus of the result equation, we obtain both squared and mixed terms. The mixed terms $S(\nu + \lambda_k)S(\nu + \lambda_j)$ correspond to an interference of frequencies λ_k and λ_j differing by either sign or absolute value. Therefore, *if interference between frequencies is small, the power spectrum reduces to the sum of the window functions shifted in frequency:*

$$P(\nu) \quad \approx \quad \sum |[\mathcal{F}s](\nu + \lambda_k)|^2 \equiv \sum W(\nu + \lambda_k) \tag{12.12}$$

In the opposite case of strong interference, ghost patterns may arise in the power spectrum due to interference of window function patterns belonging to positive as well as negative frequencies. The ghost patterns produced at frequencies nearby or far from the true frequency are called *aliases* and *power leaks*, respectively.

12.4 MIDAS utilities for time series analysis

In this section we describe the functioning of the MIDAS time series analysis (TSA) package. For a detailed description of syntax and usage we refer the reader to the respective `HELP` information. We precede the command description with the description of the scope of the applications and the structure of the TSA input and output files and of the keywords holding the relevant parameters.

12.4.1 Scope of applications

Our package is well suited to the analysis of small to modest sized data sets, with no regard to the sampling which may be even or uneven. Thus our package suits astronomers who often have to deal with unevenly sampled observations well. One of the advantages of the package is the availability of tools for a statistical evaluation of the results.

Data sets containing many observations but covering only few cycles and/or characteristic time intervals can be reduced in number by averaging or decimation, usually with little loss of information. However, the analysis of very extensive datasets, which cover many cycles, contain, say, over 10^5 observations and/or are sampled evenly, is more demanding in terms of computing efficiency than in the choice of the method. With the present package, MIDAS offers an excellent general purpose environment and a variety of tools for the analysis of astronomical data at the price of some computing overheads. Very large data sets usually concern important problems and therefore deserve extra attention in the analysis. For such cases any extra overhead is undesirable, whereas extra efficiency can be gained from specialized algorithms implemented as purpose-built stand-alone codes. One class of such specialized algorithms not covered here is based upon the fast Fourier transform technique (see e.g. Bloomfield, 1976, Press and Rybicki, 1991).

12.4.2 The TSA environment

Before any of the commands of the TSA package can be used, the package must be enabled by `SET/CONTEXT TSA`.

Internally, the functioning of the MIDAS TSA commands depends on a number of parameters whose values are stored in a number of keywords. The keywords specify the time lag or the frequency grid used in the results (`START`, `STEP` and `NSTEPS`), parameters of statistics (`ORDER`) and names of I/O files. These keywords are created and given their initial default value (where applicable) by the command `SET/CONTEXT TSA`. The current values of all TSA-specific keywords can be inspected at any time with the `SHOW/TSA` command. The value of any keyword can be changed either by entering the desired parameter value together with any of the TSA commands using this keyword and by the commands `SET/TSA`, `WRITE/KEYWORD` and `COMPUTE/KEYWORD`. If on a command line a parameter is omitted, its value will by default be taken from the respective keyword.

Execution of the command `DELAY/TSA` for certain options requires the source code of a user-defined ACF function to be available in the local directory. The function name and calling sequence must follow the conventions layed down in the `HELP` information for the command `DELAY/TSA`.

12.4.3 Input data format

The input to all basic routines is a MIDAS table containing at least two columns with the mandatory labels `:TIME` and `:VALUE`. For analysis in the time domain a third column, `:VAR`, containing the variances of the column `:VALUE` is required. Since `DOUBLE PRECISION` computations are used throughout the package, all input columns must be also in `DOUBLE PRECISION`. In any case, it is prudent to chop off (by subtraction of a suitable constant) from the input `:TIME` column all leading digits which are insignificant for the present TSA purposes. Also for numerical reasons, it is recommended to use the command `NORMALIZE/TSA` to subtract the mean from `:VALUE` prior to the analysis.

12.4.4 Output data format

Output periodograms are computed at constant frequency steps and stored in the first row of an image. The second row of this image contains some information on the quality of the periodogram, specific for each method. Both can be conveniently plotted with the `PLOT/ROW` command. `WIDTH/TSA` is available for a first analysis.

Output time lag functions, resulting from the analysis of aperiodic signals, are stored in a table containing the columns `:LAG` and `:FUNCT`. They can be inspected by with standard MIDAS table commands such as `PLOT/TABLE` or `READ/TABLE`.

12.4.5 Fourier analysis

Because of its universal relevance, it is recommended to always start the analysis by computing the power spectrum. However, since the power spectrum is not the optimal method for quite a number of TSA applications, other methods should always be tried thereafter.

`POWER/TSA` – **Power spectrum:** This command computes the discrete power spectrum for unevenly sampled data a by relatively slow method. The discrete Fourier power spectrum (cf., e.g., Deeming, 1975) corresponds to a pure sinewave model and has basic significance in time series analysis. The corresponding test statistic is $S(\nu) = |\mathcal{F}X^{(m)}|^2$. Because the statistic is the sum of the squares of two generally correlated variables for sine and cosine, it has has no known statistical properties. Therefore, we recommend to use other statistics for a more reliable signal detection and evaluation.

One of the important applications of the power spectrum analysis is the computation of the window function in order to evaluate the sampling pattern. For this particular application, set all data values in column `:VALUE` to 1 (for instance by using `COMPUTE/TABLE`) and then apply `POWER/TSA`. The resulting power spectrum is the window function of the data set.

12.4.6 Time series analysis in the frequency domain

For the detection of smooth signals, e.g. sinusoids, use either `ORT/TSA with 'order'=1 or 2 harmonics, SCARGLE/TSA` or `AOV/TSA` with 'order'=3 or 4 bins. The sensitivity of these statistics to sharp signals (such as strongly pulsed variations or light curves of very wide eclipsing binaries) is poor. For the detection of such signals better use `ORT/TSA or AOV/TSA` with the width of these features matched by the width of the of the top harmonics or the width of a phase bin, respectively.

The command `SINEFIT/TSA` serves two purposes: a) least squares estimation of the parameters of a detected signal and b) filtering the data for a given frequency (so-called prewhitening). The trend removal (zero frequency) constitutes a special case of this filtering. For a pure sinusoid model, the χ^2 statistic used in `SINEFIT/TSA` is related to that used in `SCARGLE/TSA` (Lomb, 1976, Scargle, 1982).

`ORT/TSA` – **Multiharmonic analysis of variance periodogram:** The command computes the analysis of variance (AOV) periodogram for fitting data with a (multiharmonic) Fourier series. The fit of the Fourier series is done by a new efficient algorithm, employing projection onto orthogonal trigonometric polynomials. The results of the fit are evaluated using the AOV statistics, a powerful method newly adapted for the time series analysis (Schwarzenberg-Czerny, 1996, 1989). The model used in this method is the Fourier series of n harmonics. The resolution of the method may be tuned by change of n. Hence it is the method of choice for both smooth and sharp signals. The AOV statistic is the ratio $S(\nu) = Var_m/Var_r$. The distribution of S for white noise (H_o hypothesis) and $n \equiv$ `order` bins is the Fisher-Snedecor distribution $F(2n+1, n_o - 2n - 1)$. The expected value of the AOV statistics for pure noise is 1 for uncorrelated observations and n_{corr} for observations correlated in groups of size n_{corr}.

`SCARGLE/TSA` – **Scargle sine model:** This command computes Scargle's (1982) periodogram for unevenly spaced observations x. The Scargle statistic uses a pure sine model and is a special case of the power spectrum statistic normalized to the variance of the raw data,
$|\mathcal{F}X^{(m)}|^2/Var[X^{(o)}]$. The phase origins of the sinusoids are for each frequency chosen in such a way that the sine and cosine components of $\mathcal{F}X$ become independent. Hence for white noise (H_o hypothesis) S is the ratio of $\chi^2(2)$ and $\chi^2(n_o)$. For large numbers of observations n_o, numerator and denominator become uncorrelated so that S has a Fisher-Snedecor distribution approaching an exponential distribution in the asymptotic limit: $F(2, n_o - 1) \rightarrow \chi^2(2)/2 = e^{-S}$ for $n \rightarrow \infty$.

We recommend this statistic for larger data sets and for the detection of smooth, nearly sinusoidal signals, since then its test power is large and the statistical properties are known. In particular the expected value is 1. For observations correlated in groups of size n_{corr}, divide the value of the Scargle statistics by n_{corr} (Sect. 12.2.4). The slow algorithm implemented here is suitable for modest numbers of observations. For a faster, FFT based version see Press and Rybicki (1991).

`SINEFIT/TSA` – **Least-squares sinewave fitting:** This command fits sine (Fourier) series by nonlinear least squares iterations with simultaneous correction of the frequency. Its main applications are the evaluation of the significance of a detection, parameter estimation, and massaging of data. The values fitted for frequency and Fourier coefficients are displayed on the terminal. For observations correlated in groups of size n_{corr} multiply the errors by $\sqrt{n_{corr}}$ (Sect. 12.2.4). With the latter correction and for purely sinusoidal variations `SINEFIT/TSA` computes the frequency with an accuracy comparable to the one of the power spectrum (Lomb, 1976, Schwarzenberg-Czerny, 1991). Additionally, the command displays the parameters of the fitted base sinusoid, i.e. of the first Fourier term.

`SINEFIT/TSA` returns also the table of the residuals $X^{(r)}$ (i.e. of the observations with the fitted oscillation subtracted) in a format suitable for further analysis by any method supported by the TSA package. In this way, the command can be used to perform a CLEAN-like analysis manually by removing individual oscillations one by one in the time domain (see Roberts *et al.*, 1987, Gray & Desikhary, 1973). Since in most astronomical time series the number of different sinusoids present is quite small, we recommend this manual procedure rather than its automated implementation in frequency space by the CLEAN algorithm.

Alternatively, the command can be used to remove a trend from data. In order to use `SINEFIT/TSA` for a fixed frequency, specify one iteration only. The corresponding value of χ^2 may in principle be recovered from the standard deviation $\sigma_o = \sqrt{\chi^2(df)/df}$, where $df = n_{obs} - n_{parm}$ and n_{obs} and n_{parm} are the number of observations and the number of Fourier coefficients (including the mean value), respectively. However, the computation of the χ^2 periodogram with `SINEFIT/TSA` is very cumbersome while the results should correspond exactly to the Scargle periodogram (Scargle, 1982, Lomb, 1976).

`AOV/TSA` – **Analysis of variance for phase bins:** The command computes the analysis of variance (AOV) periodogram for phase folded and binned data. The AOV statistics is a new and powerful method especially suitable for the detection of nonsinusoidal signals (Schwarzenberg-Czerny, 1989). It uses the step function model, i.e. phase binning. Its statistic is $S(\nu) = Var_m/Var_r$. The distribution of S for white noise (H_o hypothesis) and $n \equiv$ `order` bins is the Fisher-Snedecor distribution $F(n-1, n_o - n)$, where n is number of bins. The expected value of the AOV statistics for pure noise is 1 for uncorrelated observations and n_{corr} for observations correlated in groups of size n_{corr}.

Among all statistics named in this chapter, AOV used by ORT/TSA and AOV/TSA is the only one with exactly known statistical properties even for small samples. On large samples, AOV is not less sensitive than other statistics using phase binning, i.e. the step function model: χ^2, Whittaker & Roberts and PDM. Therefore we recommend the ORT/TSA and AOV/TSA commands for samples of all sizes and particularly for signals with narrow sharp features (pulses, eclipses). If on the average n_{corr} consecutive observations are correlated, divide the value of the periodogram by n_{corr} and use the $F(n-1, n_o/n_{corr} - n)$ distribution (Sect. 12.2.4). For smooth light curves use low `order`, e.g. 4 or 3, for optimal sensitivity. For numerous observations and sharp light curves use phase bins of width comparable to that of the narrow features (e.g. pulses, eclipses). Note that phase coverage and consequently quality of the statistics near 0 frequency are notoriously poor for most observations.

12.4.7 Analysis in the time domain

The commands `COVAR/TSA` serves for the calculation of the covariance and autocovariance functions. Pairs of signals with matching ACF functions may be analysed further with `DELAY/TSA`. Matching ACF functions may be obtained for some data after some massaging.

`COVAR/TSA` – **Covariance analysis:** This command computes the discrete covariance function for unevenly sampled data. Edelson and Krolik's (1988) method is used for the estimation of the cross correlation function (CCF) of unevenly sampled series. The binned covariance function is returned with its gaussian errors. Significant are the portions of the curve differing from 0 by more than a number of standard deviations.

This command can also be used for the calculation of the autocovariance function (ACF) by simply using the same series for the two input data sets. Here one shifted series is used as a model for the other. The covariance statistic is used to evaluate the consistency of the two series.

The covariance statistics is akin to the power spectrum statistics and hence to the χ^2 statistics (Sect. 12.3.2, Lomb, 1976, Scargle, 1982). The number of degrees of freedom varies among time lag bins. Thus, in order to facilitate the evaluation of the results, errors of the ACF are returned. The expected value of the ACF for pure noise is zero. The value returned for 0 lag corresponds to the correlation of nearby but not identical observations. This is so because the correlation of any observation with itself is ignored in the present algorithm, for numerical reasons. The correlation function for a lag identical to zero can be easily computed as the signal variance.

`DELAY/TSA` – χ^2 **delay analysis with interpolation:** The command computes the χ^2 time lag function for two time series by the Press et al. (1992) method. One series is used as a model for the other one, and the χ^2 statistics is used to evaluate the consistency of the two series. `DELAY/TSA` differs from `COVAR/TSA` in that each series is interpolated to the times of observation in the respective other series. The interpolation is carried out in an elaborate way by using the common autocorrelation function (ACF) of the series. The average value is computed and subtracted from the series so that the resulting χ^2 is uncorrelated with the average value. This feature of the model enables application to non-stationary series where a mean value is not defined. Because of the interpolation, no coarse binning of the lags is required. Minima of the χ^2 at a given lag and at a level acceptable for the corresponding number of degrees of freedom indicate a physically significant correlation between the two time series via that lag. The corresponding number of degrees of freedom n_r is the number of observations minus the number of fitted parameters (usually 2).

For input, individual measurements must be given with their variances. `DELAY/TSA` requires the smoothed ACF, common for the two series, to be supplied by the user in analytical form. The form of the ACF can be determined using `COVAR/TSA` and the MIDAS FIT package (Vol. A, Chapter 8). For this purpose, the ACF of both series should be the same. Often this can be achieved after some massaging of the data. To broaden the ACF, pass the series through a low pass filter. `NORMALIZE/TSA` may be used to normalize the variances and thus to normalize the ACF maxima. The ACF is passed to the command either via values of the parameters of one of the functions predefined within the TSA package or as the source code of a user-supplied FORTRAN function.

The method is quite new; it should be applied with some caution. Its only presently known practical test has been a consistency check of the results of independent analyses of optical and radio light curves of a pair of gravitationally lensed quasar images (Press *et al.*, 1992). Not only shapes but also values of the ACF should match. This may be achieved by scaling the variances of the observations with `NORMALIZE/TSA`.

12.4.8 Auxiliary utilities

The following commands implement auxiliary utilities for time series analysis:

`BAND/TSA` – **Frequency grid:** The frequency grid suitable for the analysis of evenly sampled observations is well determined. However, for uneven sampling no simple rules exist in general. `BAND/TSA` may be used to find a reasonable guess for the frequency grid. The results are returned in the keywords START, STEP and NSTEPS. `BAND/TSA` may err, as usual in guessing; its results must be checked for consistency.

`COVAR/TSA` – **Correlation length:** The statistical evaluation of TSA results rests on the assumption that the noise in the data is white noise. However, quite often this assumption is wrong. One way to test its justification is to compute the residuals from the model fit (e.g. by using `SINEFIT/TSA`) and to examine the correlation length in the residuals from the autocorrelation function (ACF; computed with `COVAR/TSA`). The average number of observations per correlation length is the average number of correlated observations n_{corr}. For white noise this number should be of order 1.

`NORMALIZE/TSA` – **Normalize mean & variance:** Normalize mean and (optionally) variance of a column to 0 and 1, respectively. Subtraction of the average value from the data is always recommended for numerical reasons. Certain commands will not work correctly for large mean values. Normalization of the variances to the same unit value is required for `DELAY/TSA`.

`SINEFIT/TSA` – **Filtering of the series:** This versatile command may be used not only for parameter estimation but also for data massaging. `SINEFIT/TSA` may also be used to remove a trend from data (high pass filtering). For this purpose choose a low value for frequency, so that only few cycles cover the time interval spanned by the observations. Specify just 1 iteration, so that the routine does not attempt a frequency correction. Subtraction of such data from the raw data returns the fitted trend (low pass filtering). The results are in a format suitable for renewed input to the other frequency domain TSA commands.

`WIDTH/TSA` – **Width of spectral lines:** Profiles of spectral lines appearing in test statistics may be used to refine signal parameters and their errors. In its present primitive form, this command analyses the strongest maximum (an 'emission' line) in the specified sub-spectrum. The window should not be too narrow as it is also used for the determination of the continuum level. A comparison of the continuum level with the expected value of the statistic for a pure noise signal may reveal interference and/or noise correlation. The width of the line may be used to estimate confidence interval of the frequency associated with the line. However, note that the width at half intensity is generally not a good estimate. For Scargle's statistic and power spectra use the width at the level corresponding to the peak level minus the average noise level $P - N$ (Schwarzenberg-Czerny, 1991). For this purpose, this command returns a small table of widths on the screen and in keyword `OUTPUTR`. In power spectra, the peak height of the line is a good measure of the square of the amplitude of the oscillation. – An 'absorption' line can be converted into a 'emission' line by simply changing the sign with `COMPUTE/TABLE`.

12.5 Command summary

`AOV/TSA intab outima start step nsteps [order] [cover]`
Compute analysis of variance periodogram (Sect. 12.4.6).

`BAND/TSA intab [maxobs]`
Evaluate frequency band for time series analysis (Sect. 12.4.8).

`COVAR/TSA intab1 intab2 outtab start step nsteps [scale]`
Compute discrete covariance function for unevenly sampled data (Sect. 12.4.7).

`DELAY/TSA intab1 intab2 outtab start step nsteps [func,mode] [parm]`
Compute χ^2 - time lag function (Sect. 12.4.7).

`NORMALIZE/TSA intab1 outtab column [mode]`
Normalize mean and (optionally) variance to 0 and 1, respectively (Sect. 12.4.8).

`ORT/TSA intab outima start step nsteps [order]`
Compute multiharmonic analysis of variance periodogram (Sect. 12.4.6).

`POWER/TSA intab outima start step nsteps`
Compute discrete power spectrum for uneven sampling by slow method (Sect. 12.4.6).

`SCARGLE/TSA intab outima start step nsteps`
Compute Scargle periodogram for unevenly spaced observations (Sect. 12.4.6).

`SET/CONTEXT TSA`
Enable TSA context. Only after this command has been executed become the comands of the TSA package available.

`SET/TSA` *keywordname=value*
Set global keywords for TSA context (Sect. 12.4.2).

`SINEFIT/TSA intab outtab freque order iter`
Fit sine (Fourier) series, return residuals (Sect. 12.4.6).

`SHOW/TSA`
Show contents of global keywords used within TSA context (Sect. 12.4.2).

`WIDTH/TSA inima [width] [centre]`
Evaluate spectral line width and profile (Sect. 12.4.8).

12.6 Examples

12.6.1 Period analysis

Let us assume that table `OBSERVP.tbl` contains observations of a periodic phenomenon. The observations are stored in the `DOUBLE PRECISION` columns `:TIME` and `:VALUE`.

```
CREATE/GRAPHICS                      ! Create graphics window
SET/CONTEXT TSA                      ! Enable TSA package
NORMALIZE/TSA OBSERVP :VALUE         ! Subtract mean
BAND/TSA OBSERVP                     ! Find suitable frequency band
SHOW/TSA                             ! Inspect corresponding settings
POWER/TSA OBSERVP POWERSPEC          ! Compute power spectrum and
                                     ! spectral window
PLOT/ROW POWERSPEC                   ! Display power spectrum
PLOT/ROW POWERSPEC 2                 ! Display spectral window
ORT/TSA ? AOVSPEC ? ?  ?             ! Compute multiharmonic spectrum
PLOT/ROW AOVSPEC                     ! Inspect results
POWER/TSA ? LINESPEC 20 .01 501      ! Compute detail of power spec.
PLOT/ROW LINESPEC
! Plot it
WIDTH/TSA LINESPEC                   ! Find parameters of
                                     ! the oscillation
SINEFIT/TSA OBSERVP CLNOBS 23 1      ! Remove one particular
                                     ! Fourier component
POWER/TSA CLNOBS POWCLN              ! Inspect periodogram of data
                                     ! after removal of 1 oscillation
```

12.6.2 Comparison of two stochastic processes

Let the two tables `OBSERVA.tbl` and `OBSERVB.tbl` contain two sets of observations. Each set is stored in the `DOUBLE PRECISION` columns `:TIME`, `:VALUE` and `:VAR` containing the times of observation, data value and their variances.

```
CREATE/GRAPHICS                      ! Create graphics window
SET/CONTEXT TSA                      ! Enable TSA package
NORMALIZE/TSA OBSERVA :VALUE V       ! Normalize variance in both light
NORMALIZE/TSA OBSERVB :VALUE V       ! curves to the same value of 1
COVAR/TSA OBSERVA OBSERVA AUTOCOVA 1.  0.1 24 LOG
                                     Compute autocov. of 'A'
PLOT/TAB AUTOCOVA :LAG :COVAR        ! Plot autocov. function of 'A'
COVAR/TSA OBSERVB OBSERVB AUTOCOVB ? ?  ?  LOG
                                     Compute autocov. of 'B'
PLOT/TAB AUTOCOVB :LAG :COVAR        ! Plot autocov. function of 'B'
COVAR/TSA OBSERVA OBSERVB CROSSCOV ? ?  ?  LOG
                                     Compute crosscov. of 'A' and 'B'
PLOT/TAB CROSSCOV :LAG :COVAR        ! Plot crosscovariance function

! Now you have to fit a common analytic formula to both autocor-
! relation functions, AUTOCOVA and AUTOCOVBB. The MIDAS FIT package
! or any other suitable tool may be used for this purpose.
```

```
! Choose one of the predefined function forms or code your own
! function UR_i, 0 < i < 10, in FORTRAN. Then, the analysis
! of the delay can proceed:

DELAY/TSA OBSERVA OBSERVB CHI2LAG 0 5 200 EXP 0,1,-0.25
                                      ! Do Chi2-time lag analysis
PLOT/TAB CHI2LAG :LAG :CHI2           ! Plot the results
```

References

Abramovitz, M. & Stegun, I.A.: 1972, *Handbook of Mathematical Functions*, Dover, New York.

Acton, F.S.: 1970, *Numerical Methods that Work*, Harper & Row, New York.

Bloomfield, P.: 1976, *Fourier Analysis of Time Series: An Introduction*, Wiley, New York.

Brandt, S.: 1970, *Statistical and computational methods in data analysis*, North Holland, Amsterdam.

Chatfield, C.: 1985, *The Analysis of Time Series: An Introduction*, Chapman & Hall, London.

Deeming, T.J.: 1975, *Astron. Astrophys. Suppl.* **36**, 137.

Dvoretsky, M.M.: 1983, Mon. Not. R. astr. Soc. **203**, 917.

Eadie, W.T., Drijard, D., James, F.E., Roos, M. & Sadoulet, B.: 1971, *Statistical methods in experimental physics*, NorthHolland, Amsterdam.

Edelson, R.A. & Krolik J.H.: 1988, *Astrophys. J.* **333**, 646.

Gray, D.F. & Desikachary, K.: 1973, *Astrophys. J.* **181**, 523.

Lafler, J. & Kinman, T.D.: 1965, *Astrophys. J. Suppl.* **11**, 216.

Lomb, N.R.: 1976, *Astrophys. Space Sci.* **39**, 447.

MIDAS Users Guide: 1992 November, European Southern Observatory, Garching.

Press, W.H., Flannery, B.P, Teukolsky, S.A. & Vetterling, W.T.: 1986, *Numerical Recipes*, Cambridge University Press, Cambridge.

Press, W.H. & Rybicki, G.B.: 1989, *Astrophys. J.* **338**, 277.

Press, W.H. et al.: 1992, *Astrophys. J.* **385**, 404.

Renson, P.: 1978, *Astron. Astroph.* **63**, 125.

Roberts, D.H. et al.: 1987, *Astron. J.* **93**, 968.

Scargle, J.H.: 1982, *Astrophys. J.* **263**, 835.

Schwarzenberg-Czerny, A.: 1989, *Mon. Not. R. astr. Soc.* **241**, 153.

Schwarzenberg-Czerny, A.: 1991, *Mon. Not. R. astr. Soc.* **253**, 198.

Schwarzenberg-Czerny, A.: 1996, *Astrophys. J.* **460**, L107.

Stellingwerf, R.F.: 1978, *Astrophys. J.* **224**, 953.

Chapter 13

PEPSYS general photometry package

This chapter describes the PEPSYS package for general photometric reductions. The function of photometric reductions is to measure and remove, so far as possible, the instrumental signature from the data. If we regard the Earth's atmosphere as a part of the instrument not under the control of the observer, we see that measuring and correcting for extinction is an important part of this process. PEPSYS performs extinction correction and transformation to a standard system, if possible.

The package contains two main parts: a planning program and a reduction program. The planning program interacts with the user to find out the goals of the observing program, and then produces a schedule of the observations needed to meet those goals efficiently. The reduction program is flexible enough to model most types of photometric observations accurately, and produce reliable estimates of the uncertainties of the parameter values it obtains from the raw data.

In addition, some auxiliary tools are provided to simplify the work of making the required tables.

Because the data reduction must model the instrument and observational procedure as accurately as possible, we must discuss photometric techniques. The recent book by Sterken and Manfroid [22] and an earlier review [10] are good general references. We also discuss the modelling process in some detail, to provide users with some assurance that good models are used.

Please read through the documentation carefully before trying to use the programs! Many problems can be avoided if you are thoroughly familiar with the contents of this chapter.

13.1 Introduction

Good photometry requires careful attention to calibrations. If there are no calibration observations of standard and extinction stars, the program-star observations cannot be transformed to a standard system. At the opposite extreme, if there are *only* observations of standard and extinction stars, there is no time for program-star observations. In either case, the data produce no useful information. Obviously, somewhere in between these extremes lies an optimum distribution of program and standard stars.

The choices of which standards to observe, and when to observe them, involve not only balancing the gain in information by adding a calibration observation against the loss of a program-star observation, but also assumptions about the stability and linearity of the equipment used. A good distribution of standards in each color index is also needed to determine accurate transformations. When all these details had to be attended to by hand, it was easy for observers to neglect some essential item. Much experience was required to distribute observations effectively.

Fortunately, the computer can be told to keep track of all these details, and will not forget them. The planning program helps you approach the optimal distribution of standards, which gives the most accurate and precise results possible for a given amount of observing time.

13.1.1 What is needed

A basic principle of photometry goes back to Steinheil, who built the world's first stellar photometer in the early 1830's. Steinheil's principle is that only *similar* things should be compared. That is, good results are obtained when as many variables as possible are kept fixed. This minimizes the number of instrumental parameters that have to be calibrated, and allows the necessary calibrations to be a small part of the total data gathered. All observations must be made with the same instrument, used under the same conditions.

However, we cannot make all our observations at the same time, or at the same place in the sky. So we must be able to separate different effects that can vary with time, such as extinction coefficients and instrumental zero points. To do this, we must distribute the observations so that the main independent variables (such as airmass, star color, and instrumental temperature) are uncorrelated with each other. In particular, they should all be uncorrelated with time, so far as possible. As some (like temperature) are inherently likely to be correlated with time, it is doubly important that others (like airmass) be uncorrelated. Achieving this condition requires a certain amount of advance planning.

13.1.2 How to get it

To help you get the necessary calibration data in the least observing time, the planning program generates a schedule of required observations. A considerable amount of photometric expertise is built into this program, so that observers without much photometric experience can confidently adopt its recommended defaults, and obtain a plan that should be satisfactory for most photometric observing programs. At the same time, the program is flexible enough to let experienced observers design special-purpose programs.

This flexibility requires considerable interaction between the user and the planning program. The program needs information about the program stars, the calibration stars, the peculiarities of the instrument, and the requirements of the observer. This input information is described in detail below, and in the appendix on data-file formats.

13.1.3 What to do with it

Once the observations are made, we must remove the instrumental signature, including atmospheric effects, as fully as possible. To use the observational data effectively, the reduction program must make correct assumptions about the way the data were gathered, and the way the instrument itself (including the atmosphere!) really works.

The atmospheric part can be removed quite accurately, but some remaining instrumental effects, due to a mismatch between the instrumental and standard passbands, cannot be completely removed, because some information is missing (especially in the conventional photometric systems). Nevertheless, if the filters are quite close to the standard ones, the missing information is fairly small, and good results are possible.

Again, interaction between the observer and the program is necessary, both to obtain the necessary information and to decide how to proceed when choices are not clear-cut. Safe defaults are offered to the beginner; more experienced observers can try various assumptions to see what works best for their purposes. In both programs, the goal is to allow a wide variety of choices, but to warn users of potential problems.

13.2 Getting started

The first step is to gather the information that will be needed to plan a program and to reduce the data. Information that is more or less permanent is stored in several MIDAS table files:

- Star tables: These give names and positions of stars to be used for calibrations, as well as program stars. Common supplementary information, such as spectral types, may be included. Standard-star tables are provided for some popular systems, but you will have to compile your own table of program stars.
- Observatory table: This table contains the positions of various telescopes, together with essential information such as apertures, and subsidiary information.
- Horizon table: This table describes the apparent horizon as seen from a particular telescope.

All of the above information, except for program-star tables, should already be available to you. Only if you are the first user of PEPSYS at an observatory will you need to construct the Observatory and Horizon tables yourself. They are described in the Appendix; see also section 13.6.1 below. The Observatory file is very simple, and data for the Horizon file for a telescope can be collected in a few hours (the command `MAKE/HORFORM` will help you compile the horizon data). Program-star tables are described below, along with the `MAKE/STARTABLE` command provided to produce them.

In addition, you will need information about the instrument. As instruments tend to evolve with time, you will probably need to put this information together for each observing run separately. The `MAKE/PHOTOMETER` command will ask you questions, and make a new file from your answers; it can also show you the contents of an existing file.

If an instrument is fairly stable, and a previous instrument-configuration file is available, you may be able to just edit the old file, using the `EDIT/TABLE` command.

Some of the instrumental information may not be available at the time you plan your observing run. You can give "don't know" responses to questions about such things, or leave items blank if you have no information. However, for the sake of accuracy, you should try to obtain as much of the missing information as you can before reducing the observations.

13.2.1 Star tables

The essential data needed for program stars are names, positions, and the equinox to which the positions are referred. You may also include proper motions and the epoch of the position; spectral types; rough magnitudes, such as might be obtained from the HD; and comments. The command `MAKE/STARTABLE` helps you create a table of program stars.

Some telescopes record the position toward which the control system thinks the telescope is pointing as part of the data stream, so it is tempting to extract these data to make a positional catalog for the program stars. In general, this is very unwise. The telescopes usually used for photometry generally have neither accurate nor precise pointing.

The *precision* (repeatability) of the recorded positions does not indicate their *accuracy* — which could be far worse, if the observer adjusted the zero-points of the telescope's coordinates to agree with mean places for some equinox removed a few years in time from the date of observations. A $2'$ altitude error corresponds to an airmass error of 0.002 at 2 airmasses, more than 0.005 at 3 airmasses, and 0.01 airmass at 4 airmasses. Thus, such errors can cause appreciable systematic errors in extinction determination and correction. Although such recorded positions are available (by means of the CONVERT/PHOT command) as a last resort, they are *not* recommended. Good catalog positions should be used whenever they are available.

Standard-star tables also require columns for the standard values, which are often not known for program stars. Because of this difference in content, you should keep standard and program stars in separate table files. You can have several files of each type. The programs will ask for the standard-star files first, and then the program-star files.

You should also plan to keep extinction stars (and other constant objects, such as comparison stars in variable-star programs) in a *separate* file from variable stars. The reduction program treats standard, constant, and variable stars differently, so each group should be kept in one or more separate files (see subsection 13.5.6, "Reduction procedure").

ASCII source files

If you are lucky, your program stars are available in machine-readable form. If so, simply copy out the data for your program stars as fixed-format ASCII records, one star per record. It doesn't matter if the ASCII file contains extraneous data; they can be ignored in converting to MIDAS format. The command `MAKE/STARTABLE` helps you turn this flat ASCII file into a MIDAS table of program stars. It will first ask for the name you want to give your new program-star table. Then it checks to make sure you have the necessary information, such as the ASCII table of stellar data, and a format file.

It is important to make sure the star names in your source files match those in your data files. In building star files, keep in mind the need to have the *same* designations appear in the observational data. If your data-logging system makes you enter star names when they are observed, try to keep names short to avoid typing errors. That means using short, unique names in your program-star files, to match those that will appear in the data. If your data-logging system takes star names from files prepared in advance, try to adhere to the standard IAU name format (see the guidelines published in A&A "Indexes 1990 and Thesaurus, Supplementary Issue, May, 1991", pp. A11-A13; PASP **102**, 1231-1233, 1990; and elsewhere).

If possible, use at least two names for each star, as recommended by the IAU. You can use two names in the star files, and just use one in data files; the reduction program is smart enough to match them up properly, or will ask for help if similar but not identical names occur. It is a good idea to separate aliases with an "=" sign; just leave a space on either side of it, so the program doesn't take it as part of a name string. Ordinary spacing is allowed in names to make things readable: `HR 8832`, `Chi Cygni`, `BD +4 4048`, etc. Thus a name field might contain "`HD 24587 = CD -24 1945 = HR 1213`".

Although you can use any naming system you like for program stars, so long as the *same* name appears in star files and data files, a system of priorities is suggested for making up catalogs of standard stars. The basic principle is that small catalogs generally yield shorter names that are easier to use than do bigger catalogs. Generally, catalogs for brighter stars contain fewer entries; so the lists with the brightest limiting magnitude are preferred. Thus, common designations like Bayer letters and Flamsteed numbers are usually included for the brightest stars. The bright stars are also listed by HR number, and fainter ones by HD or DM number. Don't forget that there is considerable overlap among the BD, CD, and CPD; specify the catalog, not just a zone and number. The HST Guide Star Catalog is recommended for still fainter objects.

If the reduction program finds stars with different names but nearly identical positions, it will ask you if they are the same star. Be prepared to answer such questions, if you enter names inconsistently. Many users find that repeatedly answering such questions becomes tedious and irritating; you can avoid this problem by using identical name strings in both star files and data files.

Don't use names like `STAR A` and `STAR B`, as the matching algorithm will spot the common word STAR and ask if they are the same; instead, just use the letter. If it is necessary to intermix several similar names, try to make unique strings out of them. For example, if you are working on a group of clusters, and have local standards designated by the same letters in each, attach the letter to the field designation: `M67_A`, `M67_B`, etc.

Although it is not recommended, the CONVERT/PHOT command can extract apparent star positions from raw data files in ESO and Danish formats (see section 13.2.1). As intermediate output, this command will produce an ascii file that can be edited. Manual editing may be necessary if you do not use consistent naming conventions while observing.

Finally, don't try to create star files with incomplete data. Missing values cause problems when you try to reduce data. Be sure every entry is filled in correctly.

Format files

If you have no format file ready, `MAKE/STARTABLE` copies a standard table-format template named **`starfile.fmt`** into your working directory. The ASCII format file describes the input file by giving the column numbers (character positions) of the input file in which each field (table column) begins and ends; it describes the final display formats with Fortran-like format specifications. Note that the widths of the input and display formats do *not* have to match.

You can edit this ASCII format file with any editor. All you need to do is fill in the correct first and last column numbers of the various fields, and delete references to fields you do not need. You may also want to modify some formats; for example, you may have Right Ascensions given only to minutes and tenths instead of minutes and seconds. The appropriate formats are explained in comment lines (beginning with "!") in the dummy format file. After editing the format file, re-invoke `MAKE/STARTABLE`. This time, tell it you have a format file ready; it will use the formatting information to convert your ASCII file to a MIDAS table file.

Manual entry of data

If you have no ASCII table ready, you can enter stars manually from printed tables, using the `EDIT/TABLE` command to add them to the list. This editor is briefly described under "Interactive Editing of Tables" in the MIDAS Users Guide. `EDIT/TABLE` will automatically be invoked by `MAKE/STARTABLE` if you have no ASCII file to convert. If you don't like the EDT-like editor provided by `EDIT/TABLE`, create an ASCII file with whatever editor you like and set up a *.fmt file to convert it, as described above. Be *very* careful to double-check the results, as transposed digits and other errors are almost certain to creep in when entering data by hand!

Multiple star files

Probably you will have several sources of program stars. Each one can be turned into a separate MIDAS table file, as just described; then, if you wish, the tables can be combined into one big table, using the MERGE/TABLE command. Note that the column display formats will be taken from the *first* of the merged tables; this may influence the order in which you do the merging.

If the formats are different, you may want to keep the program-star tables separate. In any case, you should keep standard, constant, and variable stars in separate tables. Then you just enter the names of the different tables one by one, as you run the planning and reduction programs.

A more convenient way to handle multiple files is to make a catalog, and then just supply the catalog name. Only stars of a single type (standard, constant, or variable) should be kept in the same catalog. At present, only the reduction program can use star catalogs.

Standard-star files

Some standard-star files are supplied for the UBVRI and uvby-H-Beta systems. These should be installed in the `$MIDASHOME/$MIDVERS/calib/data/pepsys` directory; see section 13.6.1 if they are not. The files are named `UBVSTD.tbl`, `UVBYST.tbl`, `saaoUBVRI.tbl`, and `saaouvbyHB.tbl`. The latter two are the E- and F-region standards, established by A.W.J.Cousins.

You can also make new tables of standard stars. The easiest way to make standard-star tables is to begin as though they were program-star tables, but to include columns of standard values in the ASCII file, and field descriptions for them in the format file, before running `MAKE/STARTABLE`. The standard column names are tabulated in the "Standard values" subsection of "Star tables" in the Appendix. Don't forget to add the SYSTEM descriptor when you are done!

In making new tables of standard stars, remember that they will be used as extinction stars by the planning program. Therefore, you should look for standards that pass within about 20 degrees of your zenith; this picks a zone of declination centered on your latitude. In right ascension, you want a fairly uniform distribution, so that a standard will be available at large air mass frequently during the night.

The planning program will not select standards that are so near the Sun's right ascension that they can only be observed at large air masses. The instrumental magnitudes of stars that cannot be observed at small air masses must be determined by transforming the standard values to the instrumental system, and these transformed values have relatively large errors; therefore, they contribute relatively little extinction information (see pp. 162-163 of Young [10] for a discussion of this problem, and Beckert and Newberry[1] for examples of transformation errors).

If you must go far from the zenith to pick up standards of rare types, put them in a special file, and treat them as program stars when running the planning program. Then observe them only near the meridian, and don't try to use them as extinction stars. They can still be treated as standards when you run the reduction program.

13.2.2 Observatory table

The standard ESO observatory table, called `esotel.tbl`, is stored in the MIDAS directory system at `$MID_PEPSYS/esotel.tbl` . This directory is *not* distributed with the regular MIDAS updates, but is part of the calib directory that can be reached by anonymous ftp from mc3.hq.eso.org (see the Dec. 1991 issue of the *Courier* for details). Just cd to the midaspub directory, and see the README file there. These table files must be installed on your local system before they can be used; there are procedures for creating them in the calib directory. These and other files in the calib directory subtree can also be obtained on tape by special request.

This is a short table, so if you need to make up a new one, just copy the ESO table and edit the copy with the `EDIT/TABLE` command. (If you do this, don't forget to change the descriptors as well as the table entries!) Or, using the dummy format file `$MIDASHOME/$MIDVERS/contrib/pepsys/lib/esotel.fmt` to convert a *.dat file to MI-

DAS table form, you could create a whole new table. The absolutely essential columns are the designation of the telescope focus, accurate coordinates (including height above sea level), and the diameter of the telescope. Please consult the Appendix to clarify the meaning of the more obscure columns! They are not yet required by PEPSYS, but may be in the future.

13.2.3 Horizon tables

Every telescope needs a horizon table, which tells where the telescope can look in the sky, and also from which parts of the sky the Moon can shine on the objective. Without a horizon table, the programs will assume that the telescope has an unobstructed view of the horizon — an assumption that is probably wrong. The name of the table is keyed to the name of the telescope; see the Appendix.

If you need to make up a new table, the command `MAKE/HORFORM` will provide you with a form to fill in while gathering the data, which can then be entered into the empty table provided.

13.2.4 Instrument file

Instruments are more complicated than star catalogs, but a MIDAS table file is still useful for holding the description of an instrument. The file is partitioned into two sub-tables. The main sub-table contains information about the passbands used, and the second sub-table describes the detector(s). General information about the instrument is stored in descriptors.

The `MAKE/PHOTOMETER` command will solicit the necessary information (if you need to make up a new table), or display an existing table (so you can check its contents). Because the structure of instrument files is rather complicated, you should use `MAKE/PHOTOMETER` to build a new file when an instrument changes, rather than trying to edit the file. This will ensure that all the necessary information is included, and that the file contains an internally consistent description of your instrument.

Basic data for an instrument file include the passbands used (and any coding that is used to represent them in the data); the detectors used, and their properties; and general information about temperature control and optical cleanliness. If your data include neutral-density filters or indicate which measuring aperture was used, be sure to include the corresponding information in this file. See the "File Formats" Appendix for details.

One property that needs special attention in pulse-counting systems is the *type* of dead time. There are (in the textbook approximation) two kinds of counters; these are known as paralyzable and non-paralyzable, and the corresponding dead times are described as extending or non-extending. Different analytical expressions relate observed and true counting rates for the two types, so it is important to know which kind you have. The matter is very clearly discussed by Evans [4], which is a standard reference on this subject.

If an instrument contains filters for more than one system, it may be useful to maintain separate files for the different systems used (e.g., UBV and uvby). This should certainly be done if different filter wheels are used for different systems.

13.2.5 General advice about table files

The MIDAS table file system contains several commands for editing and modifying table files. Columns can be re-named, added, or deleted; various operations can be performed on column contents, and the results stored in a new column; and so on. You should read the on-line help for all the commands with the TABLE qualifier, just to see what operations are supported.

One common problem arises with columns that contain character strings. When you create a table from an ASCII file, the width of the table column is (by default) the width of the field in the original *.dat file. Later, you may want to add information to such a column, but find that the existing table column is too narrow to hold the whole string. For example, you might want to add aliases to the OBJECT names, or expand a COMMENT column.

The column cannot be made wider by the `EDIT/TABLE` command; however, there is a way to widen it. First, use the "concat" operation of `COMPUTE/TABLE` to add a string of blanks to the existing column, and store the widened column under a temporary name. Then delete the original column, and rename the new column with the old name.

You can avoid this problem by specifying a width wider than the original ASCII field by using (say) C*32 instead of just C for the column in the format file, when you create the table. Specifying the width in the format file overrides the default width. However, you may find that this wastes a lot of disk space if the column is mostly blanks.

13.3 Planning your observing run

13.3.1 Introduction

Now suppose you have all the files set up to make MIDAS happy, and you are ready to plan an observing run. The proper distribution of extinction and standard stars was discussed by Young [10]. The program is based on this discussion, but includes other considerations as well (such as the need to measure time-dependent extinction, and to measure instrumental zero-point drifts). It should help you get the right amount of good calibration data with a minimum of observing effort.

The command `MAKE/PLAN` will ask questions about what you are trying to do, and produces a schedule of observations that will meet your goals, if possible. If you are trying to do something impossible, it will tell you so, and offer suggestions. In general, if you follow the suggestions of the planning program, you will get the data you need — assuming that the weather cooperates!

The planning program needs to know which photometric system you want to work in, and will try to supply appropriate standard stars. You can provide other table files of standard stars, if you wish; but be sure they really are on the same system as the built-in standards!

13.3.2 Preparing to use the planning program

The planning program will ask you for a lot of information, so be ready with the answers to its questions. Obviously, it needs to know the location and size of the telescope you will use, and the stars you want to observe; so you must tell it the names of the files in which this information is stored. It also needs to know the accuracy you want to achieve, when you want to observe, and so on. Some questions ask you for choices between plausible alternatives, such as the kind of time you want to use (UT, local zone, or sidereal).

You may need to look up or find out some of the information ahead of time. If you are not using a common photometric system, you will need to supply the central wavelengths and widths of the passbands you are using. Even if you are using a common system, information on your actual instrumental passbands will be helpful, if it is available. If you get halfway through and discover you lack some vital piece of information, just enter Q to quit. Get what you need, and begin again.

Star coordinates are needed to a minute of arc or better for accurate data reduction, so you may as well supply accurate coordinates for planning, too. A few files of standard stars are already available.

If you are doing pulse-counting photometry, the uncertainty in the pulse-overlap correction sets a limit to the brightest stars that can be used as standards. Therefore, you will need to have both the effective dead-time of your system, and a realistic estimate of its uncertainty.

13.3.3 Using the planning program

The planning program asks a series of questions; you provide the answers. Many are simple yes/no choices that can be answered as simply as `y` or `n`. In general, when a choice among several alternatives is required, you can truncate your answer (usually, to just the initial letter) so long as it uniquely specifies the choice. The less typing you do, the less chance you have to make a mistake.

In many cases, the program will offer some guidance, or suggest reasonable default values. You will not go far wrong if you adopt its suggestions. However, experienced observers may have good reasons to "bend the rules" a little at times. The program will allow this, though it may complain if you are doing something it thinks is unwise.

In general, input that is comprehensible to an astronomer is also understood by the program. For example, the first and last dates of your observing run can be entered in any reasonable format, so long as the month is specified either in full or by the usual 3-letter abbreviations. Because European and North American conventions for dates differ in the order of month, day, and year, the program will not accept ambiguous forms like 3/8/95. Because of the possibility of misinterpreting dates, the program will tell you how it has interpreted what you typed in, and give you a chance to correct any misunderstanding.

If you make a mistake when entering the name of a file, the program will give you a chance to recover. The program checks to make sure files exist and appear to be of the proper type. The suffix ".tbl" will be supplied automatically if you do not type it in. If you mistakenly say there is another star file, and the program finds your last filename does not exist, you can enter "`NONE`" when it asks for another name, and it will go on to the next question.

13.3.4 Selection criteria

An overall criterion for standard-star selection is the accuracy needed in the final results: if you need more accuracy, you will need more standard stars. The program asks what accuracy you are trying to reach, and tries to select enough stars to meet your request without being excessive.

To determine the extinction accurately, you must observe some "extinction stars" at both high and low altitudes. While it is possible to get a rough estimate of the extinction by observing *different* standard stars at high and low airmasses, the relatively large conformity errors in most photometric systems, together with the low accuracy of some standards, make this method very inefficient (see Young [10], p.178). The practical problem is that the extinction coefficient can change considerably in the few hours needed for an extinction star to move from low to high airmass (or vice versa). To minimize this problem, the program selects stars that traverse a large range in airmass in the shortest possible time; these are stars that pass near your zenith. Furthermore, it asks you to observe extinction stars that are both rising and setting, to avoid a correlation of airmass with time.

However, the airmass changes slowly when stars are near the zenith. But to separate extinction drift from instrumental drift, you must observe a wide range in air masses in a short period of time. Therefore, the program selects times when the extinction stars cross an almucantar a little removed from the zenith, rather than when they are on the meridian. This almucantar typically corresponds to about 1.1 airmasses. The times of these crossings are denoted by `'EAST'` and `'WEST'` in the output of the planner. To optimize the precision of the extinction determination, the low-altitude observations are placed at about 25° altitude, near 2.36 airmasses. Those scheduled observations are denoted as `'RISING'` and `'SETTING'`.

Furthermore, to track changes in the extinction accurately, you need an extinction measurement (i.e., an observation of an extinction star at large airmass) two or three times per hour. This means that the stars used must be in the right places in the sky to be at large airmasses when you need them. In particular, although the Cousins E-region standards are excellent secondary standards for *transformation* purposes, Southern-Hemisphere observers should augment them with *extinction* stars more evenly distributed on the sky.

Obviously, standard stars used for the transformation from instrumental to standard system can also be used for the transformation from inside to outside the atmosphere (traditionally called "extinction correction"). To minimize the number of calibration observations, the planning program makes standard stars do double duty as extinction stars. These stars should be bright enough that their photon noise is negligible; the proper magnitude range depends on telescope size and the bandwidth of the filter system used. On large telescopes, bright stars are *too* bright, especially if you are doing pulse counting.

Finally, the standard stars must have a good distribution in each of the color indices of the system you are using. Because transformations are generally non-linear ([1], [29]), a wide range of each color should be covered rather uniformly; it is *not* enough to observe a few very red and a few very blue stars. All these requirements impose constraints on the selection of standard stars.

Notice that, in using standard stars to measure extinction, we need not use the standard values transformed to the instrumental system (though this is possible). Instead, we use the actual observed instrumental values for these stars, which are considerably more accurate than standard values transformed to the instrumental system (cf. p.184 of [10]), because of conformity errors [16]. We observe standards at both large and small airmasses, and determine the extinction directly from these observations. This matter is discussed more fully in connection with the reduction program (see below).

13.3.5 Influencing the plan

The program will propose reasonable default choices, and will provide some approximate estimates of the magnitudes where photon noise becomes excessive, and where photon and scintillation noise are comparable. Because the crossover between photon and scintillation noise depends on both zenith distance and azimuth, a set of values will be presented for your inspection. Table 13.1 shows what a typical crossover table looks like.

```
   SCINTILLATION = PHOTON NOISE   Photon Noise   Present
at  secZ = 2.36   secZ = 1.10   of  5.sec.int.   FAINT
      between        between     is0.005 mag.at   limit
 U    9.9 &  8.9    7.1 &  7.0       10.5          5.5
 B   10.6 &  9.7    7.6 &  7.5       11.2          6.0
 V   10.7 &  9.8    7.5 &  7.4       11.3          5.9
```

Table 13.1: A sample crossover table

There's a lot of useful information in this table, so let's go over it carefully. First, notice the similar pairs of columns at the left, under the title `SCINTILLATION = PHOTON NOISE`. The left-hand pair of columns gives crossover magnitudes for an airmass near 2.36; the right-hand pair, for an airmass of 1.10. These are the maximum and minimum airmasses at which the planning program expects to schedule observations of extinction stars. The smaller airmass will be adjusted by the program to provide more or fewer extinction stars, as needed; you can push it back and forth a little if you want.

For each airmass, there is one pair of columns. These contain the magnitudes at which scintillation noise and photon noise are expected to be equal, for two different lines of sight: one looking along the projected wind vector in the upper atmosphere, and the other looking at right angles to the wind. When we look at right angles to the wind, the scintillation shadow pattern moves with the full wind speed across the telescope aperture, and we have the maximum possible averaging of scintillation noise in a given integration time, the least scintillation noise, and thus the brightest possible crossover magnitude. But when we look directly along the wind azimuth, the motion of the shadow pattern is foreshortened by a factor of sec z; then the scintillation noise is maximized (for a given zenith distance), and the crossover does not occur until some fainter magnitude, where the photon noise is big enough to match the increased scintillation.

As we do not know the wind azimuth in advance, we can only say that the scintillation and photon noises will be equal somewhere in the interval between these two extremes. We would generally prefer to have the extinction measurements limited only by scintillation noise, so the initial faint limit for extinction stars is set 1.5 magnitudes brighter than the brightest of the crossover values. This makes the photon noise half as big as scintillation, near the zenith. (Our sample table shows these initial values.) These are conservative values.

However, we may need to set the actual faint limit used in selecting standard and extinction stars (shown in the rightmost column of the table) somewhat fainter. For example, we may be using such a large telescope that we cannot observe such bright stars. (In particular, if you are doing pulse counting, the program will impose a bright limit as well as a faint one.) In this case, both photon and scintillation noise will be quite small, and we can safely use considerably fainter stars without compromising our requested precision.

In the example above, the user has requested an accuracy of 0.01 magnitude. The planning program divides this error budget into four parts: scintillation, photon noise, transformation errors, and instrumental instabilities. If these are uncorrelated, each can have half the size of the allowed total; in our example, that's 0.005 mag. So the table gives the magnitude at which the photon noise reaches its allowed limit (in the next-to-last column), for the adopted integration time (5 seconds, in our example). This magnitude should be regarded as an absolute limit for extinction and standard stars.

Obviously, we could actually push the extinction stars close to this photon-noise limit, without exceeding the requested error tolerance. However, between the photon-noise limit in the right half of the table and the crossover values to the left, there is a substantial contribution of photon noise to the total, and hence a substantial advantage to using brighter stars. If we use this advantage, we provide some "insurance" — a little slack in the error budget.

Whenever the crossover table appears, you will be given an opportunity to change the actual planning limits, whose current values are given in the last column. The columns to its left provide you with the information you need to make a good choice: the crossover magnitudes, and the pure photon limit, for each band.

Although these values are given to 0.1 mag precision, you should be aware that the scintillation noise can fluctuate by a factor of 2 or more within a few minutes, so that only a rough estimate is really possible. Furthermore, the photon-noise estimates are only as good as the estimates available to the program for the transmission of the instrument and the detective quantum efficiency of your detector. So all these numbers are a little "soft"; you should not take that last digit literally. Just bear in mind that the photon noise varies with magnitude, and that the scintillation varies with airmass and azimuth, by the amounts shown in the table.

Now, let's consider adjusting the circumzenithal airmass. If the high-altitude, low-airmass almucantar is too close to the zenith, only a few standard stars will be available in the small zone it intercepts as the diurnal motion carries the stars past it. Then it may be necessary to choose a larger minimum airmass to expand the region of sky available for choosing extinction stars. Conversely, if too many extinction stars are selected, it makes sense to reduce the zone width a little, thereby getting not only a more reasonable number of stars, but also a little bigger range in airmass for each star.

The planning program will make coarse adjustments in the minimum airmass (i.e., in the width of this zone) to get about the right number of extinction and standard stars, but you can also make fine adjustments yourself. The program gives you this option after making a preliminary selection of standards. When it asks whether you want to adjust the sky area or magnitude limits, reply "sky," and it will make a small change and try again. You may need to make several adjustments to get what you want.

If you expect from past experience that you will have excellent photometric conditions, you may be able to reduce the number of extinction stars a little. However, this is risky: if the weather turns against you, you may need more stars than you expected! Conversely, if you know the observing run is at a place or time of year that usually has mediocre conditions, you will surely want to play it safe and add some extra extinction stars.

You can also alter the number of candidate stars by adjusting the magnitude limits. By adjusting magnitude limits separately in different bands, you can manipulate the range of colors available. For example, because signals tend to be low in ultraviolet passbands, the photon noise is high there, so it often happens that the default faint-magnitude limits do not allow enough very red stars. In this case, making (say) the faint U limit fainter and the V limit brighter biases the selection toward redder standards.

Thus, you can manipulate the region of sky and the magnitude limits to select a reasonable number of standards with a good range of colors. You can make such adjustments iteratively until you are satisfied with the selection. At each stage, the program will show you the distribution of the selected stars in a two-color diagram, as well as on the sky, and ask if its selection is satisfactory. If it is not, you reply "no" to the question "Are these stars OK?", and then have another chance to manipulate the zone width and magnitude range. When you finally reply "yes", the program will make up an observing schedule for each night of the observing run, using the set of stars you approved.

Keep in mind the possibility that a star previously certified as a photometric standard can still turn out to be variable! A surprising number of bright eclipsing binaries continue to be discovered every year. Furthermore, stars of extremely early or late spectral type, and stars of high luminosity, tend to be intrinsically variable by small amounts. Therefore, it is important to use a few more standards than would otherwise be absolutely necessary, just in case of trouble. A little redundancy is cheap insurance against potential problems. We also need some redundancy to find data that should be down-weighted (see section 13.5.3 to see why.)

13.4 Getting the observations

When you go to the telescope, follow the plan as closely as you can. If the telescope and instrumentation are unfamiliar to you, you may have trouble keeping to the schedule for the first night or two, so try to work with a previous observer on the equipment for a night or two of training before your observing run. If you fall behind, try to keep as many low-altitude observations of extinction stars as you can, and drop some circumzenithal ones if necessary.

Don't change your mind in the middle of the night! If you think you will need to adopt a different plan in the middle of the run, make sure you generated that alternate plan ahead of time. The time to be creative is during the planning stage, not while observing.

Remember Steinheil's principle: only *similar* data can be compared. That means: DON'T switch to a different filter set or detector in the middle of a run. DON'T use different focal-plane apertures for different stars or different nights; they change the instrument's spectral response as well as its zero-point. DON'T alter temperature, high-voltage, or discriminator settings.

There are very great advantages to getting homogeneous data, because then the determination of the instrumental parameters can be spread over several nights; for example, see [26], [10], [19], [15], and [21]. Likewise, the extinction can be determined much more accurately from observations spread over full nights than from partial nights. If you are only interested in a small region of sky that is not up all night, either use the rest of the night to get extinction and standard-star data, or try to combine your run with another program that uses the same equipment; then all the data from both programs can be reduced together.

If there were instrumental parameters you weren't sure of when you made up the plan, make sure you find out their values during the observing run. The printed schedule will provide blank spaces for you to write them in. Many useful instrumental tests can also be run during the daytime, or during cloudy nights.

Remember that the extinction changes more on long time scales than on short ones; so arrange your observations to chop as much extinction variation as possible out of the most important data. If you need very good $(u - v)$ colors, observe those filters in direct sequence. If you need to determine an eclipsing binary's light curve in several filter bands, you may do better chopping between program and comparison stars before you change filters (see [28] for further discussion).

A very common problem in observing records is an incorrect clock setting. Even a one-second error is worth correcting. An hour off is a common and much more serious mistake. Sometimes tired observers enter the wrong *kind* of time (say, UT instead of LST) into the observing records. It's a good idea to note both times in your observing log — you *do* keep a written log of your observations, don't you? (It's easy to annotate the printed output of the planning program for this purpose.) Even if the computer is supposed to keep track of these things, what happens if there is a power failure, or the computer crashes in the middle of the night?

It doesn't hurt to take note of moonrise and moonset; they can be used to double-check the recorded times for gross errors. Be sure to write down anything that seems suspicious in the data, or possible problems with the equipment. Don't imagine you will remember all the details of what went on when you get around to reducing the data, which might be weeks or months later. For example, which night was it the mirror got snowed on? You can expect a big zero-point shift (or worse!) after that.

If temperature and humidity readings are not recorded automatically, be sure you write them down *at least* once an hour. (Space for this information is provided on the planning schedules.) The temperature that is important is the temperature of the photometer itself, not the observing room! But sometimes you only have a thermometer available on the wall; then record that, as any information is better than none at all. Be sure to record the *time* when each reading was taken. And don't forget to check the dark level now and then; it is a useful check on the health of the instrument.

It is very easy to forget that "nuisance" parameters, like extinction and sky brightness, have effects that propagate directly into errors of the measurements you care about. But this means you must be just as careful in measuring sky as in measuring stars, and as careful with extinction and standard stars as with program stars. Accurate and careful centering of stars is very important, particularly with photometers using opaque (inclined) photocathodes. Machine-like regularity is the ideal, attainable with some (but not all) automatic centering systems.

Likewise, one must always measure the sky the *same distance* from each star, regardless of brightness. Otherwise, you include a different fraction of the star's light in the sky measurement. A common mistake is to offset the telescope until the glow from the star is no longer visible in the measuring aperture; this guarantees that bright stars are measured with a different system than faint ones. A fixed angular offset — with due regard to the telescope's diffraction spikes – should always be used. If possible, always measure the sky in the same *place* for each star, to make sure you include the same unseen background stars in every measurement.

Finally, because sky brightness fluctuates on short time scales, it's best to measure sky for each star. You can use much shorter integrations for sky than for bright stars; but don't let more than a few minutes go by without re-measuring sky. During lunar twilight, or when the Moon is near the horizon, when the sky is changing rapidly, measure sky both before and after each star.

And, if you are new to photometry, you won't believe how sensitive the results are to the slightest *hint* of a cloud anywhere in the sky. Even a tiny cloud anywhere in the sky means you probably have "incipient clouds" (variable patches of haze) all over the sky, because of the layered structure of the atmosphere. Even Steinheil, a century and a half ago, found that the slightest trace of cirrus made photometric measurements impossible — that's how gross the effects are.

It's a good idea to look at the sky around sunset. If there are "layers" visible in the sky near the western horizon, or "notches" in the sides of the setting sun, you are in for trouble, and will need to put extra effort into extinction determinations. During the night, any indications of clouds should be noted in your log. Often, cirrus is not visible until the Moon rises — or until the data are reduced!

13.5 Reducing the observations

13.5.1 Preliminaries

Now you have your data, and are ready to reduce them. Table 13.2 gives a list of the files you will need.

If you need to make supplemental standard-star files, refer to the "Standard-star files"

File type	Contents	Made by ...
Observatory	Telescope information	editing esotel.tbl
Built-in Standard-star	Std.-star positions & values	copying `calib` files
Additional Standard-star	Std.-star positions & values	edit files, `MAKE/STARTABLE`
Program-star	Program-star positions	`MAKE/STARTABLE`
Instrument	Instrument information	`MAKE/PHOTOMETER`
Data	Stellar measurements	`CONVERT/PHOT`

Table 13.2: Checklist of necessary table files

subsection at the end of section 13.2.1. If your data files use different names for standard stars than are used in some existing standard-star file, you can just supply a "program-star" file that contains your names for these stars; use `MAKE/STARFILE` to make this table. The reduction program will try to make the cross-identifications, using positional coincidences as well as names to decide which entries to match up.

If you were able to find additional information about the instrument while you were observing, be sure to add any missing information to the instrument file. The reduction program needs to have an accurate model of the instrument, as well as your procedure. You can check an existing instrument file by running `MAKE/PHOTOMETER`.

Next, you must convert your observational data to the format MIDAS can digest. If other people have already used the same equipment you did, there are probably programs available to do this conversion. If not, you will have to devise the conversion yourself.

In any case, you should look over the data files, line by line, to make sure there are no garbled records. Often, either equipment malfunctions, operator errors, or problems on reading and writing tapes convert some part of the data into nonsense. You cannot reasonably expect any file-conversion program to cope with nonsense, so edit it out first.

The sooner mistakes are removed from data, the less trouble they cause. Each data-logging system is likely to make its own peculiar kinds of mistakes; and each observer tends to make some particular kinds of errors at the telescope. Therefore, it is **strongly recommended** that users have their own preprocessing software, to detect mistakes and inconsistencies in raw data files, *before* running `CONVERT/PHOT`.

A common problem is an incorrect time or date setting; be particularly careful to correct any errors that were discovered belatedly while observing. Other common problems include misidentified stars; incorrect designations of star and sky measurements; missing or incorrect data in star files; and non-standard data formats.

In fixing errors in the data, *always* save a copy of the original in case you make a mistake while editing.

13.5.2 Format conversion

If you have a program to convert data to MIDAS input format, fine. Use it and go on to the next section.

If you do not, first see if there is *any* existing program that reads your data format. (Very likely there is, unless the instrument is new.) All you need to do is cannibalize the part of the existing program that reads the data, and add a short piece of code to re-format the data first as a flat ASCII file, one measurement per line; and then convert this to *.tbl format. Sound familiar? It's just like the process of creating a program-star table. Once again, a dummy *.fmt file is provided to help you.

That's usually the easiest way to go. You may decide to write a short MIDAS script to apply your format-conversion program and the edited *.fmt file to each night's data. That can be put into a procedure, given a name, and invoked as needed. This has already been done for data in the formats produced by the photometers at the 1-meter ESO and the Danish telescopes on La Silla; the command CONVERT/PHOT will convert their data files to the proper MIDAS tables.

If you are *very* lucky, your data may already be in the form of a flat ASCII file of consistent format; then all you need to do is edit the *.fmt file and apply it to generate the *.tbl files MIDAS needs. More likely, you have most of your data in this form, but interspersed with comments and/or records of other types. In this case, it may be possible to write a simple pre-processor (or use some UNIX utility, like `awk`) to strip out the data as a fixed-format ASCII file.

If there are just a few distinct record formats in your data files, it may be most convenient to strip out each type separately; convert each one first to a homogeneous ASCII file, and then to a MIDAS *.tbl file; and then use MERGE/TABLE to combine the separate tables into the final file for input to the reduction program. If each line of each file is correctly time-tagged, a simple sort on the MJD_OBS column will then put everything back together in the right order.

In any case, you should automate this procedure as much as possible. That means writing a short MIDAS or shell script. Whatever you do, *don't* resort to copying things out by hand! At worst, you might need to insert a little timing information by hand to label comments in the original data.

You should be careful to make sure enough information is in the final *.tbl file to tell the reduction program everything it needs to know. For example, if you measure dark current, and have more than one detector channel, the table must show which detector's dark current is being measured. That is usually indicated by the filter position; so you will probably need to have a non-blank filter position for "dark" measurements. Note, however, that you do *not* need to distinguish among standard, extinction, program, and variable stars in data files, as this information is either supplied when you read in the star files, or determined dynamically during the reductions.

The minimum of essential columns in a data file are object identification, band identification, signal, time, and integration time. The section "Required observational data" in the Appendix describes these columns. For standard band names in commonly-used systems, see the Appendix (particularly Table 2 in the section "Standard values" under "Star tables", and the description of the `BAND` column in the "Passbands" subsection under Section 5). Usually, you will also need the `STARSKY` column, to distinguish between star and sky measurements. But many other data can be useful: temperatures, relative humidity, various instrumental settings, and error estimates. The section "Additional information" in the Appendix describes these in detail.

Note that temperature and humidity data can be interpolated, if they are not routinely recorded with each observation. If your instrument table shows that neutral-density filters may be used, you need to include this information as part of the band designation in the data files. Conversely, if your data indicate that ND filters have been used, their presence and nominal attenuation factors should be included in the instrument table.

It would be nice to have a universal data-conversion utility. Unfortunately, the amazing variety of formats produced by dozens of home-grown data-logging systems seems to prevent it. You know your data better than anyone else; so you can put them in standard form better than anyone else.

13.5.3 Reductions — at last!

Now you really are ready to run the reduction program. You have all the necessary information in the right file formats. Usually, it's convenient to keep each night's observations in a separate file.

The command `REDUCE/PHOT` will start up the reduction program. It starts out very much like the planning program, as it also needs the telescope, standard-star, and instrument information. But instead of asking about output formats and the date of the observing run, it requests the names of the data files. Normally, each data file is one night; you can reduce up to 30 nights together.

To save typing, it is convenient to make a catalog that refers to all the data files; use the MIDAS command `CREATE/TCAT` to do this. For example, if your data files have names like night1.tbl, night2.tbl, and so on, the MIDAS command line

```
CREATE/TCAT  data  night*.tbl
```

will make a catalog named data.cat that refers to the whole group.

As with star files, if you say by mistake that there is another data file, you can enter "`NONE`" when it asks for another file name, and it will end the list and go on.

If the data are pulse counts, they are corrected for the nominal dead time (using a formula appropriate to the type of counter used) as they are read in. If they are raw magnitudes, they are converted to intensities at this stage, so the reduction can proceed in the same way for all types of data. At this point, observations are in intensity units on some arbitrary instrumental scale. The intensities are then corrected for the nominal values of any neutral attenuators that may have been used.

Once the data are read, the reduction proceeds in two main stages: first, filling in missing meteorological data, subtracting dark readings, and subtracting sky; and second, fitting the remaining stellar data to appropriate models.

If they are present, the temperatures and relative humidities for each night are displayed graphically. After looking at each graph, you can choose one of three treatments: polygon interpolation (i.e., just a straight line segment between adjacent data); linear smoothing (a straight line fitted to the whole set); or adopting a constant value for the whole night. Usually, polygon interpolation is adequate. However, it is sensitive to aberrant or bad points. If you believe there are outliers in the data, and the rest are reasonably linear, use the linear fit, which resists the effects of bad points. If you think a bad point can be fixed, or should be removed before proceeding, just enter "q" to quit, and deal with the bad datum.

Robust fits and bad points

Already, we must decide how to deal with discordant data. As this is a subject of considerable importance, a short digression is in order.

Poincaré attributed to Lippmann the remark that "Everyone believes in the normal law of errors: the mathematicians, because they think it is an experimental fact; and the experimenters, because they suppose it is a theorem of mathematics." However, it is neither an experimental fact nor a theorem of mathematics.

Experimentally, numerous investigations have shown that real errors are rarely if ever normally distributed. Nearly always, large errors are much more frequent than would be expected for a normal distribution (see [18], pp. 10 – 12, and [12], pp. 20 – 31). Menzies and Laing [17] show clear examples in photometric data.

Mathematically, the reason for this behavior is well understood: although the Central Limit Theorem promises a Gaussian distribution *in the limit* as the number of comparable error sources approaches infinity, the actual approach to this limit is agonizingly slow — especially in the tails, where a small number of large individual contributors dominate. In fact, if there are n independent and identically distributed contributors, the rate of convergence is no faster than $n^{-1/2}$ [11]. If we wanted to be sure that our distribution was Gaussian to an accuracy of 1%, we would need some 10^4 elemental contributions — clearly, an unrealistic requirement. In practice, a few large error sources dominate the sum.

Furthermore, the proportionality constant in the convergence formula changes rapidly with distance from the center of the distribution, so that convergence is very slow in the tails. This guarantees that the tails of real error distributions are always far from Gaussian.

In the last 30 years, the implications of these deviations from "normality" for practical data analysis have become widely appreciated by statisticians. Traditionally, the excess of large errors was handled by applying the method of least squares, after rejecting some subset of the data that appeared suspiciously discordant. There are several problems with this approach.

First, the decision whether to keep or reject a datum has an arbitrary character. A great deal of experience is needed to obtain reliable results. But manual rejection may be impractical for large data sets; and automated rejection rules are known to have inferior performance. Second, rejection criteria based on some fixed number of standard deviations result in no rejections at all when the number of degrees of freedom is small, because a single aberrant point greatly inflates the estimated standard deviation ([12], pp. 64 – 69). The common "3-σ" rejection rule rejects nothing in samples smaller than 11, no matter how large the biggest residual is; the inflation of the estimated standard deviation by just one wild point outruns the largest residual in smaller data sets. There is no hope of rejecting a bad point this way in samples of 10 or smaller; but one rarely measures the same star 10 times. For the more typical sample sizes of 3 and 4, the largest possible residuals are only 1.15 and 1.5 times the estimated standard deviation. Third, including or rejecting a single point typically introduces discontinuous changes in the estimated parameters that are comparable to their estimated errors, so that the estimated values undergo relatively large jumps in response to small changes in the data. We would have more trust in estimators that are continuous functions of the data.

Finally, the nature of most observed error distributions is not that data are clearly either "good" or "bad", but that the few obviously wrong points are accompanied by a much larger number of "marginal" cases. Thus the problem of rejection is usually not clear-cut, and the data analyst is left with doubts, no matter where the rejection threshold is set. The reason for this situation is also well understood: most data are affected by error sources that vary, so that the "marginal" cases represent data gathered when the dominant error source was larger than average. Such observations are not "wrong", though they clearly deserve smaller weights than those with smaller residuals.

In particular, we know that *photometric* data are afflicted with variable errors. For example, scintillation noise can vary by a factor of 2 on time scales of a few minutes; and by an additional factor of $\sec Z$ at a given air mass, depending on whether one observes along or at right angles to the upper-atmospheric wind vector. Menzies and Laing [17] discuss other possible sources of error. Therefore, we know we must deal with an error distribution that is longer-tailed than a Gaussian. Furthermore, both scintillation and photon noise are decidedly asymmetrical. As these are the main sources of random error in photometric observations, we can be sure that we *never* deal with truly Gaussian errors in photometry.

Unfortunately, the method of least squares, which is optimal for the Gaussian distribution, loses a great deal of its statistical efficiency for even slightly non-Gaussian errors. (Statistical efficiency simply refers to the number of observations you need to get a desired level of reliability. If one estimator is twice as efficient as another, it will give you the same information with half as many observations.) The classical example is Tukey's contaminated distribution. Suppose all but some fraction ϵ of the data are drawn from a normal distribution, and the remainder are drawn from another Gaussian that is three times as broad. Tukey [23] asked for the level of contamination ϵ that would make the mean of the absolute values of the residuals (the so-called *average deviation*, or A.D.) a more efficient estimator of the population width than the standard deviation, which is the least-squares estimator of width.

Although the mean absolute deviation has only 88% of the efficiency of the standard deviation for a pure Gaussian, Tukey found that less than 0.2% contamination was enough to make the A.D. more efficient. The reason is simply that least squares weights large errors according to the squares of their magnitudes, which gives them an unreasonably large influence on the results.

Similar, though less spectacular, results exist for position estimators. For example, about 10% contamination is enough to make the median as efficient as the mean (the least-squares estimator); while several "robust" estimators are some 40% more efficient than the mean at this level of contamination. Real data seem to be somewhat longer tailed than this, so the mean (i.e., least squares) is typically even worse than this simple example suggests.

Because convergence of the central limit theorem is much faster near the center of the error distribution than in the tails, we can expect real error distributions to be nearly Gaussian in the middle, and this is in fact observed to be true. A practical approach to data analysis is then to treat the bulk of the data (in the middle of the distribution) as in least squares; but to reduce the contribution of the points with large residuals, which would be rare in a genuinely Gaussian distribution, in a smooth and continuous fashion.

There is now a large literature on "robust" estimation — that is, on methods that are less critically dependent on detailed assumptions about the actual error distribution than is least squares. They can be regarded as re-weighted least squares, in which the weights of data with moderate to large residuals are decreased smoothly to zero. There are many ways to do this; all produce rather similar results. The really "wild points" are completely rejected; the marginal cases are allowed to participate in the solution, but with reduced weight. The result is only a few per cent less efficient than least squares for exactly Gaussian errors, and much better than least squares — typically, by a factor of the order of two —for realistic error distributions. These methods are also typically 10% or so more efficient than results obtained by experienced data analysts using careful rejection methods ([12], pp. 67 – 69).

The particular method used here for reducing photometric data is known to the statisticians as "Tukey's biweight"; it is easy to calculate, and produces results of uniformly high efficiency for a range of realistic distributions. To prevent iteration problems, it is always started with values obtained from even more robust (but less efficient) estimators, such as the median and its offspring, Tukey's robust line [13]. The usual method starts with a very robust but inefficient estimator such as the median or Tukey's robust line; switches to Huber's M-estimator for initial refinement until scale is well established; and then iterates to the final values using the biweight. If you are unaware of the need to precede the biweight with an estimator having an non-redescending influence function, don't worry. This is known to be a numerically stable procedure.

As robust methods depend on "majority logic" to decide which data to down-weight, they obviously require a certain amount of redundancy. One cannot find even a single bad point unless there are at least three to choose from (corresponding to the old rule about never going to sea with two chronometers). Therefore, it is better to obtain a large number of short integrations than a smaller number of longer ones, provided that the repetitions are separated in time enough to be independent. The planning program will help you get the necessary data.

In summary, photometric data are known to have decidedly non-Gaussian error distributions; so we use methods designed to be nearly optimal for these distributions, rather than the less reliable method of least squares. These methods are closely related to least squares, but are much less sensitive to the bigger-than-Gaussian tails of real error distributions. From the point of view of the average user, the methods employed here are simply a more effective refinement of the old method of rejecting outliers.

The advantage of using these well-established, modern methods is a gain in efficiency of some tens of per cent — exactly equivalent to increasing the amount of observing time by such an amount. It's about like getting an extra night per week of observing time. This advantage is well worth having.

Subtraction of dark and sky measurements

After interpolating any meteorological values that exist, you may have to subtract dark current and/or sky values. If there are no data for dark current, the reduction program will complain but continue. Sky values may have already been subtracted (e.g., in CCD measurements or other data marked as RAWMAG instead of SIGNAL.)

DARK CURRENT: Dark current is usually a strong function of detector temperature. If you have regulated the detector temperature, then only a weak time dependence might be expected — perhaps only a small difference from one night to the next, or a weak linear drift during each night. You will have the choice of the model to be used in interpolating dark values.

If the detector temperature is measured, you should look for a temperature dependence that is the same for all nights. The program will show you all the dark data, with a separate symbol for each night, as a function of temperature. If all nights seem to show the same behavior, you can then fit a smoothing function of temperature to the dark values. You can choose a constant, a simple exponential (i.e., a straight line in the plot of log (DARK) vs. temperature), or the sum of two exponential terms.

Although in principle these ought to be of the form $D = a\,exp\,(-b/T)$, the range of temperature available is usually insufficient to distinguish between this and a simple $a\,exp\,(cT)$ term. Furthermore, though the temperatures *ought* to be absolute temperatures in Kelvins, you may have only some arbitrary numbers available, which might even assume negative values. In this case, an attempt to fit the correct physical form would blow up, but the simple exponential term might still give reasonable results. So the simpler form is actually used.

If the plot of log (DARK) vs. temperature bends up at the ends, or at least at the right end, you should be able to get a good two-term fit. If it looks linear, you can just fit a single line. You also have the option of adopting a single mean value.

On the other hand, if the data are not consistent from night to night, or show a temperature dependence that is different from the expected form, or if you have no temperature information at all, you may have to interpolate the dark data simply as a function of time. As with the weather data, you have a choice of polygon, linear, or constant fits. Remember that a polygon fit uses every datum, right or wrong, and so is not robust.

After removing a temperature-dependent fit, you will see the remaining residuals plotted as a function of time. This provides a double check on the adequacy of dark subtraction.

SKY SUBTRACTION: Sky data must be treated separately for each night and passband. Here, your options are more numerous. You can choose the usual linear or constant fits; but those are likely to be a poor representation of sky brightness.

More conventional choices are to use either the preceding or following sky for each star observation, or the "nearest" sky (in which both time and position separations are used to decide what "nearest" means). Linear interpolation between successive sky measurements (i.e., a polygon fit) is also an option. These choices, while conventional, are not robust. They are sensitive to gross errors in sky data, such as star observations that have been marked as sky by mistake.

One might argue that bad sky data will stand out in the plots discussed below, and that careful users will remove them and re-reduce their data. One might also argue that really bad sky values will cause the stellar data to be discarded or down-weighted in the later reductions, so that a robust fit at this stage is not absolutely necessary. However, such arguments are not completely convincing. Therefore a more elaborate sky subtraction option is available, which tries to model the sky brightness, discriminating against outlying points in a robust regression.

To help you choose the best method, the program displays three plots of sky brightness against different independent variables: time, airmass, and lunar elongation. In the time plot, the times of moonrise and moonset are marked, and twilight data are marked t; Figure 13.1 shows an example. In the other two plots, points with the Moon above the horizon are marked with the letter M, points with the Moon below the horizon are marked by a minus sign, and twilight data are marked t. In these and other plots, the characters `^` on the top line or `v` on the lower edge indicate points outside the plotting area; and $ indicates multiple overlapping points. You can re-display the plots if you want to look at them again before deciding which sky-subtraction method to use.

```
u  SKY on  FEB 14 1993    R = moonrise    t = twilight sky

  X  I  ^                                                ^      ^^
  10.I
     I  t                      R
   3.+                         |                                t
     I                         |
     I
     I
   2.+                                                     *   tt
     I          *                                          $
     I           $*                       *      * **$**   *
     I  t         $*                     **$$**$*$* $$ * *     t
   1.+   tt       *      *       $ $$*$*$$$**$*$* **  * $      t
     I   t *             *   $$$* $*$$**$
     I   t  ****$*  *          **
     I       *$    * $$$$ $
   0.+               $
     I
     I
     I
  -1.+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+---
     0  2  4  6  8 10 12 14 16 18 20 22 24 26 28 30 32 34 36 38 40 42 44 4E-02
    TIME (MJD = 49032 + decimal of day) -->
```

Figure 13.1: Plot of sky brightness as a function of time

Note that no one method is best for all circumstances. While modelling the sky should work well under good conditions, there are certainly cases in which it will fail.

For example, when using DC or charge-integration equipment, an observer commonly uses the same gain setting for both (star+sky) and sky alone. This is perfectly appropriate, as it makes any electrical zero-point error cancel out in taking the difference. But often the limited precision available — for example, a 3-digit digital voltmeter — means that the sky brightness is measured with a precision of barely one significant figure when bright stars are observed. If a bright star reading is 782 and the sky alone is 3, one does not have much information to use in modelling the sky.

Another case where one does better to subtract individual sky readings is observations made during auroral activity. While one would prefer not to use such data, because of the rapidity of sky variations, they must sometimes be used. Here again, subtraction of the nearest sky reading is better than using a model, because the rapid fluctuations are not modelled. Likewise, when terrestrial light sources around the horizon make the sky brightness change rapidly with azimuth and/or time, no simple sky model would be adequate.

If it is necessary to make measurements of some objects through two or more different focal-plane diaphragms, these measurements cannot be combined directly. Ordinarily, all observations to be reduced together should be measured through the same aperture, because the instrumental system changes in an unpredictable way with aperture size. Even the sky measurements are not exactly proportional to the diaphragm area. However, it may be possible to reduce program objects observed with a non-standard aperture *as if* they were measured through the standard one, and then apply a suitable transformation after the fact. This means that a sufficient number of calibration measurements of stars having a considerable range in color *must* be taken, using both aperture sizes, to determine the transformation between the two instrumental systems. In such cases, individual sky readings taken through the same apertures must be used in the reductions. The reduction program will complain if you try to intermix data taken through different diaphragms, and data taken with a peculiar aperture will be rejected if there are no corresponding sky measurements.

Finally, when very faint stars are observed (as in setting up secondary standards for use with a CCD), so that the sky is a large fraction of the star measurement, it may be necessary to subtract individual sky readings simply because the model used is not sufficiently accurate. The model is reasonably good, but is not good enough to produce estimates free of systematic error.

In any case, the plots of sky vs. time, airmass, and lunar elongation should prove useful in assessing the quality of the sky data, and in choosing the best subtraction strategy. Furthermore, the residuals from the sky model may be useful in identifying bad sky measures that should be removed; so it is a good idea to run the sky model, even if you decide not to subtract its calculated values from the star data.

Sky models

For the user interested in the details of the methods used, here are the actual algorithms used in selecting the "nearest" sky, and in the sky-brightness model:

In the "nearest" method, a distance estimator is computed that includes separations both in time and on the sky. The estimator is

$$S = 20|t_1 - t_2| + |AM_1 - AM_2| + |(AM_1 + AM_2)(AZ_1 - AZ_2)|$$

where the t's are times in decimal days, the AM's are air masses, and the AZ's are azimuths (in radians) of the two observations being compared. (The azimuth difference is always taken to be less than π radians.) S crudely takes account of the greater variation in sky brightness with position near the horizon. At moderate air masses, it makes a separation of a minute in time about equivalent to a degree on the sky.

This estimator is computed for the two sky samples closest in time to each star observation, one before the star and the other after it. Only observations made with the same filter (and, if diaphragm size is available, with the same diaphragm) are used. The sky observation that gives the smaller value of S is the one used in the "nearest" sample method. Obviously, if the star is the first or last of the night, there may be no sky sample on one side; then the one on the other side (in time) is used.

The angular part of S is necessary to prevent problems when groups of variable and comparison stars are observed. It can happen that the sky is observed only after two stars in a group have been observed, if only a single sky position is used for the whole group. If sky was measured after the last star before the group, that previous sky may be closer to the first star of the group, in time alone, than the appropriate sky within the group. Then a purely time-based criterion would assign the (distant) previous star's sky to the first star of the group, instead of the correct (following) sky. Similar effects can occur at the end of a group, of course.

While this problem can be prevented by careful observers, not all observers are sufficiently careful to avoid it. The crude separation used here is adequate to resolve the problem without going into lengthy calculations. Note that both the airmass dependence of sky brightness and the possible presence of local sky-brightness sources around the horizon (e.g., nearby cities) make the horizon system preferable to equatorial coordinates for this purpose.

This brings us to the sky model. The general approach is to represent the sky brightness as the sum of two terms, a general one due to scattered starlight, zodiacal light, and airglow; and an additional moonlight term that applies only when the Moon is above the horizon.

The airglow and scattered starlight are proportional to the airmass, to a first approximation. Actually, extinction reduces the brightness of the sky light near the horizon. However, the full stellar extinction appears only in the airglow and zodiacal components, not in the scattered light. All three components are of comparable magnitude in the visible part of the spectrum.

While Garstang [6, 7, 8] has made models for the night-sky brightness, these require much detailed information that is not usually available to the photometric observer.

Garstang's models also were intended to produce absolute sky brightnesses, while data at this preliminary stage of reduction do not have absolute calibrations. Finally, they do not include moonlight. Therefore, a simpler, parametric model is used.

Some guidance regarding the scattered starlight can be obtained from the multiple-scattering results given by van de Hulst [24]. In photometric conditions, the aerosol scattering is at most a few per cent, and the total optical depth of the atmosphere is less than unity. In the visible, the extinction is dominated by Rayleigh scattering, which is not far from isotropic, and nearly conservative. Therefore, we are interested in cases with moderate to small optical depth, and conservative, nearly isotropic scattering. Because the light sources (airglow, stars, and zodiacal light) are widely distributed over the sky, we expect small variations with azimuth, and can use the values in van de Hulst's Table 12 to see that azumuthally-averaged scattering has the following properties:

1. For air masses such that the total optical depth along the line of sight is less than 1, the brightness is very nearly proportional to air mass, regardless of the altitude of the illuminating source.

2. For vertical optical depths τ less than about 1.5, the sky brightness reaches a maximum at an air mass on the order of $1/\tau$, and then declines to a fixed limiting value as $M \to \infty$ (remember that for the plane-parallel model, the airmass does go to infinity at the horizon).

The decrease in the scattered light at the horizon is also to be expected in the direct airglow and zodiacal components attenuated by extinction; so the same general behavior is expected for all components. The simplest function that has these properties is

$$B_1 = (aM + bM^2)/(1 + dM^2),$$

where M is the airmass; the limiting brightness at the horizon is just b/c. Actually, a substantially better fit can be obtained to the values in van de Hulst's Table 12 by including a linear term in the denominator; so the approximation

$$B_1 = (aM + bM^2)/(1 + cM + dM^2),$$

is used to represent the airglow and scattered light.

There is a problem in fitting such a function to sky data that cover a limited range of airmass. Except for optical depths approaching unity (i.e., near-UV bands), the maximum in the function lies well beyond the range of airmasses usually covered by photometric observations. That means that the available data usually do not sample the large values of M at which the squared terms become important. Thus, one can usually choose rather arbitrary values of these terms, and just fit the well-determined linear terms. It turns out that choosing $b = 0$ is often satisfactory. If the data bend over enough, d can be determined; otherwise, it defaults to 0.01 times the square of the largest airmass in the data.

An example of data that extend to large enough airmass to determine all four parameters is the dark-sky brightness table published by Walker [25], which were used by Garstang to check his models. These data extend to $Z = 85°$. The model above fits them about as well as do Garstang's models; typical errors are a few per cent. Various subsets, with the largest-Z data omitted, give similarly good fits. This indicates that the model is adequate for our purposes here.

In principle, one could add separate terms for zodiacal and diffuse galactic light that depend on the appropriate latitudes; but this seems an excessive complication, as these components vary with wavelength and longitude as well. We have also neglected the ground albedo. Unless the ground is covered with snow, this is a minor effect except near the horizon. Furthermore, the airglow can vary by a factor of 2 during the night; so we cannot expect a very tight fit with any simple formula.

The moonlight term is more complicated. In principle, it consists of single scattering, which in turn depends on the size and height distributions of the aerosols, as well as Rayleigh scattering from the molecular atmosphere; and additional terms due to multiple scattering. The radiative-transfer problem is complicated further by the large polarization of the Rayleigh-scattered component, which can approach 100%. Rather than try to model all these effects in detail, we adopt a simple parametric form that offers fair agreement with observation, but does not have too many free parameters to handle effectively.

First of all, van de Hulst [24] points out that the brightness of the solar aureole varies nearly inversely with elongation from the Sun. We assume the lunar aureole has the same property. And, for the small optical depths we usually encounter, and the moderate to small airmasses at which we observe, we can simply assume the brightness of the lunar aureole is nearly proportional to airmass.

Second, interchanging the directions of illumination and observation would give geometries related by the reciprocity theorem if the ground were black. For typical ground albedoes, we can still assume approximate reciprocity. We can also assume $\cos Z = 1/M$, where M is the airmass, accurate to a few per cent for actual photometric data, in calculating elongations from the Moon.

The adopted form is

$$B_2 = M(a/E + b + cE) \cdot [\exp(-dS) + e/P],$$

where M is the airmass in the direction of observation, E is the angular elongation from the Moon, S is the sum of observed and lunar airmasses, and P is their product. (Note that the lower-case parameters here are different from the ones in the dark-sky model.)

The factor in parentheses mainly represents the single-scattering phase function, and should be nearly constant in good photometric conditions. It is plotted as a “normalized” sky brightness. Its parameter a is a measure of the lunar aureole strength; if a/b is large, you probably have non-photometric conditions. The factor in brackets handles the reciprocity effects. The e/P term produces the correct asymptotic behavior for a homogeneous atmosphere; however, it cannot represent “Umkehr” effects at wavelengths where ozone absorbs strongly.

Both factors have the symmetry required by the reciprocity theorem. This condition may be violated by ground-albedo effects, and by photometers that have a large instrumental polarization.

Unfortunately, when the telescope is pointed so that the Moon can shine on the objective, the apparent aureole is nearly always dominated by light scattered from the telescope optics, not from the atmosphere. Even if the mirror has been freshly aluminized, the scattered light may not be negligible, because of surface scattering. This scattering has different angular and wavelength dependences from those of the true sky brightness, and so is not represented by the sky model. This means we should avoid observing so near the Moon that it can shine directly on the primary mirror.

However, we cannot avoid having the *star* shine on the mirror; so we must include a term in the "sky" model that is proportional to the brightness of the measured star, to allow for the wings of the star image that we include in our "sky" measurements. Even if this fraction is so small (say, below 10^{-3}) as to have no effect on the sky-subtracted stellar data, it can be important in the sky *model*, if fairly bright stars are observed. The fraction of starlight measured depends on the distance from the star to the sky area chosen; as mentioned before, one must keep this distance fixed for all stars when observing.

Using the sky model

If you elect to model the sky, you will see some graphs showing the progress of the fitting procedure. First, if some bright stars were observed, you see a plot of sky data (for a given band and night) as a function of the adjacent star's brightness. A crude linear fit, indicated by + signs, shows the initial estimate of the star contribution to "sky" data. Second, if there are enough dark-sky data to model, you will see a plot of the dark-sky data (corrected for star light) as a function of airmass, with the fitted B_1 term drawn in as + signs. Both these plots show instrumental intensity units on the vertical scale.

Next, if there are enough data with the Moon above the horizon to fit the B_2 term, you will see a plot of the moonlit sky, corrected for both the stellar and the dark-sky terms, as a function of elongation from the Moon. To show the importance of the lunar aureole (whose presence is an indicator of considerable aerosol scattering, and hence probably non-photometric conditions), this plot is normalized by multiplication by the factor in square brackets (cf. the equation for B_2 in the previous subsection) and division by the airmass; that is, it is simply a plot of the E-dependent factor in parentheses. Again, the fit is shown with + signs. If the subtraction of the stellar and dark-sky terms produced some apparently negative intensities, the calculated zero level for the B_2 term will be drawn as a horizontal line.

If the sky is good, this plot will be nearly horizontal. Usually, it bends up near 0 and 3 radians, with a minimum near 1.7; your data may not cover this whole range, so pay attention to the numbers on the horizontal scale. The vertical scale is chosen to make the range of most measurements visible, so the zero level may be suppressed; look at the numbers on the vertical scale.

It often happens that the range of the independent variables in these fits is inadequate to allow a full fit of all the parameters. Reasonable starting values will be used for the indeterminate parameters. The fitting strategy is to adjust the best-determined parameters first, and release parameters in turn until nothing more can be done. At each stage, the results of a fit are examined to see whether the values determined are reasonable. For example, most of the parameters must be positive; and the dimensionless ones should not be much larger than unity.

You will be informed of the progress of these fitting attempts. Do not be alarmed by "error messages" like `DLSQ FAILED` or `SINGULAR MATRIX` or `Solution abandoned`. It is quite common for solutions to fail when several parameters are being adjusted, if the data are not well distributed over the sky. Sometimes, when a solution fails, the fitting subroutine will check to make sure the partial derivatives were computed correctly. This is a safety feature built into this subroutine, and you should always find that the error in the derivatives is 10^{-6} or less. The program should then comment that the model and data are incompatible, because we are usually trying to fit more parameters than the data can support. Remember that the program will adopt the best solution it can get; so watch for messages like `3-parameter fit accepted`, and don't worry about the failures.

Pay more attention to the graphs that compare the fits to the data. Are there regions where they become widely separated? If so, the fit is poor, and you can forget about the model. If the fit looks good, the model is useful for detecting bad sky data, and may even be useful for interpolating missing sky measurements.

When the fitting is complete, the program will print the terms used in the fit. Then, three summary graphs display the quality of the fit. The moonlit data are marked M in each of these three plots. The first shows the observed sky brightness as a function of the model value. This plot should be a straight line.

The second diagnostic plot shows the *residuals* of the sky fit as a function of the adjacent star brightness. These points should be clustered about the axis indicated by dashes. If you tend to measure sky farther from bright stars than from faint stars, as some beginners tend to do, the points will show a downward trend toward the right. That's a sign that you need to be more careful in choosing sky positions. Likewise, a large scatter in this plot probably means you have been careless in choosing sky positions, sometimes measuring closer to the star and sometimes farther away. (Here, "large" means large compared to the typical sky values on previous plots.)

The final plot in this group shows the *ratio* of the observed to the computed (modelled) sky brightness, as a function of time. If the airglow changes with time, you will see waves and wiggles in the dark-sky portion. Likewise, if the aerosol is varying with time, you will see coherent variations in the moonlit portion. The upper and lower limits of this plot are fixed in absolute value, so the scatter visible here is a direct indication of the quality of the overall fit.

Finally, if there are aberrant points that do not fit the model, they will be tabulated. If the data are not well represented by the model, there will be many entries in this table. Pay particular attention to the last column, which gives the ratio plotted in the last diagnostic graph. If the fits were generally satisfactory, the few sky measurements tabulated here may be in error, and may indicate a instrumental or procedural problem. They should be examined carefully to determine the cause of the problem.

After all this information has been presented, you will be asked whether you want to subtract the modelled sky from the stellar data. You can reply Yes or No, or enter R to repeat the whole process, or H to ask for help. If you reply Yes, the model values will be subtracted from all the stellar data that were not taken during twilight. However, as twilight is not modelled, the nearest-neighbor method will be used to correct stars observed during twilight. If you reply No, you will be given the option of subtracting the nearest neighbor in all cases.

The sky models do not work well if only a few observations are made with the Moon either above or below the horizon. They do not handle solar or lunar twilight. They can have difficulties if the observations are not well distributed over the sky. In these, as in some other cases discussed above, one should choose some other method instead of using a model for sky subtraction.

13.5.4 Choosing a gradient estimator

Now the program knows which stars have been observed, and can show you a two-color diagram of the available standards. It displays a plot of the two nearest color indices for each band. For example, in the *uvby* system, plots of $(u - v)$ vs. $(v - b)$ are displayed for the u and v passbands, and plots of $(v - b)$ against $(b - y)$ for both b and y. The degree of correlation between the two displayed color indices is also printed. The goal is to select an appropriate estimator of the stellar spectral gradient across each passband.

You are now asked to choose whether to use the single default color index (which would be $(u - v)$ for u, and $(u - b)$ for v), or *both* colors displayed on the plot, as the basis for transformations. In general, if the plot shows a high degree of correlation, adding the second index adds little information, and may weaken the solution by allowing an additional degree of freedom to be adjusted. If the points cover an appreciable two-dimensional area, or follow a curved line rather than a straight one, you may gain substantial accuracy in transforming by using both indices as independent variables. When in doubt, ask the program for help.

Note that you make this choice separately for each passband. (It might make sense to use two colors in reducing b, but only one in reducing y, even though the same 2-color diagram is presented in each case.) Once you have set this choice, it will be followed throughout the remainder of the run.

Note that you will be given this choice only if there are enough standard stars to make it a reasonable course of action. With only a few standards, and too many adjustable parameters, there is the danger that the program may simply fit a function to a few points and ignore the rest.

In principle, these plots should employ the instrumental rather than the standard colors of the standard stars. Unfortunately, the choice of a gradient estimator must be made before extinction-corrected instrumental colors are available, so the standard colors must stand in for the instrumental ones. In practice, this should not be a problem, unless the instrumental system is a very poor match to the standard one. For example, if the B band of a UBV photometer includes too much ultraviolet, the instrumental B-V index may contain appreciable contamination from U that would not be seen in the standard 2-color plot.

13.5.5 Extinction and transformation models

To determine the parameters we want (stellar magnitudes and colors) from the observations we have, we fit the data to a model. The model should represent the circumstances of the observations as realistically as possible; otherwise, effects present in the data but not in the model will be distributed among the available adjustable parameters, such as stellar magnitudes, extinction coefficients, and transformation parameters. The result may well be a good fit to the data, with small residuals; but the parameters will be biased — that is, they will have significant systematic errors. Such errors are called *reduction errors* by Manfroid and Sterken [16]

The bias problem

Bias can also be introduced by improper fitting procedures. For example, the reduction program known as SNOPY, used for many years at La Silla and also at La Palma, uses a simple iterative technique in which some subsets of the parameters are solved for, while others are held fixed; then the ones last solved for are held fixed, while the ones previously fixed are redetermined. The earliest version of PEPSYS used this technique [26]. However, it required *hundreds* of iterations to approach the true minimum in the sum of squares, and consequently was abandoned as soon as computers became large enough to allow simultaneous solutions for all the parameters.

A particularly treacherous aspect of such subspace alternation is that the sum of squares decreases rapidly for the first few iterations and then levels off. If the iterations are performed by hand, as they are with SNOPY, the user is likely to think that the system has converged when both the residuals and the parameter changes have become small. Nevertheless, the current values of the parameters can be far from the desired solution.

What happens is that the solution finds its way to the floor of a long, narrow valley in the sum-of-squares hypersurface in the n-dimensional parameter space. Even with optimal scaling, such valleys occur whenever the independent variables are partially correlated, so that the parameters themselves are correlated. This nearly always happens in multiparameter least-squares problems. At each iteration, the descent to the valley floor (i.e., the minimum in the mean-square error) occurs within the subspace of the parameters that are adjusted at each iteration, but is necessarily at the starting values of the parameters that were held fixed. At the next iteration, the solution moves a short distance in this orthogonal subspace, and again finds a point on the valley floor; but, again, it only finds a *nearby* point on the axis of the valley, and is unable to progress *along* the axis, because of the parameters that are held fixed. Succeeding iterations take very small steps, back and forth across the valley axis, while making only very slow progress toward the true minimum at the lowest point on that axis. This behavior is notorious in the numerical-analysis community, where it is known as "hemstitching".

The fatal slowness of convergence along coordinate directions is explicitly described and illustrated on pp. 294-295 of *Numerical Recipes* [20], although the schematic diagram shown there does not show how slow the process really is. Actual computations show that hundreds or even many thousands of iterations may not be enough to get acceptably close to the true minimum: see, for example, Figure 4j of Gill et al. [9] or Figure 9.7 of Kahaner et al. [14]. But in every iteration after the first few, the sum of the squared residuals is small and changes very slowly. One must not forget that obtaining small residuals is not an end in itself, but is merely the means of finding what we really want, namely, the best estimates of the parameters.

The most effective way to prevent hemstitching is to adjust all the parameters simultaneously. In photometric problems, this means solving a rather large system of simultaneous equations. For example, if we have 100 stars observed in 4 colors on 5 nights, there are 400 different stellar parameters, 25 extinction coefficients and nightly zero-points, and a few additional transformation coefficients to be determined, if they are estimated simultaneously. In broadband photometry, there are also parameters proportional to the second central moments of the passbands to be estimated, which introduce cross-terms and hence nonlinearity. Fortunately, the nonlinearity is weak, and can be very well handled by standard techniques (cf. [3]) that converge quickly.

The problem can be speeded up further by taking advantage of its special structure: the matrix of normal equations is moderately sparse, with a banded-bordered structure; the largest block (containing the stellar parameters) is itself of block-diagonal form. Thus matrix partitioning, with a block-by-block inversion of the stellar submatrix, provides the solution much more quickly than simple brute-force elimination of the whole matrix. A single iteration takes only a few seconds, in typical problems.

Equations of condition

Assured of a reliable method of solving the normal equations, we now consider the equations of condition. The observed magnitude m of a star at M air masses with true instrumental magnitude m_0 is modelled as ([10], [28])

$$m = m_0 + A_{eff} M + Z_n,$$

where Z_n is a nightly zero-point. The effective extinction coefficient A_{eff} depends on the effective spectral gradient [29] of the star, for which we use some color index, affected by half the atmospheric reddening ([26], [10]):

$$A_{eff} = A_0 - W(C_0 + RM/2).$$

Here A_0 is the extinction coefficient for the effective wavelength of the passband; W is proportional to the second central moment of the instrumental passband; C_0 is the extra-atmospheric color of the star (obtained by differencing the m_0 values in the appropriate pair of passbands); and R is the atmospheric reddening per unit airmass, obtained by differencing the corresponding pair of A_0 values.

On general numerical-analysis grounds, we initially assume the best pair of passbands to use for C_0 and R to be the pair that flank the passband in which m was observed. Thus, for the B band of UBV, we use the (U-V) color. This is not conventional practice; however, it provides a general rule that works reasonably well for all photometric systems. Such a rule is required in a general-purpose reduction program. For a well-sampled system, this produces a strikingly accurate representation of the extinction correction [30]. For under-sampled systems, we can use a somewhat more accurate gradient estimator, using a linear combination of the two adjacent color indices, as described above (see subsection 13.5.4, "Choosing a gradient estimator"). For bands at the extreme wavelengths of our system, we adopt just the neighboring color index (e.g., (U-B) for the U magnitude and (B-V) for the V magnitude of UBV).

Regarding the use of (U-V), we may note Bessell's remark [2] (in connection with the problems created by the wide and variable B band) that "in retrospect, it would have been much better had (U-V) rather than (U-B) been used by Johnson in his establishment of the UBV system." In any case, there are special problems in UBV, both due to the inadequate definition of the original system, and the neglect of the transformation from inside to outside the atmosphere entirely in the (U-B) index, which made the "B" of (B-V) and the "B" of (U-B) have different effective wavelengths; in principle, it is incorrect to add the two indices to a V magnitude and come up with a "U magnitude" as a result. Consequently, one must be very careful in doing UBV photometry, no matter how it is reduced; the only safe course is to observe at least the 20 standard stars recommended by Johnson (more would be preferable), and to look very carefully for systematic trends in the transformation residuals.

Although, for astrophysical reasons, there may be partial correlations of the true stellar gradient at a given band with color indices that are remote in wavelength from the band in question, such correlations will depend on metallicity, reddening, and other peculiarities of individual stars. If correlations obtained from one group of stars are applied to another group, the results may easily be worse than if the additional correlation had been ignored in the first place. Thus, it is exceedingly dangerous to employ distant color indices unless the calibration and program stars are very similar in all these respects. We cannot rely on such good matching in general, so these partial correlations are not used in the present package. However, one can expect that very good results will be obtained if the program and extinction/standard stars cover the same region of parameter space. That is, they should have the same ranges of spectral types, reddening, chemical composition, etc.

Because filter photometers observe different bands at different times, we have to do the reduction in terms of magnitudes (which are measured), rather than colors (which are not — see [10], pp. 152 – 154). This also allows isolated observations of extinction stars in a single filter, or a partial subset of filters, to be used. Furthermore, the best estimate of the extra-atmospheric color C_0 is used to reduce every observation, so that errors in individual measurements have the least effect on the reduction of each particular observation.

Of course, it may be necessary to allow the extinction A_0 to be a function of time; we always solve for individual values on different nights, as many careful investigations have demonstrated the inadequacy of "mean extinction". The instrumental zero-point terms Z may likewise be functions of time, temperature, or relative humidity. Finally, the zero-points may be kept the same in each passband for all nights in a run if the instrument is sufficiently stable.

Strategy

Bear in mind that you should have *enough* standard stars, but not too many. It is better to have 5 observations each of 30 standard/extinction stars, than to have 3 observations each of 50 stars; in the latter case, we are spreading the same observational weight over nearly twice as many parameters. Because one observation is "used up" in determining the stellar values, only $(n-1)$ of n observations per star are available for measuring extinction. Thus a star observed 5 times contributes twice as much to the extinction determination as a star observed only 3 times, which in turn contributes twice as much as a star observed only twice. The planning program will help you choose the right number of stars.

Furthermore, the running time of the reduction program is roughly proportional to the number of observations, but to the *cube* of the number of parameters to be adjusted. Thus, the program will take almost 5 times as long to run if there are 50 stars to be adjusted than if there are only 30. The storage required is proportional to the square of the number of parameters, and this may limit the number that can be handled on machines with small memories. At some point, any machine becomes overloaded; there is a practical limit to the number of stars that can be adjusted in a single solution. The reduction program automatically excludes non-standard stars that have been observed only once from the extinction solution.

In sum, though one can hardly have too many *observations* of standard stars, one should not go beyond the optimum *number of stars* recommended by the planning program. Some observers who follow Hardie's [10] method observe a large number of standard stars just once each; this is extremely inefficient use of telescope time.

Although these questions ought to be considered before one goes to the telescope, some observers only consider them when trying to reduce data that have been badly distributed. Even a sophisticated reduction program cannot generate information that was not gathered at the telescope.

Standard stars

Standard stars can be included in the extinction solution, if one is careful. The problem is that there are usually appreciable "conformity errors" due to mismatch between the instrumental and the standard passbands [16]. These are due to the integrated influence of features in the spectra of individual stars, which differ in age, metallicity, reddening, etc. at a given color index value. Thus, it is wrong to assume that the transformation between the instrumental system and the standard system is exact, even if there were no measurement error in either system [1].

Usually, one finds that the instrumental system can be reproduced internally with much more precision than one can transform between it and the standard system. This means we should regard the standard values as imprecise estimates of the extra-atmospheric instrumental values, after transforming from standard to instrumental system. In effect, we regard the standard values as noisy pseudo-observations of the instrumental, extra-atmospheric magnitudes.

Therefore, a reasonable equation of condition to use for the standard values is:

$$m_{std} = m_0 + aC_0$$

where we effectively regress the noisy (because of conformity errors) standard values on the internally precise instrumental system. No zero-point term appears, as we already took it out in conjunction with the extinction. As before, m_0 is the extra-atmospheric instrumental magnitude, and C_0 is the estimator of stellar gradients across the passband being transformed; the same gradient estimator should be used in both extinction and transformation equations. The transformation coefficient a is then estimated in the general solution. In general, we expect a higher-order polynomial in C_0 to be needed [29]; it can be determined if enough standard stars are available. This equation then gives an explicit transformation from the instrumental to the standard system.

One problem is to know what weights to assign such pseudo-observations. The conformity errors in the transformation are usually much larger than the internal errors of measurement in the standard system, so that quoted uncertainties in well-observed standard stars are usually irrelevant to our problem. For convenience, unit weight in the extinction solution is intended to correspond to errors of 0.01 magnitude. This is a typical order of magnitude of conformity errors as well, so we can start with unit weight for these values, and adjust the weights after a trial solution, followed by examination of the residuals. This process assigns self-consistent weights to the standard values.

Alternatively, we can omit the standard stars from the extinction solution, determine the extinction entirely from the observations, and then determine the transformation in a separate step. In this case, a separate set of zero-points must be determined in the transformation solution, as the nightly zero-points in the extinction solution are no longer coupled to the transformation. However, this does not increase the number of unknowns, as we must then fix one night's zero points to prevent indeterminacy. (In principle, one should do this using Lagrange multipliers; in practice, it hardly matters, because the nightly zero points are always determined much more precisely than other parameters in the solution.) This process is preferable if there are enough extinction observations, as it does not propagate systematic errors from the transformation into the extinction (see pp. 178 – 179 of [10], and [19]).

However, observers sometimes do not get enough extinction data to permit solving for extinction directly. Those who follow Hardie's [10] bad advice sometimes observe each standard star just once; then, if there are no real extinction stars, the standard values have to be used to obtain any extinction solution at all. But usually, in combined solutions, the extinction coefficients will be slightly more precise, but less accurate, than in separate solutions. The reduction program will tell you if one method or the other seems preferable; in any case, it lets you decide which method to use.

There has been much splitting of hairs in the astronomical literature over the direction in which the standard-star regression should be performed. Unfortunately, the arguments have all assumed that the regression model is functional rather than structural; that is, they assume there is an exact relation connecting the two systems, in the absence of measurement noise. In practice, conformity errors are usually larger than measurement errors, so the functional-regression model is incorrect. In any case, the problem we have here is a calibration problem: given measurements of some stars in the standard system A and the instrumental system B, we want to predict the values that *would* have been observed in A from the values we *have* observed in B, for the program stars. In this case, the regression of A on B provides the desired relationship [5].

While the conformity errors make it reasonable to do the regression in the sense described above, it is also clear that they involve significant effects that are not accounted for in the usual treatments of transformation. In particular, substantial *non-linear* terms are to be expected in transformations, and the neglect of higher-order cross-product terms in general makes transformations appear to be *multivalued* [29]. The correct treatment of these problems requires a well-sampled system. Trying to correct for missing information after the fact is a lost cause; effort should instead go into minimizing the conformity errors in the first place.

In general, one cannot emphasize too strongly the need to measure the instrumental spectral response functions, and to choose filters — by actual transmission, not by thickness! — that reproduce the standard system as accurately as possible. The difference between instrumental and standard response functions represents a serious disparity that cannot be made up by any reduction algorithm, no matter how cleverly contrived.

13.5.6 Reduction procedure

After subtracting dark current and sky brightness from your stellar data, and asking you to select a gradient estimator for each band, the reduction program converts intensities to magnitudes. At this stage, it warns you of any observations that seem to have zero or negative intensities (which obviously cannot be converted to magnitudes). These are often an indication that you have confused sky and star readings. The program then estimates starting values for the full solution.

The program treats four different categories of stars differently: standard stars, extinction stars, program stars, and variable stars. It asks for the category of each file of stars when the positions are read in. *Standard* stars are those having known standard values, which are used only to determine transformation coefficients. They may also be used to determine extinction. Be cautious about using published catalog values as standards; these often have systematic errors that will propagate errors into everything else. *Extinction* stars are constant stars, observed over an appreciable range of airmass, whose standard magnitudes and colors are unknown, or too poorly known to serve as standards. Ordinary stars taken from catalogs of photometric data are best used as extinction stars; later, you can compare their derived standard values with the published ones as a rough check. *Program* stars may be re-classified as either extinction or variable stars during the course of the extinction solution. If they are not used as extinction stars, they are not included in the solutions, but are treated as variable stars at the end. *Variable* stars are excluded from the extinction and transformation solutions. Their individual observations will be corrected for extinction and transformation after the necessary parameters have been evaluated.

You will do best to maintain separate star files for each category; however, you can intermix extinction and variable stars in a "program star" file, and PEPSYS will do the best it can to sort them out, if you ask it to use program stars for extinction (see below). Remember that only star files need to be separated this way; a data file normally has all the observations for a given night, regardless of the type of object.

You can also group related files in a MIDAS "catalog" file. Use the MIDAS command `CREATE/TCAT` to refer to several similar *.tbl files as a catalog file. Note that (a) you must enter the ".cat" suffix explicitly when giving PEPSYS a catalog; and (b) a catalog *must* contain only tables of the same kind — e.g., only standard stars, or only program stars. Catalogs can be used for both star files and data files.

When all the star files have been read, the program asks for the name of the table file that describes the instrument. As usual, you can omit the ".tbl" suffix and it will be supplied.

Then the program asks for the data files. Remember to add the ".cat" suffix if you use a catalog. While reading data, it may ask you for help in supplying cross-identifications if a star name in the data did not occur in a star table. If all your data files have been read, but it is still asking for the name of a data file, reply `NONE` and it will go on. (This same trick works for star files too.)

The program will display a plot of airmass vs. time for the standard and extinction stars on each night, so you can judge whether it makes sense to try to solve for time-dependent extinction later on. If the airmass range is small, it will warn you that you may not be able to determine extinction.

If the data are well distributed, and there are numerous standard stars, it will obtain starting values of extinction coefficients from the standard values; otherwise, it assumes reasonable extinction coefficients and estimates starting values of the magnitudes from them. (These starting values are simple linear fits, somewhat in the style of SNOPY, except that robust lines are used instead of least squares.) From the preliminary values of magnitudes, it tries to determine transformation coefficients, if standards are available.

This whole process is iterated a few times to obtain a self-consistent set of starting values for all the parameters, except for bandwidth factors. Each time the program loops back to refine the starting values, it adds a line like "`BEGIN CYCLE  n`" to the log. Don't confuse these iterations, which are just to get good starting values, with the iterations performed later in the full solution.

One problem in this preliminary estimation is that faint stars with a lot of photon noise might just add noise to the extinction coefficients. The program tries to determine where stars become too faint and noisy to be useful for estimating extinction. The rough extinction and transformation coefficients will be displayed as they are determined; if any fall outside a reasonable range of values, the program will tell you and give you a chance to try more reasonable values.

When reasonable starting values have been found for the parameters that will be estimated in the full solution, the program still needs to decide how to do the solution. Should the standard values of the standard stars be used in determining extinction (which requires estimating transformation coefficients simultaneously), or should extinction be determined from the observations alone, and the transformations found afterward? The program will ask your advice.

If you have designated no extinction stars, the reduction program will ask you whether you want to treat program stars as extinction stars. If you believe most of them are constant in light, you can try using all of them as extinction stars. If some turn out to be variable, you will have a chance to label them as such later on. If many of the program stars are faint, they may be too noisy to contribute anything useful to the extinction solution; then you could leave them out and speed up the solution. On the other hand, values obtained from multiple observations in the general solution will be a little more accurate than values obtained by averaging individual data at the end.

When the program is ready to begin iterating, it will ask how much output you want. When you first use PEPSYS, you may find it useful to choose option 2 (display of iteration number and variance). After you get used to how long a given amount of data is likely to take to reduce, you can just choose option 1 (no information about iterations). The detailed output from the other options is quite long, and should only be requested if the iterations are not converging properly and you want to look at the full details.

If you ask for iteration output, you will always see the value of the weighted sum of squares of the residuals (called WVAR in the output), and the value of the typical residual for an observation of unit weight (called SCALE), which should be near 0.01 magnitude. The weights are chosen so that the error corresponding to unit weight is intended to be 0.01 mag. When SCALE changes, there can be considerable changes in WVAR — in particular, don't be alarmed if it occasionally increases. A flag called MODE is also displayed, which is reset to zero when SCALE is adjusted. During normal iterations, MODE = 4; ordinarily, most of the iterations are in mode 4, with occasional reversions to mode 1 when SCALE is adjusted. More detailed output contains the values of all the parameters at each iteration, and other internal values used in the solution; these are mainly useful for debugging, and can be ignored by the average user.

Typically 20 or 30 iterations are needed for convergence; if the data are poorly distributed, so that some parameters are highly correlated, more iterations will be needed. If convergence is not reached in 99 iterations, the program will give up and check the values of the parameters (see next paragraph). This usually indicates that you are trying to determine parameters that are not well constrained by the data. It will also check the values of the partial derivatives used in the problem; if there is an error here larger than a few parts in 10^6, you have found a bug in the program.

At the end of a solution, the program prints the typical error, and then examines the parameters obtained. If extinction or transformation coefficients or bandwidth parameters seem *very* unreasonable, the program will simply fix them at reasonable values and try again. If they seem marginally unreasonable, the program will ask for your advice.

When reasonable values are obtained for all the parameters, the program will check the errors in the transformation equations, if standard values have been used in the extinction solution. Then, if necessary, it will readjust the weights assigned to the standard-star data, and repeat the solution. Usually 3 or 4 such cycles are required to achieve complete convergence. Thus, including the transformation parameters in the extinction solution means the program may take longer to reach full convergence.

Having reached a solution in which the weights are consistent with the residuals, the program examines the stellar data again. If you have "program" stars that might be used to strengthen the extinction solution, it will ask if you want to use all, some, or none of them for extinction. If you reply SOME, it will go through the list of stars one by one and offer those that look promising; you can then accept or reject each individual star as an extinction star. Only stars whose observations cover a significant interval of time and/or airmass will be offered as extinction candidates.

In examining the individual stars, the program may find some that show signs of variability. For those that have several observations, a "light-curve" of residuals will be displayed. Pay close attention to the numbers on the vertical scale of this plot! Each star's residuals are scaled to fill the screen. If there are only a few data for a star, only the calculated RMS variation is shown.

A star may show more variation than expected, but if this is under 0.01 magnitude, it may still be a useful extinction star — indeed, the star may not really be variable, but may have had its apparent variation enhanced by one or two anomalous data points. Another problem that can occur, if you have only a few extinction stars, or only one or two nights of data, is a variation in the extinction with time. This can produce a drift in the residuals of a standard or extinction star that may look like some kind of slow variation. Watch out for repeated clumps of anomalously dim measurements that occur about the same time for one star after another; this often indicates the passage of a patch of cirrus.

The program will show you several doubtful cases for every star that really turns out to be variable, so be cautious in deciding to treat them as variable stars. If you decide that a star really looks variable, you can change its category to "variable", and it will be excluded from all further extinction solutions.

If you change the category of any star, the program goes back and re-does the whole extinction solution again from the very beginning. When you get through a consistent solution without changing any star's category, the program announces that it has done all it can do, displays some residual plots, and prints the final results.

All reductions are done in the instrumental system, even if standard-star values (and hence transformations) are included as part of the extinction solution. It may be helpful to look at a schematic picture of the reduction process (see Figure 13.2 on the next page).

13.5.7 Final results

Residual plots

First, there is a set of residual plots against airmass for each night. The plots for individual passbands are compressed so that two will fit on the screen at once; this allows you to notice correlations in the behavior of residuals in adjacent bands.

Next, there are residuals as functions of time for each night (again compressed to make comparisons easy). Examine these plots carefully for evidence of trends with time. After you have inspected the run of the residuals with time, the program will ask if you want to solve for time-dependent parameters. If you reply "YES", it will show you some plots intended to help you distinguish between time-dependent extinction and time-dependent zero-point drift. These plots are accompanied by some numbers that may also be helpful (see Fig. 13.3).

The principle behind these plots is that time-dependent extinction produces errors (and hence, we hope, residuals) that are proportional to airmass; but instrumental drift produces residuals uncorrelated with airmass. Suppose we have two successive observations at airmasses M_1 and M_2, which produce residuals ϵ_1 and ϵ_2. If the extinction changes slowly with time, we expect the ratios $(\epsilon_1 - \epsilon_2)/(M_1 - M_2)$ and $(\epsilon_1 + \epsilon_2)/(M_1 + M_2)$ to be equal, as each is (on the average) the deviation of the current extinction coefficient from the mean extinction for the night, multiplied by the airmass difference of the points being

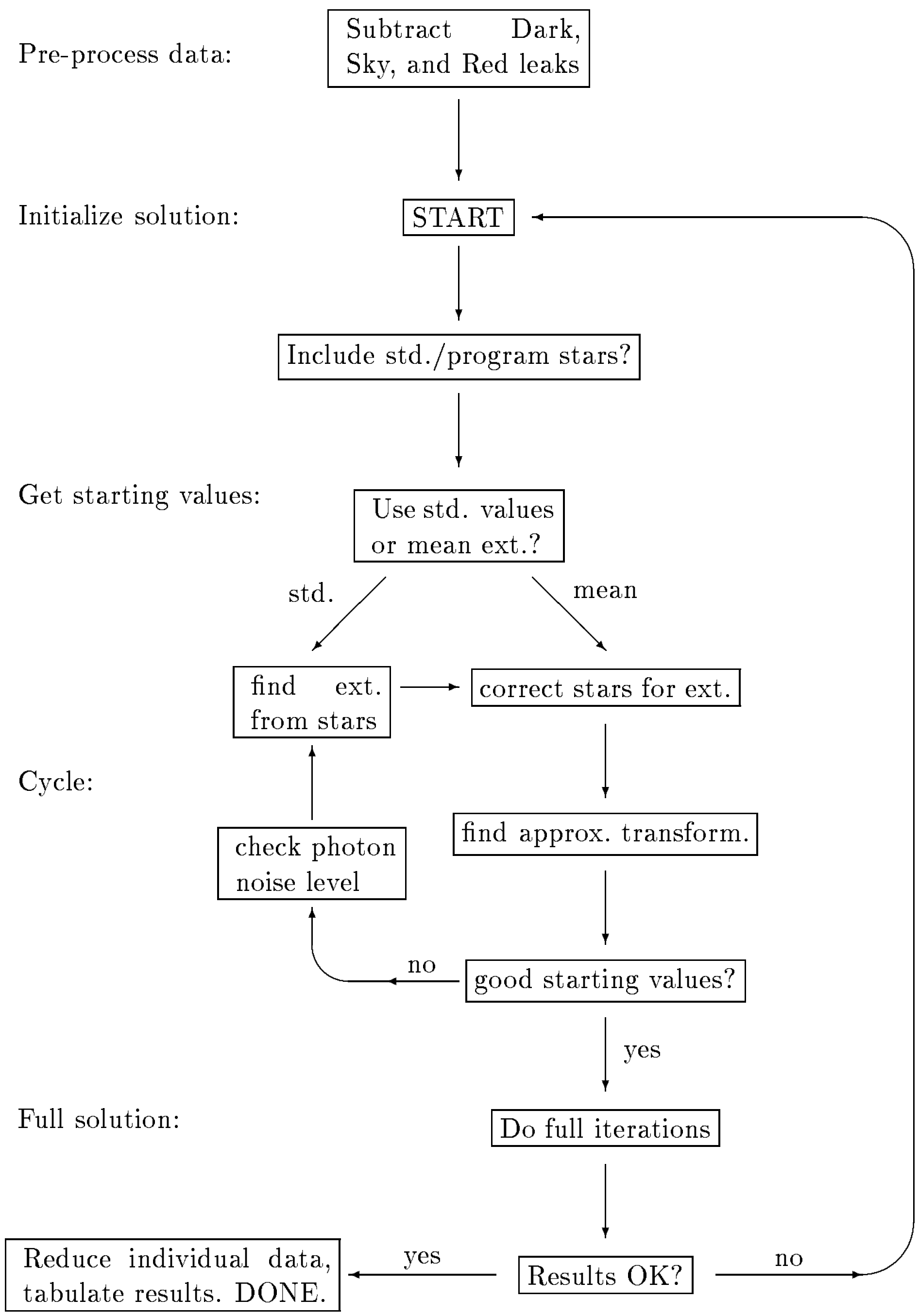

Figure 13.2: Schematic flowchart for reductions

```
                   v drift plot     slope = 1.50   32 data
differential offset
  X  I
 .01 I
     I                          *
     I
  -1.+
     I                                *
     I                   *   *  *                   *
     I
     I                 *    *  $               +++++++++++++
   0.+   - - -  - - ++++++++++*++++++++++++++++  - - -
     I  ++++++++++++ *                * *
     I   *   *                 $*       $
     I      *                    *        *
     I           *
   1.+
     I                  *
     I
     I        *
     I
   2.+---+---+---+---+---+---+---+---+---+---+---+---+---+--
     14  12  10   8   6   4   2   0  -2  -4  -6  -8 -10 E-04
     mean offset  -->     ratios:  0.089  0.083
```

Figure 13.3: Sample drift plot

compared.

Unfortunately, a plot of the first ratio against the second produces an uninformative scatter diagram, because the divisor $(M_1 - M_2)$ makes the scatter of the points very unequal. If, instead, we plot $(\epsilon_1 - \epsilon_2)$ against $(\epsilon_1 + \epsilon_2)(M_1 - M_2)/(M_1 + M_2)$, the values should all have comparable scatter. This is the plot shown in Fig. 13.3. The line drawn with + signs has the fitted slope and passes through the origin. Again, we expect unit slope if the extinction is changing with time, and zero slope if the instrument is at fault. The actual slope is printed above the plot.

We can obtain additional information from the scatter of the values. The vertical spread of the plotted points is just proportional to a typical residual. The horizontal spread has the same value, but diminished by a typical value of the ratio $(M_1 - M_2)/(M_1 + M_2)$. This average ratio, followed by the ratio of the spreads, is shown at the bottom of the plot. If the errors are dominated by changes in the extinction, we expect the ratio of spreads to be just this average ratio. But if instrumental drift dominates the errors, the spread in the ordinates should be much smaller than the spread in the abscissas, because the correlated part of the residuals cancels out in the differences.

If the drifts are small, so that the data are dominated by random noise, we expect the mean slope to be zero, and the ratio of spreads to be the average ratio of airmass difference to airmass sum, $(M_1 - M_2)/(M_1 + M_2)$. We can therefore take a mean slope greater than 0.5 as evidence in favor of extinction drift, but a smaller slope is only weak evidence of zero drift, as it could be due to noise. Similarly, a spread ratio near zero is strong evidence for zero-point drift, but a value near the mean airmass ratio could be due to noise. On this basis, the sample data shown in Fig. 13.3 show good evidence for extinction drift.

After showing the individual drift plots, the program displays a table that summarizes the results for all bands and nights. In this table, cases that favor extinction drift are marked "E" and those that favor zero-point drift, "Z". Lower-case letters indicate weaker evidence, and a dash indicates nights with too few data to make a useful plot.

At this point, you will be asked to choose whether to solve for time-dependent extinction, time-dependent zero points, neither, or both. In general, you should choose the situation that seems most typical; help is available if you want it. It is usually not wise to try to solve for both kinds of drift; unless the data are very well distributed, one usually finds a strong anti-correlation of the derived drifts in extinction and zero-point. At present, only linear drifts with time are modelled; but this should suffice for most good nights.

If you choose to change the model (i.e., either add or remove a time-dependent term, compared to what was previously used), the program loops back to the beginning of the solution, and repeats its work, using the new model. Initially, solutions are always done without time-dependent terms. This strategy lets you choose an optimal set of extinction stars before trying to solve for any time-dependent terms, which are usually rather weakly determined. If you make no changes in the time-dependence of the model, the extinction solution is finished at this point.

If you included standard-star values in the extinction solution, the transformation coefficients were included in the parameter adjustments. Then full-screen plots of the transformation residuals are shown, followed by residuals plotted as a linearity check. If you did not include standard-star values, the linearity check is plotted, and the program then determines the transformation separately.

Estimated parameters

After the last residual plot, the final parameter values (and their errors) are printed, including the magnitudes and colors of the individual stars in the extinction solution. The errors for the color indices take account of the covariances between the magnitudes. This is followed by a table of the individual residuals, which includes the symbols used to identify different stars in the residual plots.

Individual data

Finally, the individual observations themselves are reduced, using the parameters from the extinction solution. This is a little tricky, as one needs color values to include the color terms in both extinction and transformation equations. If all colors are observed simultaneously, as is done with multichannel spectrometers, these colors can easily be extracted from the neighboring data. But a filter photometer or CCD must observe passbands sequentially. Again, this may not be a problem, if the star is constant in light.

However, for variable stars, the brightness is changing in all bands, and it is difficult to define the correct color to use in the reduction. In principle, one should construct the full light curve of the star, and then interpolate the colors to the time at which each passband was observed. In practice, a simpler compromise is used: the colors are simply interpolated linearly in time, if there are earlier and later observations of the star on the same night. For the first and last observations of a star on a given night, the nearest observations (in time) in the adjoining bands are used. In every case, the values of the colors used to reduce each magnitude are shown on the output.

Every observation is given individually, including observations of standard and extinction stars, together with the U.T. day fraction and heliocentric Julian Date. These data are of interest for standard and extinction stars if they later turn out to be variable. Note that the values of stellar parameters adopted in the extinction solution are to be preferred to values obtained by averaging these individual observations that have been corrected for extinction and transformation.

NOTE: Because colors are needed to transform the observations, first to outside the atmosphere and then to a standard system, only stars that were observed in all passbands on a given night can be reduced in this section of the program. This is in contrast to the treatment of constant stars in the general solution, where even single-passband measurements are useful, provided that (a) every star was observed at least once during the run in every passband; and (b) every night has some observations in all passbands.

That is, the extinction solution can in principle produce results if we observe extinction star 1 only in passband A and star 2 in passband B on night 1, and star 1 only in passband B and star 2 in passband A on night 2. This fulfills the requirements for data in both bands for each night, and for each star. Even if the passbands are linked by color terms in the extinction, the solution is possible, because we do obtain mean colors for each star. However, none of these data could be reduced individually, as there are no observations of either star in both bands on the same night.

Logfile output

All the on-line output goes into the MIDAS logfile. Normally, you would print this out after a reduction run, to have a record of what you have done. This may run a hundred or more pages in length, so you may want to use a reduced-size or "two-up" option in printing, if you have it available. The final values are also stored in a table named results.tbl, which can be incorporated in the Archive (using a procedure yet to be written!) You can extract values from this table to make tables or plots for publication.

To avoid cluttering up your reduction log with other MIDAS output, you can run the `DELETE/LOGFILE` command *before* starting the reductions. Note that the logfile (normally kept in your `$HOME/midwork` directory) is a flat ascii file that can be edited to remove unwanted matter before printing.

Table file output

The individual data, reduced as described above, are put into a simple output table file named "results.tbl". Its columns give both MJD and heliocentric Julian Date (minus 2400000); the name of the object; the band; and the corresponding standard magnitude. This makes the results of the reductions available for further processing with MIDAS utilities.

For example, although the table entries are written in the same order they are printed in the logfile (star by star, so that the necessary color data are adjacent), they can be rearranged into chronological order, or by passband, with the `SORT/TABLE` command. Columns can be extracted to a new table, using `PROJECT/TABLE`. You could interpolate the values of a comparison star to every time in the table by using `FIT/TABLE` or `REGRESS/TABLE`, and then use `COMPUTE/TABLE` to subtract these from the values for a variable star to produce differential photometry.

Additional columns could be transferred from the original data to the new table, using `PROJECT/TABLE` and `JOIN/TABLE` or `MERGE/TABLE`, so that correlations and effects not looked for by the reduction program can be detected. In general, because the PEPSYS package makes heavy use of table files, the user should try to become familiar with the utilities MIDAS provides to deal with them.

13.5.8 Interpreting the output

When you look over the printed output, pay particular attention to the following:

1. Rejected stars: These are marked with an asterisk (*) in the right margin when the residuals are listed. An occasional reject is normal; but clumps of rejected observations are not. If *every* observation of a standard star is rejected, it may be misidentified. If the standard values of a standard star are all rejected, there may be a catalog (or copying) error; or the wrong star may have been observed.

Abnormally faint observations of program stars are often due to the dome being in the way. Be careful to check the dome slit while observing, if it is not controlled automatically. Another cause of abnormally faint star readings is confusion of star and sky measurements (see item 4 below).

2. Reading plots: Remember the conventions used in low-resolution plotting: the $ symbol marks overlapping points; `^` and `v` mark points beyond the upper and lower edges of the plot (think of them as arrowheads). Some plots have fitted lines indicated by a series of dashes. The residual plots identify different stars with different symbols; these are given in the tables of residuals.

3. Trends in residual plots: Similar trends in the run of residuals with time in different bands usually indicate either instrumental instability (zero-point drift) or varying extinction. Instrumental drifts are usually a function of temperature and/or relative humidity; but a bad high-voltage supply can produce irregular variations. Extinction variations are usually larger at shorter wavelengths, and may show short-lived dips as wisps of cirrus cross the sky.

4. Observations with negative intensities: These can be caused by star observations

misidentified as sky, or vice versa. Check your data very carefully for errors. Ordinarily, sky data should be much fainter than the star measurements, for stars brighter than about magnitude 15. If you have fainter program stars than this, use a CCD to obtain the considerable benefits of simultaneous measurements of star and sky. The sky is considerably brighter in the infrared, or during auroral displays. Here, large fluctuations in sky brightness can occur, occasionally producing negative star intensities after sky subtraction. If your sky fluctuations are appreciable, compared to your faintest program star, be sure to chop back and forth quickly between star and sky to subtract the fluctuations accurately.

13.5.9 Special problems

Missing bands

Because a full set of colors is required to account for color terms, it is not possible to reduce data for stars that have not been observed in all filters. If you have some stars that were observed in only a subset of bands, you should extract from the entire table of observations just the subset of bands, and run the whole set of stars in this band subset to obtain reduced values for the stars with incomplete observations. Then make a second set of data from which the incomplete observations are removed, and reduce these to get good values for the stars with complete observations. Be aware that the reduced values will differ systematically in the two subsets.

For example, some stars might be observed only in B and V in a run when most stars were observed in the full UBV set of bands. You would then do two separate reductions: one for all stars, using only the B and V data, and one with only the stars having full UBV data. Notice that the B and V values for the stars in common will differ slightly in the two solutions. The values from the full UBV solution should be more accurate; but you should not intermix them with values from the BV solution. If homogeneity is more important than accuracy, you could try adopting the B and V values for all stars from the BV solution, and the U-B colors from the full solution. The B's in the B-V colors then differ from the B's in the U-B colors; but this is basically what Johnson did in setting up the system. To avoid the problem, make sure you observe every star in every band.

Although the reduction program assumes you will observe a contiguous subset of the bands in any standard system, you could still force it to reduce a *non*-contiguous subset by replying `OTHER` when asked for the system name. You would then supply, in order of increasing wavelength, the bands you actually used. However, you would also have to make up a special set of standard-star files, in which the indices skip any missing bands.

Note that standard-star data that are missing some passbands may still be useful. In general, at least one magnitude is required; except for H-Beta standards, stars with only indices and no magnitudes are not useful.

Nonlinearity

Although it is possible to solve for nonlinearity (e.g., dead-time) corrections in both pulse-counting and DC photometry, *extreme* care should be used in doing so. The problem is that the nonlinearity is strongly coupled to other parameters in the solution. A common error is to include standard-star values in the solution; this aliases conformity errors for the 2 or 3 brightest stars into the nonlinearity parameter. Nonsensical values of both the nonlinearity and the transformation parameters are the usual result, accompanied by a misleadingly "good" fit (small residuals, and small standard errors on the coupled parameters).

To determine nonlinearity accurately, a neutral attenuator should be used to observe a considerable number (say, 15 or 20) of the brightest stars. The fainter stars serve to calibrate the attenuator; the brightest stars then determine the nonlinearity, through comparisons between their attenuated and unattenuated observations. A less precise determination of nonlinearity is possible by using the atmospheric extinction as the attenuator; unfortunately, this is not neutral, so the coupling between nonlinearity, extinction, and bandwidth parameters can produce systematic reduction errors. In either case, the transformation solution should be done separately from the extinction-and-nonlinearity fit.

It is particularly dangerous to have a single star much brighter than the rest, as its high leverage on the nonlinearity parameter will guarantee systematic errors. Such a star will stand out as isolated points on the right-hand side of the linearity plots. If the brightest star is extremely red or blue in any color index, the errors will affect the transformations more strongly. Try to find 2 or 3 bright stars of intermediate color within half a magnitude of one another, to determine the nonlinearity.

While the best linearity is obtained with a good DC system, used at small anode currents, an overloaded DC system can be just as nonlinear as an overloaded pulse counter. The best policy is to know and understand your equipment thoroughly, and (if possible) to avoid observing in the range where nonlinearity is known to be a problem.

Special systems

If you want to either plan or reduce observations made in your own non-standard system, both the planning and the reduction programs let you declare the system as `NONE` or `OTHER`. Use `NONE` when you want to work entirely in the instrumental system, and `OTHER` when you want to use a special system of standard stars. In either case, you will need to specify central wavelengths, bandwidths, and the relation between the passband magnitudes and whatever color indices you use.

You can maintain your own files of standard stars for your private system, or for your instrumental system. Sometimes it is convenient to maintain instrumental-system standards, to avoid the information loss of conformity errors, which should be negligible in this case. Instrumental mean values can then be used to determine extinction very accurately.

Marginal nights

Often, one finds that some nights of a run were of marginal photometric quality. There may have been clouds or cirrus visible; or the residuals may simply be anomalously large. How should these marginal nights be treated?

The safest thing is to reduce all the nights of a run together at first. If one or two nights then have a large number of rejected observations, try to decide whether the problem is instrumental or transparency problems. Nights with instrumental problems should probably be rejected completely. Nights with variable transparency may sometimes be salvaged, either by throwing out the worst part of the night, and reducing the rest together with the good data; or by removing a dubious night from the solution, and treating it separately.

In treating a bad night separately, one can either force the extinction coefficients to have particular values, or force the standard and extinction stars to have particular values (by creating a special standard-star file). Much of the time, the best thing to do with bad data is throw them away.

Sky problems

If the program complains that you have negative intensities, this is usually due to misidentifying star observations as sky. It can also be caused by marking sky observations as star data. Check the `STARSKY` column of your data tables.

Another problem with sky occurs when the program cannot find a suitable sky observation to subtract from a star datum. This can occur when different measuring diaphragms have been used. Remember that you cannot correct an observation for sky unless there are sky data taken in the same band through the same aperture.

Finally, as the sky-modelling algorithm does not try to model twilight, it uses the "nearest-neighbor" method to correct star observations made during twilight for sky. You may find that such observations cannot be corrected for sky by this method, if the star was measured during twilight, but the corresponding sky measurement occurs just after the end of evening twilight or just before the start of morning twilight. You can prevent this problem by measuring sky both before and after each star measured during twilight; but you should do this anyway. Note that the planning program tells you when twilight begins and ends.

CCD data

If CCD data are to be reduced, it is **essential** that they all be on the same instrumental system. First, all the data for each night must be reduced with a common average flat field, for a given filter. (It is possible to use a different flat for each night, which will introduce a zero-point shift from one night to the next.)

Second, all the data for a given night must be comparable, to satisfy Steinheil's principle. One can have problems with some image-extraction routines that use PSF fitting. Because of seeing variations during to night — and especially because of the dependence of seeing on airmass — there may be systematic errors introduced by using different point-spread functions on different frames. If a very detailed PSF model is available, so that the whole energy in a star image is well extracted, with very small residuals, one may expect PSF-fitting to work adequately. However, one must be sure that the extracted magnitudes refer to the total *energy* in the image, and are not just scaled to the peak.

If you use a PSF-fitting routine that leaves obvious "blemishes" when the fitted profile is subtracted from the original frame, it is likely that there will be systematic errors that depend on seeing. In turn, this means systematic errors that depend on airmass, which will spoil the determination of extinction coefficients.

In general, the safest approach with CCD data is to simulate "aperture" photometry, as it is often called — just integrate the total signal in a box (round or square) of fixed size centered accurately on each star. This may give larger random errors than PSF-fitting, but smaller systematic errors. This balance between accuracy and precision is a common dilemma in stellar photometry.

Problems with star names

If the star names used in data files are not *exactly* the same character strings as those used in star files, the reduction program will try to make cross-identifications. These are based on heuristic rules that are generally observed in naming stars — for example, many stars have a catalog abbreviation followed by a number, or a Greek letter followed by a constellation abbreviation. The program recognizes some common catalog abbreviations (HR, BS, BD, CD, CPD, HD, NGC), but will guess that two or three letters followed by a number represents such a name. It also can cope with common suffixes like A, B, and AB for multiple stars.

The program applies these rules in attempting to parse name strings containing multiple designations that are not separated by the " `=` " string (surrounded by blanks). If you sometimes write a name with an embedded blank and sometimes without (e.g., "`HR 123`" vs. "`HR123`"), it may be able to identify the two as equivalent, or it may ask for help. It is difficult to write a simple set of foolproof rules for recognizing star names; for example, the program cannot simply squeeze out embedded blanks, as it would then confuse BD +1 2345 with BD +12 345.

Occasionally a typing error can confuse the program, and it will print `Cannot parse:` followed by a name string. For example, spelling errors in constellation abbreviations, Greek letters, or the letter O for a digit 0 can cause such a message. In these cases, the program asks for help, and asks for a replacement name string (which can contain embedded blanks). Afterward, the replacement string will be used instead of the one that caused problems.

This situation can be avoided by making sure that all star names are spelled consistently, and that names in star files agree with names in data files.

13.6 Installation

The first user of PEPSYS at a new site must make sure that the necessary table files have been installed. Also, it may be useful to change some `PARAMETER` statements to suit local circumstances. Most users will never have to worry about these things; but here they are, just in case you need them.

13.6.1 Table files

MIDAS table files are binary files, to save the considerable overhead of converting between ascii and the machine's internal binary representation of numbers. The price to be paid for fast response while reading a table is that tables are not transportable between different machines. This means that the standard tables have to be created on each new machine.

Fortunately, MIDAS also understands FITS formats, which are portable. Therefore, the tables of standard stars (and some other sample tables) are available as portable ascii files, and a script is provided to create the tables. The FITS files are (on a UNIX system) in the directory `$MIDASHOME/calib/raw/pepsys`, which contains several files with the suffix `.mt`. The directory `$MIDASHOME/calib/install` contains scripts for installing such files; the script you want is `pepsys.prg`. The local MIDAS guru should run this script to create the table files in `$MIDASHOME/calib/data/pepsys`. After running the script, please check the newly-created table files to make sure users can read them but not re-write or remove them.

If the "calib" directories and files are not part of your system, they can be obtained by ftp, or (if necessary) on magnetic tape.

13.6.2 Maximum limits

The planning and reduction programs are distributed with array sizes set large enough to let most observers work without difficulty. However, you might need to reduce a very large data set, or to deal with a very large number of passbands (9 is the nominal limit). The programs are all written in FORTRAN, and the dimensions of these arrays are set by `PARAMETER` statements in "include" files. All these short files are in the `$MIDASHOME/$MIDVERS/contrib/pepsys/incl` subdirectory.

The programs check for array overflows as they read data. If you encounter a limit, the program will tell you which `PARAMETER` statement to change. Ask your MIDAS guru to edit the `PARAMETER` statement and recompile the programs. To make sure everything is internally consistent, compile the subroutines *before* compiling the main programs: move to the `pepsys/libsrc` directory and make the subroutines, then go to the `pepsys/src` directory to make the main programs.

The main limitation on what is practical is machine memory. If your machine does not have enough memory to store all the necessary arrays without swapping pages back and forth to disk, the reduction program may run very slowly. (Bear in mind that response on multi-user machines also depends on what else is being run by other users at the same time.) The main problem is holding the matrix of normal equations during the iterations; on the other hand, star catalogs can be made quite large without paying much of a price in terms of performance. (However, if the number of stars exceeds 2048, the binary-search subroutine `NBIN` will need to be modified.)

You can get a rough idea of where you will run into problems by noting that the matrix of normal equations is made of double-precision numbers, and is a square matrix (well... triangular, actually) with as many elements each way as there are parameters to be evaluated. For example, to solve for 120 extinction and standard stars in 4 colors takes

480 parameters; there will be some extinction and transformation coefficients, too, so this problem is about 500 x 500, using a quarter of a million matrix elements. Each element is 8 bytes long, on most systems; so we need roughly 2 MB of memory — no problem these days. Of course, only half the matrix is stored, as it is symmetric. On the other hand, you need space for the right-hand-side vector, and the program as well, not to mention other arrays that are needed. So this rough calculation is good enough for astrophysical accuracy.

On the other hand, if you wanted to do 1000 stars simultaneously in 4 colors, you'd need some 16 million elements of 8 bytes each, or 128 MB of storage. That's probably too big for most machines to handle gracefully. Or, if you wanted to reduce 100 channels of spectrophotometry simultaneously, even 20 stars would give you 2000 parameters (plus 100 extinction coefficients per night!); that would be over 32 MB, and heading for trouble. In this latter case, it would probably make sense to reduce smaller subsets of bands together, to allow more stars and nights to be combined. In general, you should try to reduce at least 4 or 5 nights' data together, to improve the extinction determinations.

If your machine proves to be a bottleneck, you might want to maintain two variants of the reduction program: one small enough to run quickly on ordinary data sets, and a monster version for the occasional monster problem. In this case, the small, fast version would be kept in the regular MIDAS directory tree, and the gigantic version could be owned by the rare user who needs it.

If you find it necessary to increase some of the array parameters, please let the Image Processing Group at ESO know what changes you make. This will allow revised parameters to be distributed in future releases.

13.7 A brief history of PEPSYS

This package is descended from a program originally written for the IBM 704 in the late 1950's [26]. It used all constant stars as extinction stars, combined multiple nights with separate extinction coefficients, and employed King's correct representation of color terms in the extinction. That program was written in assembly language after an initial try in FORTRAN II would not fit into the 8192 words of the machine's memory. Data were packed, two to a word, on magnetic drums. Computers at that time were so small that the program needed every cell of the machine; even so, it had to be split into overlays.

When the 7090 came along, the program was stored on a loadable tape that displaced the FMS operating system from core memory, and pulled the system back in when it finished. This special system tape was labelled PEPSYS (for Photo-Electric Photometry SYStem). This original PEPSYS died when the IBM 709x systems became obsolete.

A second attempt was made on a Cyber 175 many years later. The planning program was added in the 1980's. This was ported to a VAX, and later to other machines (AT&T 3B2 and Sun 4). These ports shook a lot of portability bugs out of the common subroutines. The Manfroid – Heck program [15], and a program used at that time by Peter Stetson were investigated, but found to be too specialized to particular photometric systems and data formats to be generally useful, so a proposal was made to NSF to develop something along the lines of the present system. The proposal was rejected on the grounds that (a) it was too expensive, and (b) such a thing is impossible anyway.

The heart of the present reduction program — the fast matrix inversion using partitioning – was developed during a simulation study in the late 1980's. However, it was used only on phony data, and lacked the extensive user interface of the present version. Some user interface, especially material now embedded in the `MAKE/PHOTOMETER` command, was developed on an Intel 80386 system running SCO Xenix System V.

The current system was sponsored by ESO; I thank Chris Sterken and Preben Grosbøl for setting up my visit to Garching. This version was developed within ESO's Image Processing Group. Although an image-processing system offers the photometrist as many inconveniences to work around as useful tools, it has been possible to hide most of the problems from the user.

13.8 Acknowledgements

Many people have contributed help and advice in developing the current version of PEPSYS. I have learned a lot about photometry over the years from A. W. J. Cousins, whose papers I recommend to every observer. John Menzies communicated the Cousins E-region standards in both UBV and uvby systems. Erik Olsen provided the latest uvby – H-Beta standards, and advice on their use. Harri Lindgren and Petr Harmanec provided much useful discussion of reduction methods. Stan Stefl was the first user of PEPSYS at ESO, and uncovered a number of bugs as well as suggesting helpful features. Chris Sterken tested the reduction program, helped find bugs, and made useful suggestions. Josef Hron carefully read the documentation and suggested numerous improvements. Fionn Murtagh pointed out a useful reference [5]. The members of the IPG all helped me with numerous MIDAS problems. Above all, I must thank Chris Sterken and Jean Manfroid for thinking hard about photometric reductions, and publishing a stimulating series of papers on the subject, as well as their recent textbook [22].

13.9 Summary of PEPSYS commands

- MAKE/HORFORM: Makes a blank form to fill in with horizon data
- MAKE/STARTABLE: Helps you make a *.tbl file for program stars
- MAKE/PHOTOMETER: Makes or displays an instrument-description file.
- MAKE/PLAN: Makes an observing schedule for you
- CONVERT/PHOT: Converts some ascii data files to MIDAS tables.
- REDUCE/PHOT: Reduces the data.

Bibliography

[1] Beckert, D. C., and Newberry, M. V. : 1989, *Pub. A. S. P.*, **101**, 849.

[2] Bessell, M. S. : 1990, *Pub. A. S. P.*, **102**, 1181.

[3] Dennis, J. E., and Schnabel, R. B. : 1983, *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*, Prentice-Hall, pp. 227-228.

[4] Evans, R. D. : 1955, *The Atomic Nucleus*, McGraw-Hill, New York, pp. 785-790.

[5] Feigelson, E. D., and Babu, G. J. : 1992, *Ap. J.*, **397**, 55.

[6] Garstang, R. H. : 1986, *Pub. A. S. P.*, **98**, 364.

[7] Garstang, R. H. : 1989, *Pub. A. S. P.*, **101**, 306.

[8] Garstang, R. H. : 1991, *Pub. A. S. P.*, **103**, 1109.

[9] Gill, P. E., Murray, W., and Wright, M. H. : 1981, *Practical Optimization*, Academic Press, New York, p. 104.

[10] Hardie, R. H., : 1962, in *Astronomical Techniques*, W. A. Hiltner, ed., Univ. of Chicago Press, Chicago, p. 178.

[11] Hall, P. : 1980, *Rates of convergence in the central limit theorem*, Pitman, Boston.

[12] Hampel, F. R., Ronchetti, E. M., Rouseeuw, P. J., and Staehl, W. A. : 1986, *Robust Statistics*, Wiley, New York.

[13] Hoaglin, D. C., Mosteller, F., and Tukey, J. W. : 1983, *Understanding Robust and Exploratory Data Analysis*, Wiley, New York, Chap. 5.

[14] Kahaner, D., Moler, C., and Nash, S. : 1989, *Numerical Methods and Software*, Prentice Hall, Englewood Cliffs, NJ, p. 368.

[15] Manfroid, J., and Heck, A. : 1984, *Astron. Astrophys.*, **132**, 110.

[16] Manfroid, J., and Sterken, C. : 1992, *Astron. Astrophys.*, **258**, 600.

[17] Menzies, J. W., and Laing, J. D. : 1985, M. N., **217**, 563.

[18] Mosteller, F., and Tukey, J. W. : 1977, *Data Analysis and Regression*, Addison-Wesley, Reading.

[19] Popper, D. M. : 1982, *Pub. A. S. P.* **94**, 204.

[20] Press, W. H., Flannery, B. P., Teukolsky, S. A., and Vetterling, W. T. : 1986, *Numerical Recipes*, Cambridge University Press.

[21] Schwarzenberg-Czerny, A. : 1991, *Astron. Astrophys.*, **252**, 425.

[22] Sterken, C., and Manfroid, J. : 1992, *Astronomical Photometry: a Guide*, Kluwer, Dordrecht.

[23] Tukey, J. W. : 1960, in *Contributions to Probability and Statistics*, I. Olkin, ed., Stanford Univ. Press, p. 448.

[24] van de Hulst, H. C. : 1980, *Multiple Light Scattering*, Academic, New York.

[25] Walker, M. F. : 1987, in *Identification, Optimization, and Protection of Optical Telescope Sites*, R. L. Millis et al., eds., Lowell Observatory, Flagstaff, p. 128.

[26] Young, A. T., and Irvine, W. M. : 1967, *Astron. J.*, **72**, 945.

[27] Young, A. T. : 1974, *Methods of Experimental Physics*, **12**, Part A, Astrophysics: Optical and Infrared, ed. Carleton, N., Academic, New York, Chap. 3.

[28] Young, A. T., et al. : 1991, P̂ub. A.S.P., **103**, 221.

[29] Young, A. T. : 1992, *Astron. Astrophys.*, **257**, 366.

[30] Young, A. T. : 1992, "High-Precision Photometry", in *Automated Telescopes for Photometry and Imaging*, (A. S. P. Conference Series, vol. 28) S. J. Adelman, R. J. Dukes, and C. J. Adelman, eds. (Astronomical Soc. of the Pacific, San Francisco, 1992) pp. 73-89.

Chapter 14

The Wavelet Transform

14.1 Introduction

The Fourier transform is a tool widely used for many scientific purposes, but it is well suited only to the study of stationary signals where all frequencies have an infinite coherence time. The Fourier analysis brings only global information which is not sufficient to detect compact patterns. Gabor [13] introduced a local Fourier analysis, taking into account a sliding window, leading to a time frequency-analysis. This method is only applicable to situations where the coherence time is independent of the frequency. This is the case for instance for singing signals which have their coherence time determined by the geometry of the oral cavity. Morlet introduced the Wavelet Transform in order to have a coherence time proportional to the period [26].

Extensive literature exists on the Wavelet Transform and its applications ([7, 9, 27, 29, 28, 31]). We summarize the main features here.

14.2 The continuous wavelet transform

The Morlet-Grossmann definition of the continuous wavelet transform [17] for a $1D$ signal $f(x) \in L^2(R)$ is:

$$W(a,b) = \frac{1}{\sqrt{a}} \int_{-\infty}^{+\infty} f(x) \psi^*(\frac{x-b}{a}) dx \tag{14.1}$$

where z^* denotes the complex conjugate of z, $\psi^*(x)$ is the analyzing wavelet, a (> 0) is the scale parameter and b is the position parameter. The transform is characterized by the following three properties:

1. it is a linear transformation,

2. it is covariant under translations:

$$f(x) \longrightarrow f(x-u) \qquad W(a,b) \longrightarrow W(a,b-u) \tag{14.2}$$

3. it is covariant under dilations:

$$f(x) \longrightarrow f(sx) \qquad W(a,b) \longrightarrow s^{-\frac{1}{2}} W(sa, sb) \tag{14.3}$$

The last property makes the wavelet transform very suitable for analyzing hierarchical structures. It is like a mathematical microscope with properties that do not depend on the magnification.

In Fourier space, we have:

$$\hat{W}(a,\nu) = \sqrt{a}\hat{f}(\nu)\hat{\psi}^*(a\nu) \tag{14.4}$$

When the scale a varies, the filter $\hat{\psi}^*(a\nu)$ is only reduced or dilated while keeping the same pattern.

Now consider a function $W(a,b)$ which is the wavelet transform of a given function $f(x)$. It has been shown [17, 19] that $f(x)$ can be restored using the formula:

$$f(x) = \frac{1}{C_\chi}\int_0^{+\infty}\int_{-\infty}^{+\infty}\frac{1}{\sqrt{a}}W(a,b)\chi(\frac{x-b}{a})\frac{da.db}{a^2} \tag{14.5}$$

where:

$$C_\chi = \int_0^{+\infty}\frac{\hat{\psi}^*(\nu)\hat{\chi}(\nu)}{\nu}d\nu = \int_{-\infty}^{0}\frac{\hat{\psi}^*(\nu)\hat{\chi}(\nu)}{\nu}d\nu \tag{14.6}$$

Generally $\chi(x) = \psi(x)$, but other choices can enhance certain features for some applications.

The reconstruction is only available if C_χ is defined (admissibility condition). In the case of $\chi(x) = \psi(x)$, this condition implies $\hat{\psi}(0) = 0$, *i.e.* the mean of the wavelet function is 0.

14.3 Examples of Wavelets

14.3.1 Morlet's Wavelet

The wavelet defined by Morlet is [16]:

$$\hat{g}(\omega) = e^{-2\pi^2(\nu-\nu_0)^2} \tag{14.7}$$

it is a complex wavelet which can be decomposed in two parts, one for the real part, and the other for the imaginary part.

$$\begin{aligned} g_r(x) &= \frac{1}{\sqrt{2\pi}}e^{-\frac{x^2}{2}}\cos(2\pi\nu_0 x) \\ g_i(x) &= \frac{1}{\sqrt{2\pi}}e^{-\frac{x^2}{2}}\sin(2\pi\nu_0 x) \end{aligned}$$

where ν_0 is a constant. The admissibility condition is verified only if $\nu_0 > 0.8$. Figure 14.1 shows these two functions.

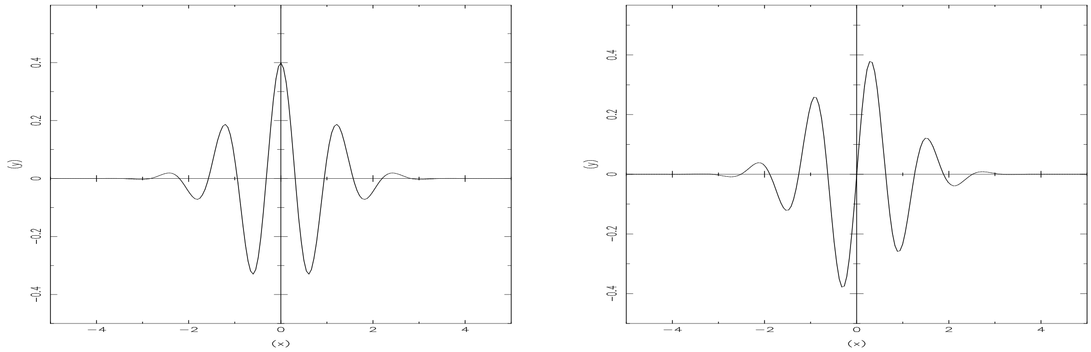

Figure 14.1: Morlet's wavelet: real part at left and imaginary part at right.

Mexican Hat

The Mexican hat defined by Murenzi [30] is:

$$g(x) = (1 - x^2)e^{-\frac{1}{2}x^2} \tag{14.8}$$

it is the second derivative of a Gaussian (see figure 14.2).

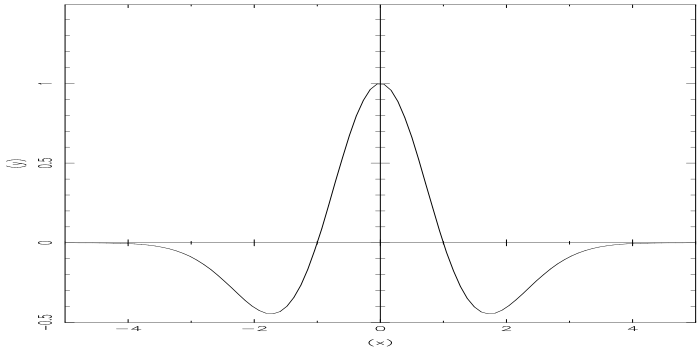

Figure 14.2: Mexican Hat

14.4 The discrete wavelet transform

14.4.1 Introduction

For processing classical images the sampling is made in accordance with Shannon's [32] well-known theorem. The discrete wavelet transform (DWT) can be derived from this theorem if we process a signal which has a cut-off frequency. For such images the frequency band is always limited by the size of the camera aperture.

A digital analysis is provided by the discretisation of formula 14.1, with some simple considerations on the modification of the wavelet pattern by dilation. Usually the wavelet function $\psi^*(x)$ has no cut-off frequency and it is necessary to suppress the values outside the frequency band in order to avoid aliasing effects. We can work in Fourier space, computing the transform scale by scale. The number of elements for a scale can be reduced, if the frequency bandwidth is also reduced. This is possible only for a wavelet which also has a cut-off frequency. The decomposition proposed by Littlewood and Paley [22] provides a very nice illustration of the reduction of elements scale by scale. This decomposition is based on an iterative dichotomy of the frequency band. The associated wavelet is well localized in Fourier space where it allows a reasonable analysis to be made although not in the original space. The search for a discrete transform which is well localized in both spaces leads to multiresolution analysis.

14.4.2 Multiresolution Analysis

Multiresolution analysis [25] results from the embedded subsets generated by the interpolations at different scales.

A function $f(x)$ is projected at each step j onto the subset V_j. This projection is defined by the scalar product $c_j(k)$ of $f(x)$ with the scaling function $\phi(x)$ which is dilated and translated:

$$c_j(k) = < f(x), 2^{-j}\phi(2^{-j}x - k) > \tag{14.9}$$

As $\phi(x)$ is a scaling function which has the property:

$$\frac{1}{2}\phi(\frac{x}{2}) = \sum_n h(n)\phi(x-n) \tag{14.10}$$

or

$$\hat{\phi}(2\nu) = \hat{h}(\nu)\hat{\phi}(\nu) \tag{14.11}$$

where $\hat{h}(\nu)$ is the Fourier transform of the function $\sum_n h(n)\delta(x-n)$. We get:

$$\hat{h}(\nu) = \sum_n h(n)e^{-2\pi n\nu} \tag{14.12}$$

Equation 14.10 permits to compute directly the set $c_{j+1}(k)$ from $c_j(k)$. If we start from the set $c_0(k)$ we compute all the sets $c_j(k)$, with $j > 0$, without directly computing any other scalar product:

$$c_{j+1}(k) = \sum_n h(n-2k)c_j(n) \tag{14.13}$$

At each step, the number of scalar products is divided by 2. Step by step the signal is smoothed and information is lost. The remaining information can be restored using the complementary subspace W_{j+1} of V_{j+1} in V_j. This subspace can be generated by a suitable wavelet function $\psi(x)$ with translation and dilation.

$$\frac{1}{2}\psi(\frac{x}{2}) = \sum_n g(n)\phi(x-n) \tag{14.14}$$

or

$$\hat{\psi}(2\nu) = \hat{g}(\nu)\hat{\phi}(\nu) \tag{14.15}$$

We compute the scalar products $< f(x), 2^{-(j+1)}\psi(2^{-(j+1)}x - k) >$ with:

$$w_{j+1}(k) = \sum_n g(n-2k)c_j(n) \tag{14.16}$$

With this analysis, we have built the first part of a filter bank [34]. In order to restore the original data, Mallat uses the properties of orthogonal wavelets, but the theory has been generalized to a large class of filters [8] by introducing two other filters $\tilde{h}$ and $\tilde{g}$ named conjugated to h and g. The restoration is performed with:

$$c_j(k) = 2\sum_l [c_{j+1}(l)\tilde{h}(k+2l) + w_{j+1}(l)\tilde{g}(k+2l)] \tag{14.17}$$

In order to get an exact restoration, two conditions are required for the conjugate filters:

- *Dealiasing condition*:

$$\hat{h}(\nu+\frac{1}{2})\hat{\tilde{h}}(\nu) + \hat{g}(\nu+\frac{1}{2})\hat{\tilde{g}}(\nu) = 0 \tag{14.18}$$

- *Exact restoration:*

$$\hat{h}(\nu)\hat{\tilde{h}}(\nu) + \hat{g}(\nu)\hat{\tilde{g}}(\nu) = 1 \tag{14.19}$$

In the decomposition, the function is successively convolved with the two filters H (low frequencies) and G (high frequencies). Each resulting function is decimated by suppression of one sample out of two. The high frequency signal is left, and we iterate with the low frequency signal (upper part of figure 14.3). In the reconstruction, we restore the sampling by inserting a 0 between each sample, then we convolve with the conjugate filters $\tilde{H}$ and $\tilde{G}$, we add the resulting functions and we multiply the result by 2. We iterate up to the smallest scale (lower part of figure 14.3).

Orthogonal wavelets correspond to the restricted case where:

$$\begin{aligned} \hat{g}(\nu) &= e^{-2\pi\nu}\hat{h}^*(\nu+\frac{1}{2}) && (14.20)\\ \hat{\tilde{h}}(\nu) &= \hat{h}^*(\nu) && (14.21)\\ \hat{\tilde{g}}(\nu) &= \hat{g}^*(\nu) && (14.22) \end{aligned}$$

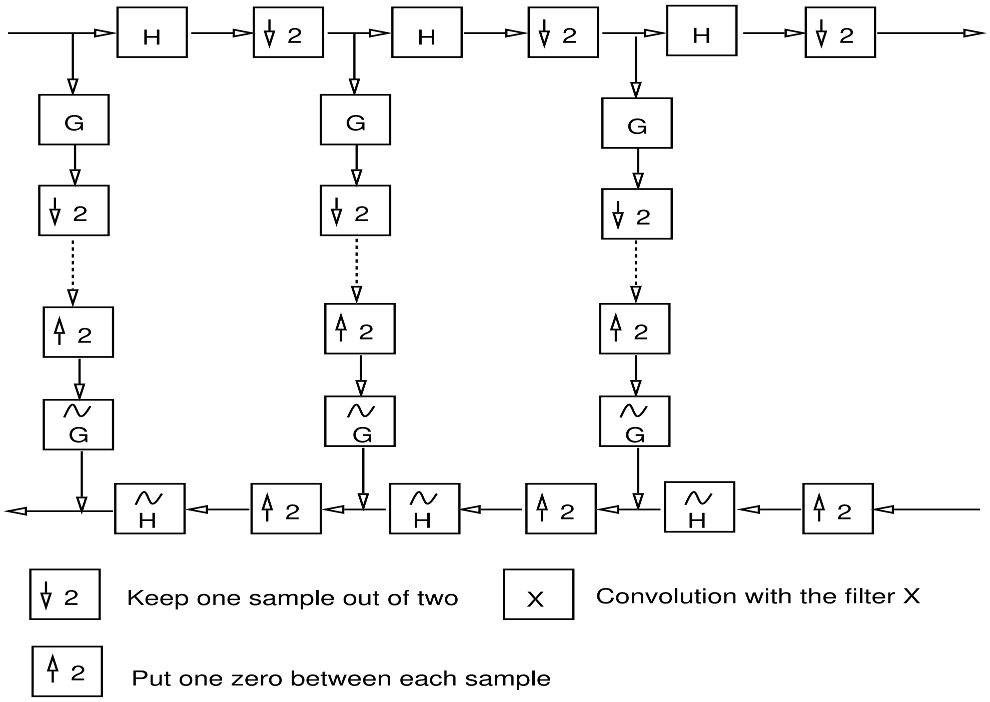

Figure 14.3: The filter bank associated with the multiresolution analysis

and

$$| \hat{h}(\nu) |^2 + | \hat{h}(\nu + \frac{1}{2} |^2 = 1 \tag{14.23}$$

We can easily see that this set satisfies the two basic relations 14.18 and 14.19. Daubechies wavelets are the only compact solutions. For biorthogonal wavelets [8] we have the relations:

$$\hat{g}(\nu) = e^{-2\pi\nu}\hat{\tilde{h}}^*(\nu + \frac{1}{2}) \tag{14.24}$$

$$\hat{\tilde{g}}(\nu) = e^{2\pi\nu}\hat{h}^*(\nu + \frac{1}{2}) \tag{14.25}$$

and

$$\hat{h}(\nu)\hat{\tilde{h}}(\nu) + \hat{h}^*(\nu + \frac{1}{2})\hat{\tilde{h}}^*(\nu + \frac{1}{2}) = 1 \tag{14.26}$$

We also satisfy relations 14.18 and 14.19. A large class of compact wavelet functions can be derived. Many sets of filters were proposed, especially for coding. It was shown [9] that the choice of these filters must be guided by the regularity of the scaling and the wavelet functions. The complexity is proportional to N. The algorithm provides a pyramid of N elements.

The 2D algorithm is based on separate variables leading to prioritizing of x and y directions. The scaling function is defined by:

$$\phi(x, y) = \phi(x)\phi(y) \tag{14.27}$$

The passage from a resolution to the next one is done by:

$$f_{j+1}(k_x, k_y) = \sum_{l_x=-\infty}^{+\infty} \sum_{l_y=-\infty}^{+\infty} h(l_x - 2k_x)h(l_y - 2k_y)f_j(l_x, l_y) \tag{14.28}$$

The detail signal is obtained from three wavelets:

- a vertical wavelet :
$$\psi^1(x, y) = \phi(x)\psi(y)$$
- a horizontal wavelet:
$$\psi^2(x, y) = \psi(x)\phi(y)$$
- a diagonal wavelet:
$$\psi^3(x, y) = \psi(x)\psi(y)$$

which leads to three sub-images:

$$C^1_{j+1}(k_x, k_y) = \sum_{l_x=-\infty}^{+\infty} \sum_{l_y=-\infty}^{+\infty} g(l_x - 2k_x)h(l_y - 2k_y)f_j(l_x, l_y)$$

$$C^2_{j+1}(k_x, k_y) = \sum_{l_x=-\infty}^{+\infty} \sum_{l_y=-\infty}^{+\infty} h(l_x - 2k_x)g(l_y - 2k_y)f_j(l_x, l_y)$$

$$C^3_{j+1}(k_x, k_y) = \sum_{l_x=-\infty}^{+\infty} \sum_{l_y=-\infty}^{+\infty} g(l_x - 2k_x)g(l_y - 2k_y)f_j(l_x, l_y)$$

The wavelet transform can be interpreted as the decomposition on frequency sets with a spatial orientation.

14.4.3 The *à trous* algorithm

The discrete approach of the wavelet transform can be done with the special version of the so-called *à trous* algorithm (with holes) [20, 33]. One assumes that the sampled data $\{c_0(k)\}$ are the scalar products at pixels k of the function $f(x)$ with a scaling function $\phi(x)$ which corresponds to a low pass filter.

The first filtering is then performed by a twice magnified scale leading to the $\{c_1(k)\}$ set. The signal difference $\{c_0(k)\} - \{c_1(k)\}$ contains the information between these two scales and is the discrete set associated with the wavelet transform corresponding to $\phi(x)$. The associated wavelet is therefore $\psi(x)$.

$$\frac{1}{2}\psi(\frac{x}{2}) = \phi(x) - \frac{1}{2}\phi(\frac{x}{2}) \tag{14.29}$$

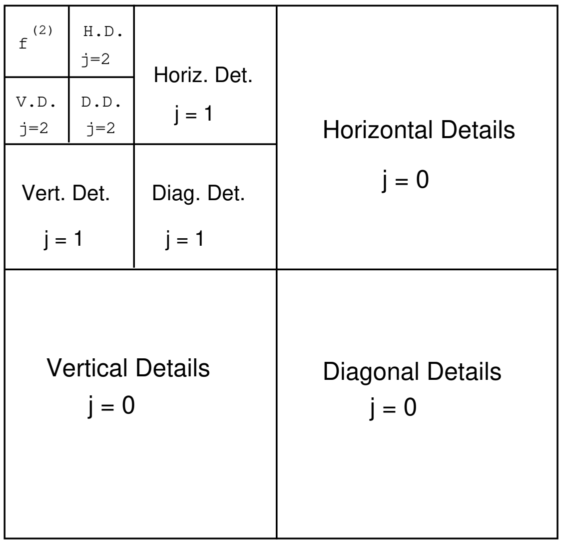

Figure 14.4: Wavelet transform representation of an image

The distance between samples increasing by a factor 2 from the scale $(i-1)$ $(i>0)$ to the next one, $c_i(k)$ is given by:

$$c_i(k) = \sum_l h(l) c_{i-1}(k + 2^{i-1} l) \tag{14.30}$$

and the discrete wavelet transform $w_i(k)$ by:

$$w_i(k) = c_{i-1}(k) - c_i(k) \tag{14.31}$$

The coefficients $\{h(k)\}$ derive from the scaling function $\phi(x)$:

$$\frac{1}{2}\phi(\frac{x}{2}) = \sum_l h(l)\phi(x-l) \tag{14.32}$$

The algorithm allowing one to rebuild the data frame is evident: the last smoothed array c_{n_p} is added to all the differences w_i.

$$c_0(k) = c_{n_p}(k) \sum_{j=1}^{n_p} w_j(k) \tag{14.33}$$

If we choose the linear interpolation for the scaling function ϕ (see figure 14.5):

$$\begin{array}{ll} \phi(x) = 1- \mid x \mid & \text{if } x \in [-1,1] \\ \phi(x) = 0 & \text{if } x \notin [-1,1] \end{array}$$

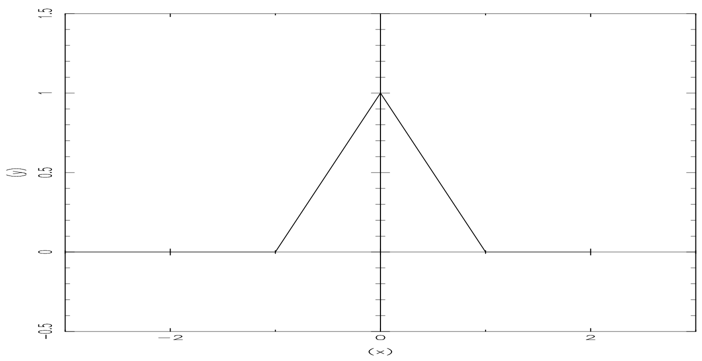

Figure 14.5: linear interpolation ϕ

we have:

$$\frac{1}{2}\phi(\frac{x}{2}) = \frac{1}{4}\phi(x+1) + \frac{1}{2}\phi(x) + \frac{1}{4}\phi(x-1) \tag{14.34}$$

c_1 is obtained by:

$$c_1(k) = \frac{1}{4}c_0(k-1) + \frac{1}{2}c_0(k) + \frac{1}{4}c_0(k+1) \tag{14.35}$$

and c_{j+1} is obtained from c_j by:

$$c_{j+1}(k) = \frac{1}{4}c_j(k-2^j) + \frac{1}{2}c_j(k) + \frac{1}{4}c_j(k+2^j) \tag{14.36}$$

The figure 14.6 shows the wavelet associated to the scaling function.

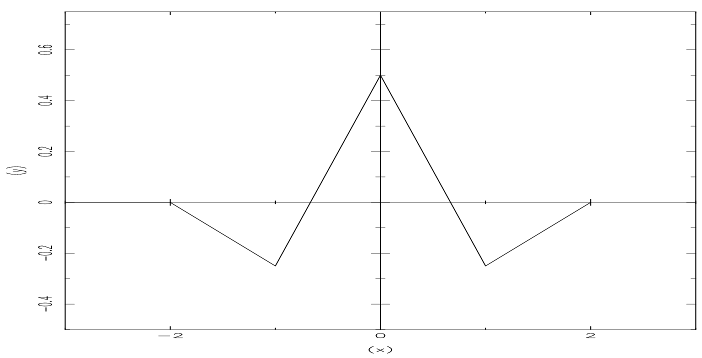

Figure 14.6: Wavelet ψ

The wavelet coefficients at the scale j are:

$$C_{j+1}(k) = -\frac{1}{4}c_j(k-2^j) + \frac{1}{2}c_j(k) - \frac{1}{4}c_j(k+2^j) \tag{14.37}$$

The above *à trous algorithm* is easily extensible to the two dimensional space. This leads to a convolution with a mask of 3×3 pixels for the wavelet connected to linear interpolation. The coefficents of the mask are:

$$\begin{pmatrix} \frac{1}{16} & \frac{1}{8} & \frac{1}{16} \\ \frac{1}{8} & \frac{1}{4} & \frac{1}{8} \\ \frac{1}{16} & \frac{1}{8} & \frac{1}{16} \end{pmatrix}$$

At each scale j, we obtain a set $\{w_j(k,l)\}$ (we will call it wavelet plane in the following), which has the same number of pixels as the image.

If we choose a B_3-spline for the scaling function, the coefficients of the convolution mask in one dimension are $(\frac{1}{16}, \frac{1}{4}, \frac{3}{8}, \frac{1}{4}, \frac{1}{16})$, and in two dimensions:

$$\begin{pmatrix} \frac{1}{256} & \frac{1}{64} & \frac{3}{128} & \frac{1}{64} & \frac{1}{256} \\ \frac{1}{64} & \frac{1}{16} & \frac{3}{32} & \frac{1}{16} & \frac{1}{64} \\ \frac{3}{128} & \frac{3}{32} & \frac{9}{64} & \frac{3}{32} & \frac{3}{128} \\ \frac{1}{64} & \frac{1}{16} & \frac{3}{32} & \frac{1}{16} & \frac{1}{64} \\ \frac{1}{256} & \frac{1}{64} & \frac{3}{128} & \frac{1}{64} & \frac{1}{256} \end{pmatrix}$$

14.4.4 Pyramidal Algorithm

The Laplacian Pyramid

The Laplacian Pyramid has been developed by Burt and Adelson in 1981 [4] in order to compress images. After the filtering, only one sample out of two is kept. The number of pixels decreases by a factor two at each scale.

The convolution is done with the filter h by keeping one sample out of two (see figure 14.7):

$$c_{j+1}(k) = \sum_l h(l-2k)c_j(l) \tag{14.38}$$

To reconstruct c_j from c_{j+1}, we need to calculate the difference signal w_{j+1}.

$$w_{j+1}(k) = c_j(k) - \tilde{c}_j(k) \tag{14.39}$$

where $\tilde{c}_j$ is the signal reconstructed by the following operation (see figure 14.8):

$$\tilde{c}_j(k) = 2\sum_l h(k-2l)c_j(k) \tag{14.40}$$

In two dimensions, the method is similar. The convolution is done by keeping one sample out of two in the two directions. We have:

$$c_{j+1}(n,m) = \sum_{k,l} h(k-2n, l-2m)c_j(k,l) \tag{14.41}$$

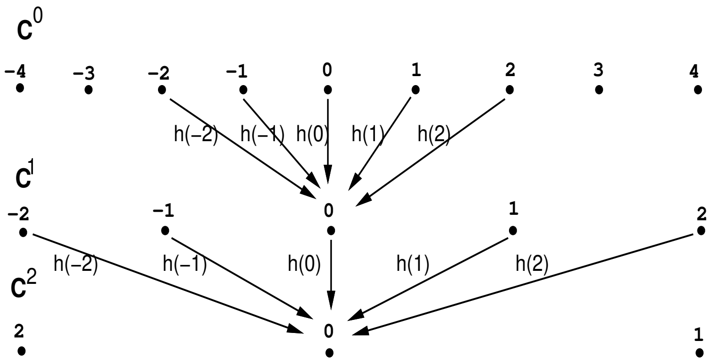

Figure 14.7: Passage from c_0 to c_1, and from c_1 to c_2.

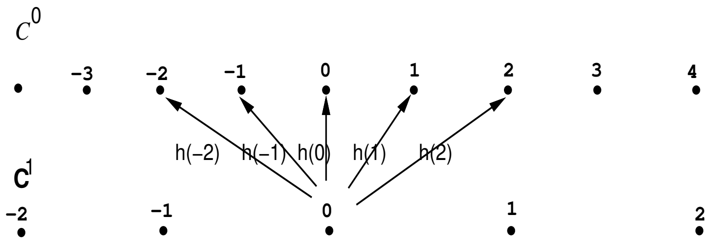

Figure 14.8: Passage from C_1 to C_0.

and $\tilde{c}_j$ is:

$$\tilde{c}_j(n,m) = 2\sum_{k,l} h(n-2l, m-2l)c_{j+1}(k,l) \tag{14.42}$$

The number of samples is divided by four. If the image size is $N \times N$, then the pyramid size is $\frac{4}{3}N^2$. We get a pyramidal structure (see figure 14.9).

The laplacian pyramid leads to an analysis with four wavelets [3] and there is no invariance to translation.

Pyramidal Algorithm with one Wavelet

To modify the previous algorithm in order to have an isotropic wavelet transform, we compute the difference signal by:

$$w_{j+1}(k) = c_j(k) - \tilde{c}_j(k) \tag{14.43}$$

but $\tilde{c}_j$ is computed without reducing the number of samples:

$$\tilde{c}_j(k) = \sum_l h(k-l)c_j(k) \tag{14.44}$$

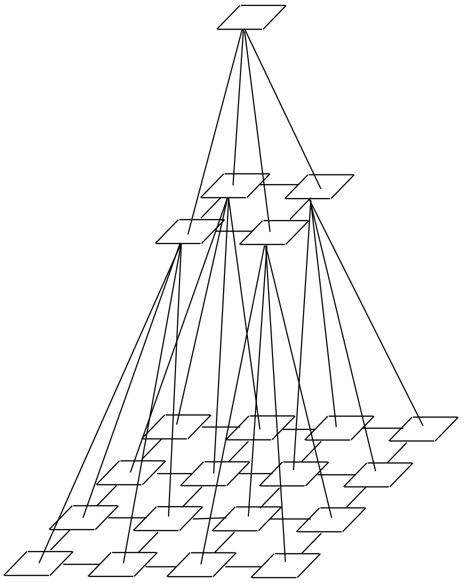

Figure 14.9: Pyramidal Structure

and c_{j+1} is obtained by:

$$c_{j+1}(k) = \sum_{l} h(l-2k)c_j(l) \tag{14.45}$$

The reconstruction method is the same as with the laplacian pyramid, but the reconstruction is not exact. However, the exact reconstruction can be performed by an iterative algorithm. If P_0 represents the wavelet coefficients pyramid, we look for an image such that the wavelet transform of this image gives P_0. Van Cittert's iterative algorithm gives:

$$P_{n+1} = P_0 + P_n - R(P_n) \tag{14.46}$$

where

- P_0 is the pyramid to be reconstructed
- P_n is the pyramid after n iterations
- R is an operator which consists in doing a reconstruction followed by a wavelet transform.

The solution is obtained by reconstructing the pyramid P_n.

We need no more than 7 or 8 iterations to converge. Another way to have a pyramidal wavelet transform with an isotropic wavelet is to use a scaling function with a cut-off frequency.

14.4.5 Multiresolution with scaling functions with a frequency cut-off

The Wavelet transform using the Fourier transform

We start with the set of scalar products $c_0(k) =< f(x), \phi(x-k) >$. If $\phi(x)$ has a cut-off frequency $\nu_c \leq \frac{1}{2}$ [35, 36, 37, 38], the data are correctly sampled. The data at the resolution $j = 1$ are:

$$c_1(k) =< f(x), \frac{1}{2}\phi(\frac{x}{2} - k) > \tag{14.47}$$

and we can compute the set $c_1(k)$ from $c_0(k)$ with a discrete filter $\hat{h}(\nu)$:

$$\hat{h}(\nu) = \begin{cases} \frac{\hat{\phi}(2\nu)}{\hat{\phi}(\nu)} & \text{if } \mid \nu \mid < \nu_c \\ 0 & \text{if } \nu_c \leq \mid \nu \mid < \frac{1}{2} \end{cases} \tag{14.48}$$

and

$$\forall \nu, \forall n \quad \hat{h}(\nu + n) = \hat{h}(\nu) \tag{14.49}$$

where n is an integer. So:

$$\hat{c}_{j+1}(\nu) = \hat{c}_j(\nu)\hat{h}(2^j\nu) \tag{14.50}$$

The cut-off frequency is reduced by a factor 2 at each step, allowing a reduction of the number of samples by this factor.

The wavelet coefficients at the scale $j+1$ are:

$$w_{j+1}(k) =< f(x), 2^{-(j+1)}\psi(2^{-(j+1)}x - k) > \tag{14.51}$$

and they can be computed directly from $c_j(k)$ by:

$$\hat{w}_{j+1}(\nu) = \hat{c}_j(\nu)\hat{g}(2^j\nu) \tag{14.52}$$

where g is the following discrete filter:

$$\hat{g}(\nu) = \begin{cases} \frac{\hat{\psi}(2\nu)}{\hat{\phi}(\nu)} & \text{if } \mid \nu \mid < \nu_c \\ 1 & \text{if } \nu_c \leq \mid \nu \mid < \frac{1}{2} \end{cases} \tag{14.53}$$

and

$$\forall \nu, \forall n \quad \hat{g}(\nu + n) = \hat{g}(\nu) \tag{14.54}$$

The frequency band is also reduced by a factor 2 at each step. Applying the sampling theorem, we can build a pyramid of $N + \frac{N}{2} + \ldots + 1 = 2N$ elements. For an image analysis the number of elements is $\frac{4}{3}N^2$. The overdetermination is not very high.

The B-spline functions are compact in this directe space. They correspond to the autoconvolution of a square function. In the Fourier space we have:

$$\hat{B}_l(\nu) = \frac{\sin \pi\nu}{\pi\nu}^{l+1} \tag{14.55}$$

$B_3(x)$ is a set of 4 polynomials of degree 3. We choose the scaling function $\phi(\nu)$ which has a $B_3(x)$ profile in the Fourier space:

$$\hat{\phi}(\nu) = \frac{3}{2} B_3(4\nu) \tag{14.56}$$

In the direct space we get:

$$\phi(x) = \frac{3}{8}\left[\frac{\sin\frac{\pi x}{4}}{\frac{\pi x}{4}}\right]^4 \tag{14.57}$$

This function is quite similar to a Gaussian one and converges rapidly to 0. For 2-D the scaling function is defined by $\hat{\phi}(u,v) = \frac{3}{2}B_3(4r)$, with $r = \sqrt{(u^2+v^2)}$. It is an isotropic function.

The wavelet transform algorithm with n_p scales is the following one:

1. We start with a B3-Spline scaling function and we derive ψ, h and g numerically.
2. We compute the corresponding image FFT. We name T_0 the resulting complex array;
3. We set j to 0. We iterate:
4. We multiply T_j by $\hat{g}(2^j u, 2^j v)$. We get the complex array W_{j+1}. The inverse FFT gives the wavelet coefficients at the scale 2^j;
5. We multiply T_j by $\hat{h}(2^j u, 2^j v)$. We get the array T_{j+1}. Its inverse FFT gives the image at the scale 2^{j+1}. The frequency band is reduced by a factor 2.
6. We increment j
7. If $j \leq n_p$, we go back to 4.
8. The set $\{w_1, w_2, \ldots, w_{n_p}, c_{n_p}\}$ describes the wavelet transform.

If the wavelet is the difference between two resolutions, we have:

$$\hat{\psi}(2\nu) = \hat{\phi}(\nu) - \hat{\phi}(2\nu) \tag{14.58}$$

and:

$$\hat{g}(\nu) = 1 - \hat{h}(\nu) \tag{14.59}$$

then the wavelet coefficients $\hat{w}_j(\nu)$ can be computed by $\hat{c}_{j-1}(\nu) - \hat{c}_j(\nu)$.

The Reconstruction

If the wavelet is the difference between two resolutions, an evident reconstruction for a wavelet transform $\mathcal{W} = \{w_1, w_2, \ldots, w_{n_p}, c_{n_p}\}$ is:

$$\hat{c}_0(\nu) = \hat{c}_{n_p}(\nu) + \sum_j \hat{w}_j(\nu) \tag{14.60}$$

But this is a particular case and other wavelet functions can be chosen. The reconstruction can be done step by step, starting from the lowest resolution. At each scale, we have the relations:

$$\hat{c}_{j+1} = \hat{h}(2^j\nu)\hat{c}_j(\nu) \tag{14.61}$$

$$\hat{w}_{j+1} = \hat{g}(2^j\nu)\hat{c}_j(\nu) \tag{14.62}$$

we look for c_j knowing c_{j+1}, w_{j+1}, h and g. We restore $\hat{c}_j(\nu)$ with a least mean square estimator:

$$\hat{p}_h(2^j\nu)|\hat{c}_{j+1}(\nu) - \hat{h}(2^j\nu)\hat{c}_j(\nu)|^2 + \hat{p}_g(2^j\nu)|\hat{w}_{j+1}(\nu) - \hat{g}(2^j\nu)\hat{c}_j(\nu)|^2 \tag{14.63}$$

is minimum. $\hat{p}_h(\nu)$ and $\hat{p}_g(\nu)$ are weight functions which permit a general solution to the restoration of $\hat{c}_j(\nu)$. By $\hat{c}_j(\nu)$ derivation we get:

$$\hat{c}_j(\nu) = \hat{c}_{j+1}(\nu)\hat{\tilde{h}}(2^j\nu) + \hat{w}_{j+1}(\nu)\hat{\tilde{g}}(2^j\nu) \tag{14.64}$$

where the conjugate filters have the expression:

$$\hat{\tilde{h}}(\nu) = \frac{\hat{p}_h(\nu)\hat{h}^*(\nu)}{\hat{p}_h(\nu)|\hat{h}(\nu)|^2+\hat{p}_g(\nu)|\hat{g}(\nu)|^2} \tag{14.65}$$

$$\hat{\tilde{g}}(\nu) = \frac{\hat{p}_g(\nu)\hat{g}^*(\nu)}{\hat{p}_h(\nu)|\hat{h}(\nu)|^2+\hat{p}_g(\nu)|\hat{g}(\nu)|^2} \tag{14.66}$$

It is easy to see that these filters satisfy the exact reconstruction equation 14.19. In fact, equations 14.65 and 14.66 give the general solution to this equation. In this analysis, the Shannon sampling condition is always respected. No aliasing exists, so that the dealiasing condition 14.18 is not necessary.

The denominator is reduced if we choose:

$$\hat{g}(\nu) = \sqrt{1 - \mid \hat{h}(\nu) \mid^2}$$

This corresponds to the case where the wavelet is the difference between the square of two resolutions:

$$\mid \hat{\psi}(2\nu) \mid^2 = \mid \hat{\phi}(\nu) \mid^2 - \mid \hat{\phi}(2\nu) \mid^2 \tag{14.67}$$

We plot in figure 14.10 the chosen scaling function derived from a B-spline of degree 3 in the frequency space and its resulting wavelet function. Their conjugate functions are plotted in figure 14.11.

The reconstruction algorithm is:

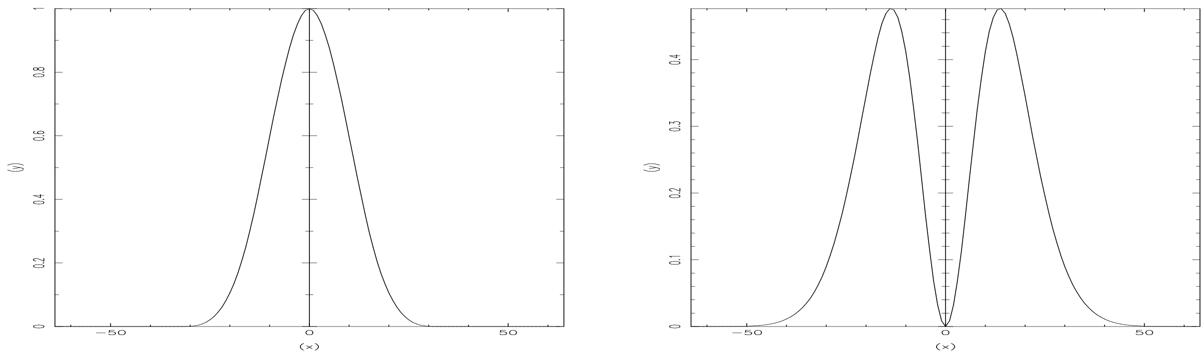

Figure 14.10: Left, the interpolation function $\hat{\phi}$ and right, the wavelet $\hat{\psi}$.

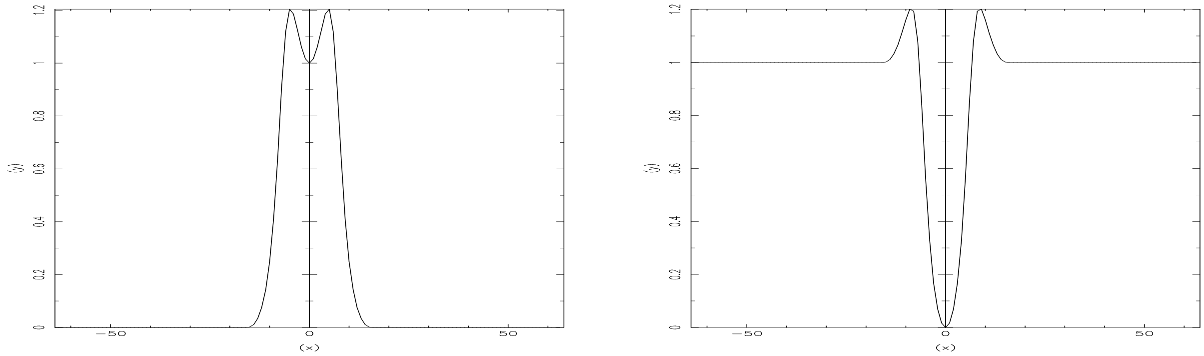

Figure 14.11: On left, the filter $\hat{\hat{h}}$, and on right the filter $\hat{\hat{g}}$.

1. We compute the FFT of the image at the low resolution.

2. We set j to n_p. We iterate:

3. We compute the FFT of the wavelet coefficients at the scale j.

4. We multiply the wavelet coefficients $\hat{w}_j$ by $\hat{\hat{g}}$.

5. We multiply the image at the lower resolution $\hat{c}_j$ by $\hat{\hat{h}}$.

6. The inverse Fourier Transform of the addition of $\hat{w}_j\hat{\hat{g}}$ and $\hat{c}_i\hat{\hat{h}}$ gives the image c_{j-1}.

7. $j = j - 1$ and we go back to 3.

The use of a scaling function with a cut-off frequency allows a reduction of sampling at each scale, and limits the computing time and the memory size.

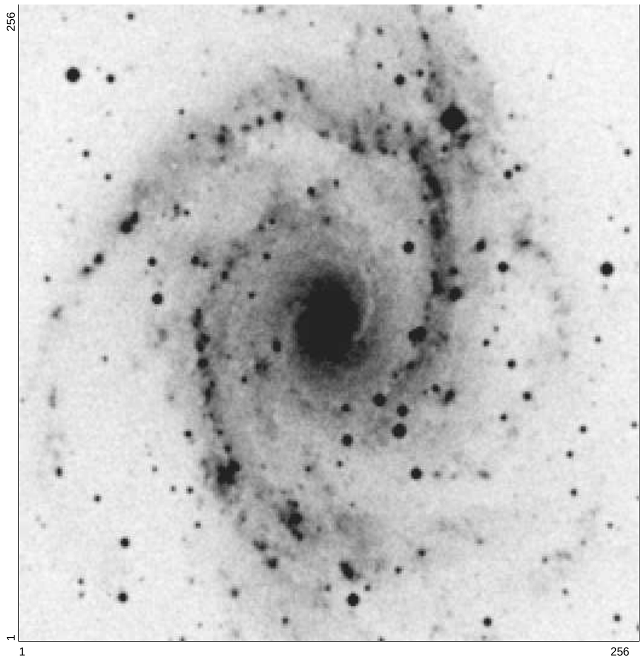

Figure 14.12: Galaxy NGC2297

14.5 Visualization of the Wavelet Transform

We have seen that the wavelet transform furnishes a number of data which depends on the algorithm used. We distinguish three classes of algorithms:

- those which do not reduce the sampling. The number of wavelet coefficients is equal to the number of pixels of the image multiplied by the number of scales. This is the case if we use the *à trous* algorithm.
- those which furnish a pyramidal set of data
- those which furnish an image

In the following, we present how the galaxy (figure 14.12) can be represented in the wavelet space.

14.5.1 Visualisation of the first class

The wavelet coefficients can be represented in several ways. Five visualisation tools are available in MIDAS.

- Through the command *visual/plan*, a window is created for each scale. The user can select one window and do all the operations available in MIDAS for an image.

- All the scales can be plotted in a unique window. Figure 14.13, obtained by the command *visual/cube*, shows the superposition of the scales in a window.
- In figure 14.14, each scale is plotted in a 3 dimensional representation (command *visual/pers*).
- In figure 14.15, each scale is binarized and represented by gray level (command *visual/synt*).
- In figure 14.16, one contour per scale is plotted (command *visual/cont*).

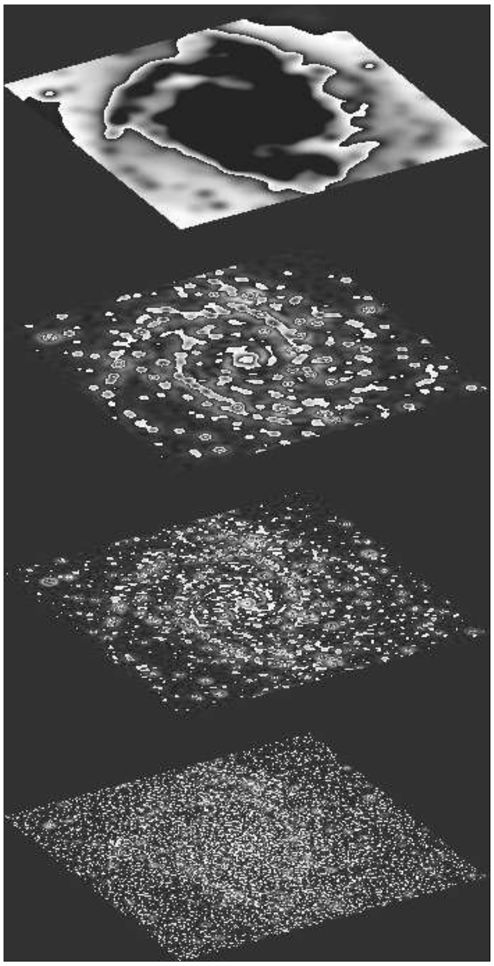

Figure 14.13: Superposition of all the scales. This image is obtained by the command *visual/cube.*

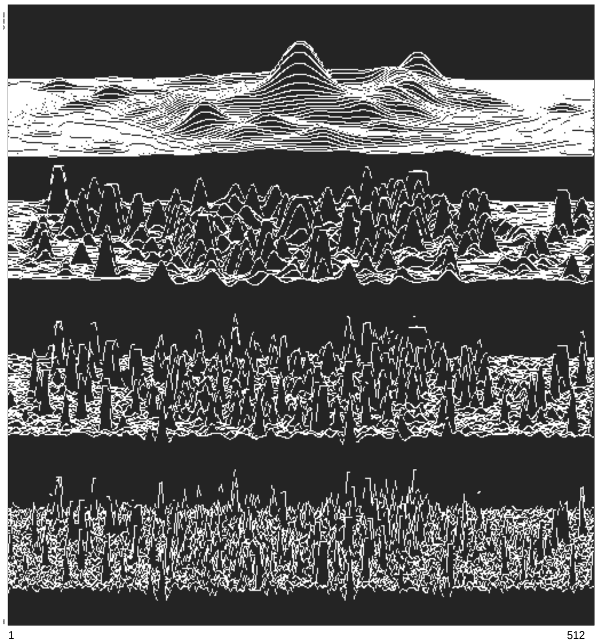

Figure 14.14: Superposition of all the scales. Each scale is plotted in a 3 dimensional representation. This image is obtained by the command *visual/pers*.

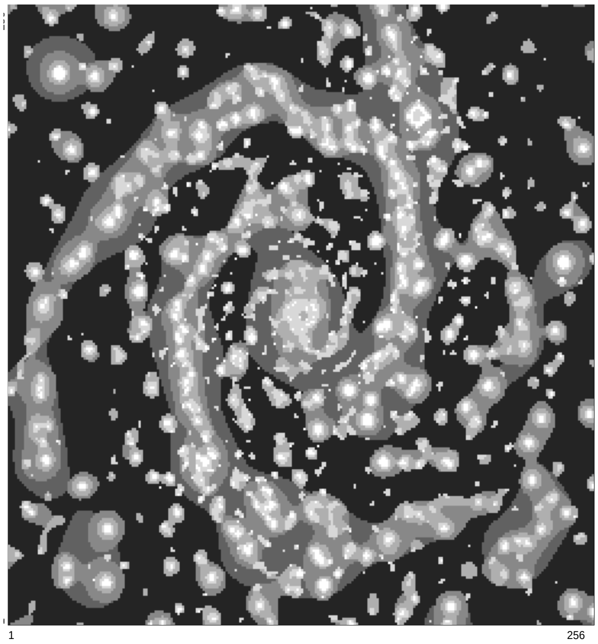

Figure 14.15: Synthesis image (command *visual/synt*). Each scale is binarized, and represented by one gray level.

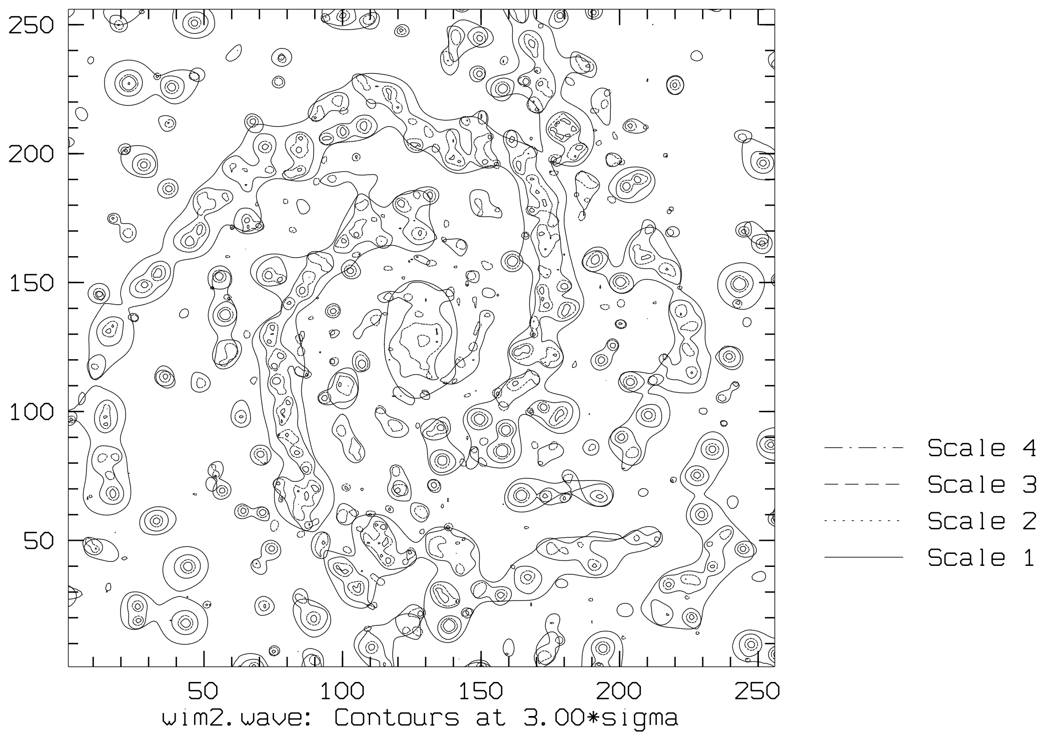

Figure 14.16: One contour per scale is plotted (command *visual/cont*).

14.5.2 Visualisation of the second class

Five visualisation tools are available in MIDAS for pyramidal wavelet transform.

- Through the command *visual/plan*, a window is created for each scale. The user can select one window and do all the operations available in MIDAS for an image. Scales are interpolated in order to have the same size as the image.
- All the scales can be plotted in a unique window. The figure 14.17, obtained by the command *visual/cube*, shows the superposition of the scales in a window.
- In figure 14.18, each scale is plotted in a 3 dimensional representation (command *visual/pers*).
- In figure 14.19, all the scales are represented in an image which is 2 times bigger than the original one (command *visual/synt*).
- By interpolating each scale to the size of the original image, a plot similar to figure 14.16 can be obtained (command *visual/cont*).

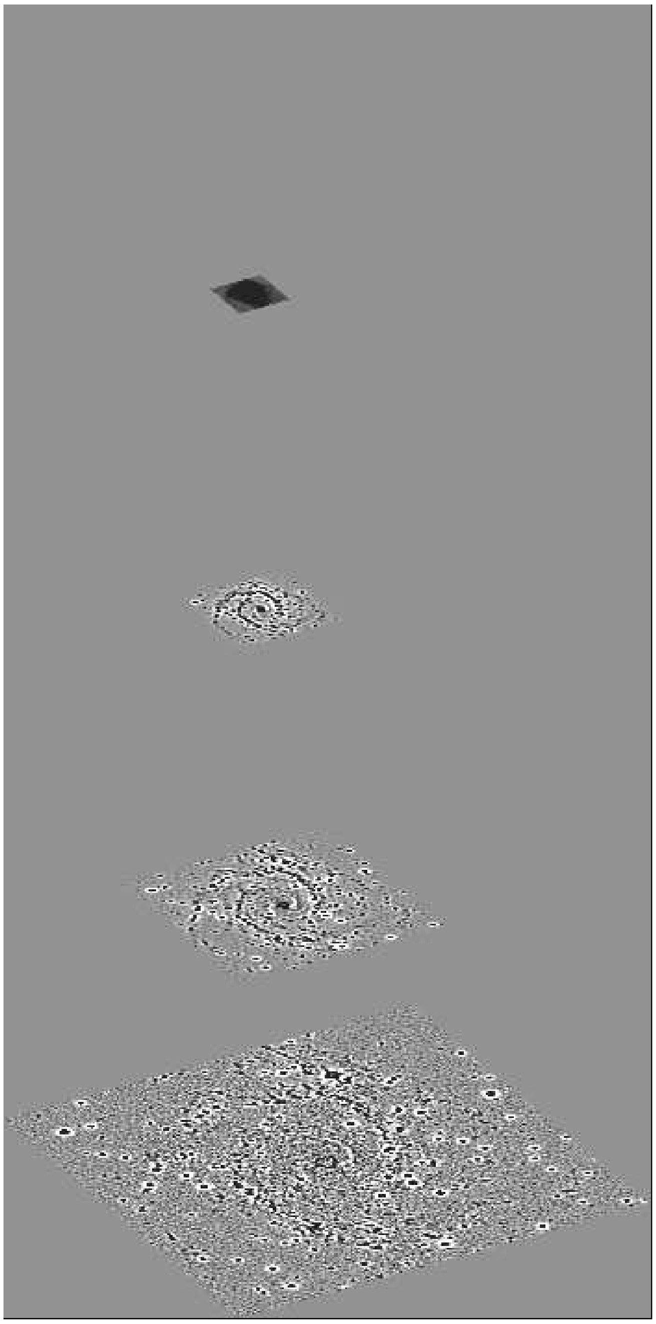

Figure 14.17: Superposition of all the scales. This image is obtained by the command *visual/cube*.

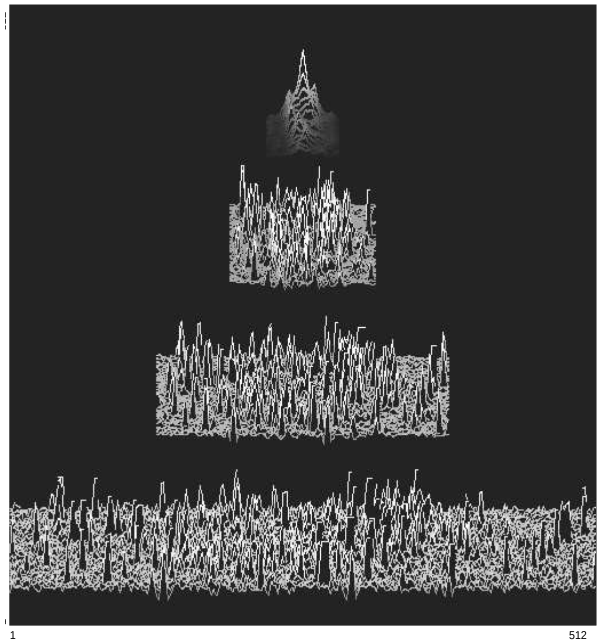

Figure 14.18: Superposition of all the scales. Each scale is plotted in a 3 dim. representation. This image is obtained by the command *visual/pers*.

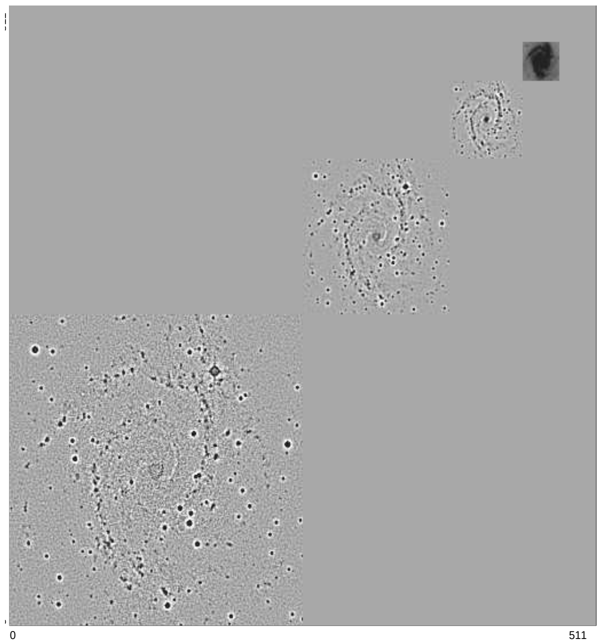

Figure 14.19: Synthesis image (command *visual/synt*).

14.5.3 Visualisation of the third class

The last class has only two visualisation tools. One consists in displaying the wavelet coefficients in an image, and the second does the same thing, but normalizing all the scales in order to have a better representation (see figure 14.20).

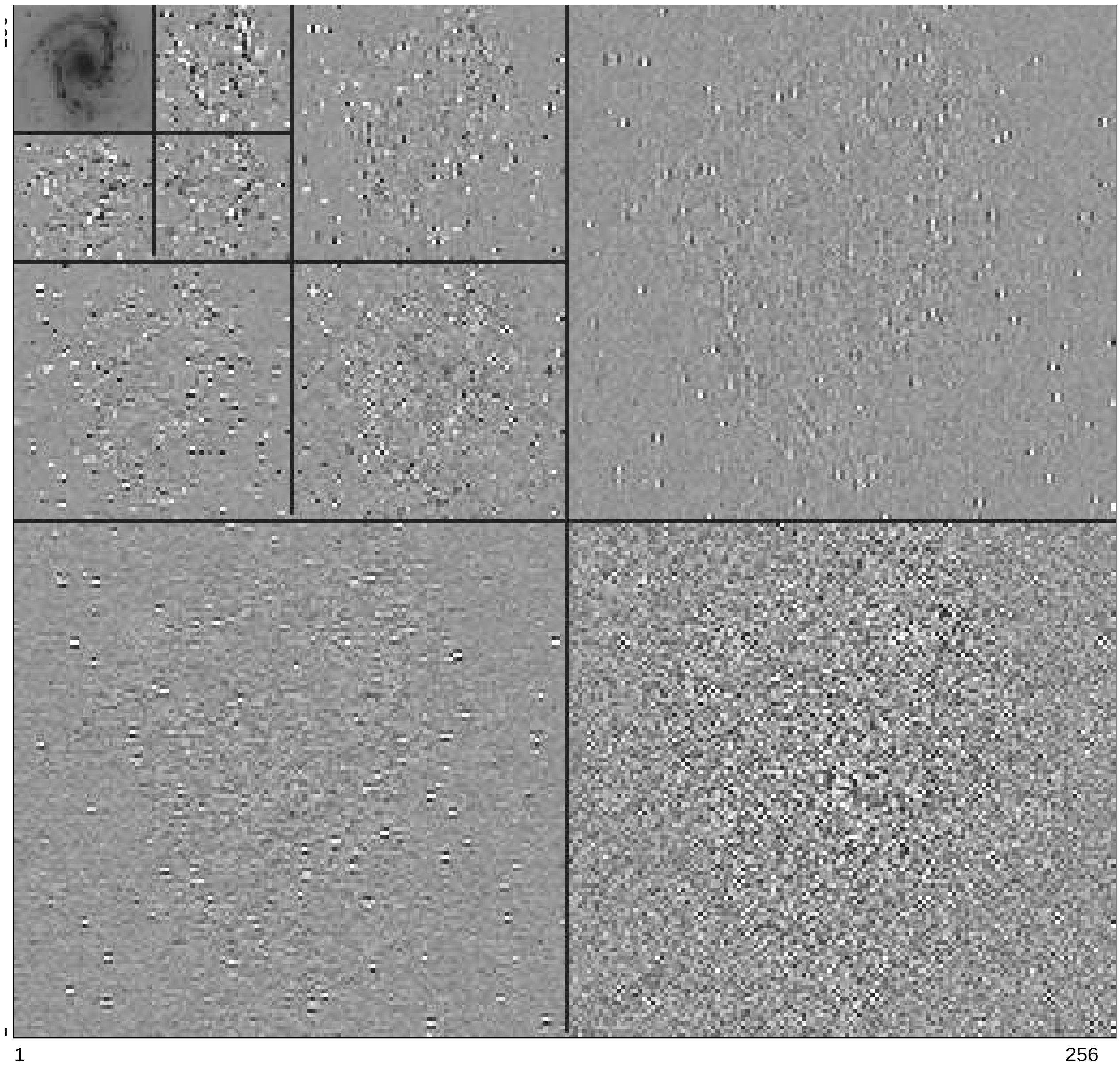

Figure 14.20: Synthesis image (command *visual/synt*). Each scale is normalized.

14.6 Noise reduction from the wavelet transform

14.6.1 The convolution from the continuous wavelet transform

We will examine here the computation of a convolution by using the continuous wavelet transform in order to get a framework for linear smoothings. Let us consider the convolution product of two functions:

$$h(x) = \int_{-\infty}^{+\infty} f(u)g(x-u)dx \tag{14.68}$$

We introduce two real wavelets functions $\psi(x)$ and $\chi(x)$ such that:

$$C = \int_{0}^{+\infty} \frac{\hat{\psi}^*(\nu)\hat{\chi}(\nu)}{\nu} d\nu \tag{14.69}$$

is defined. $W_g(a,b)$ denotes the wavelet transform of g with the wavelet function $\psi(x)$:

$$W_g(a,b) = \frac{1}{\sqrt{a}} \int_{-\infty}^{+\infty} g(x)\psi^*(\frac{x-b}{a})dx \tag{14.70}$$

We restore $g(x)$ with the wavelet function $\chi(x)$:

$$g(x) = \frac{1}{C} \int_{0}^{+\infty} \int_{-\infty}^{+\infty} \frac{1}{\sqrt{a}} W_g(a,b)\chi(\frac{x-b}{a}) \frac{dadb}{a^2} \tag{14.71}$$

The convolution product can be written as:

$$h(x) = \frac{1}{C} \int_{0}^{+\infty} \frac{da}{a^{\frac{5}{2}}} \int_{-\infty}^{+\infty} W_g(a,b)db \int_{-\infty}^{+\infty} f(u)\chi(\frac{x-u-b}{a})du \tag{14.72}$$

Let us denote $\tilde{\chi}(x) = \chi(-x)$. The wavelet transform $W_f(a,b)$ of $f(x)$ with the wavelet $\tilde{\chi}(x)$ is:

$$\tilde{W}_f(a,b) = \frac{1}{\sqrt{a}} \int_{-\infty}^{+\infty} f(x)\tilde{\chi}(\frac{x-b}{a})dx \tag{14.73}$$

That leads to:

$$h(x) = \frac{1}{C} \int_{0}^{+\infty} \frac{da}{a^2} \int_{-\infty}^{+\infty} \tilde{W}_f(a, x-b)W_g(a,b)db \tag{14.74}$$

Then we get the final result:

$$h(x) = \frac{1}{C} \int_{0}^{+\infty} \tilde{W}_f(a,x) \otimes W_g(a,x) \frac{da}{a^2} \tag{14.75}$$

In order to compute a convolution with the continuous wavelet transform:

- We compute the wavelet transform $\tilde{W}_f(a,b)$ of the function $f(x)$ with the wavelet function $\tilde{\chi}(x)$;

- We compute the wavelet transform $W_g(a,b)$ of the function $g(x)$ with the wavelet function $\psi(x)$;
- We sum the convolution product of the wavelet transforms, scale by scale.

The wavelet transform permits us to perform any linear filtering. Its efficiency depends on the number of terms in the wavelet transform associated with $g(x)$ for a given signal $f(x)$. If we have a filter where the number of significant coefficients is small for each scale, the complexity of the algorithm is proportional to N. For a classical convolution, the complexity is also proportional to N, but the number of operations is also proportional to the length of the convolution mask. The main advantage of the present technique lies in the possibility of having a filter with long scale terms without computing the convolution on a large window. If we achieve the convolution with the FFT algorithm, the complexity is of order $N \log_2 N$. The computing time is longer than the one obtained with the wavelet transform if we concentrate the energy on very few coefficients.

14.6.2 The Wiener-like filtering in the wavelet space

Let us consider a measured wavelet coefficient w_i at the scale i. We assume that its value, at a given scale and a given position, results from a noisy process, with a Gaussian distribution with a mathematical expectation W_i, and a standard deviation B_i:

$$P(w_i/W_i) = \frac{1}{\sqrt{2\pi}B_i} e^{-\frac{(w_i - W_i)^2}{2B_i^2}} \tag{14.76}$$

Now, we assume that the set of expected coefficients W_i for a given scale also follows a Gaussian distribution, with a null mean and a standard deviation S_i:

$$P(W_i) = \frac{1}{\sqrt{2\pi}S_i} e^{-\frac{W_i^2}{2S_i^2}} \tag{14.77}$$

The null mean value results from the wavelet property:

$$\int_{-\infty}^{+\infty} \psi^*(x)dx = 0 \tag{14.78}$$

We want to get an estimate of W_i knowing w_i. Bayes' theorem gives:

$$P(W_i/w_i) = \frac{P(W_i)P(w_i/W_i)}{P(w_i)} \tag{14.79}$$

We get:

$$P(W_i/w_i) = \frac{1}{\sqrt{2\pi}\beta_i} e^{-\frac{(W_i - \alpha_i w_i)^2}{2\beta_i^2}} \tag{14.80}$$

where:

$$\alpha_i = \frac{S_i^2}{S_i^2 + B_i^2} \tag{14.81}$$

the probability $P(W_i/w_i)$ follows a Gaussian distribution with a mean:

$$m = \alpha_i w_i \tag{14.82}$$

and a variance:

$$\beta_i^2 = \frac{S_i^2 B_i^2}{S_i^2 + B_i^2} \tag{14.83}$$

The mathematical expectation of W_i is $\alpha_i w_i$.

With a simple multiplication of the coefficients by the constant α_i, we get a linear filter. The algorithm is:

1. Compute the wavelet transform of the data. We get w_i.
2. Estimate the standard deviation of the noise B_0 of the first plane from the histogram of w_0. As we process oversampled images, the values of the wavelet image corresponding to the first scale (w_0) are due mainly to the noise. The histogram shows a Gaussian peak around 0. We compute the standard deviation of this Gaussian function, with a 3σ clipping, rejecting pixels where the signal could be significant;
3. Set i to 0.
4. Estimate the standard deviation of the noise B_i from B_0. This is done from the study of the variation of the noise between two scales, with an hypothesis of a white gaussian noise;
5. $S_i^2 = s_i^2 - B_i^2$ where s_i^2 is the variance of w_i.
6. $\alpha_i = \frac{S_i^2}{S_i^2 + B_i^2}$.
7. $W_i = \alpha_i w_i$.
8. $i = i + 1$ and go to 4.
9. Reconstruct the picture from W_i.

14.6.3 Hierarchical Wiener filtering

In the above process, we do not use the information between the wavelet coefficients at different scales. We modify the previous algorithm by introducing a prediction w_h of the wavelet coefficient from the upper scale. This prediction could be determined from the regression [2] between the two scales but better results are obtained when we only set w_h to W_{i+1}. Between the expectation coefficient W_i and the prediction, a dispersion exists where we assume that it is a Gaussian distribution:

$$P(W_i/w_h) = \frac{1}{\sqrt{2\pi}T_i} e^{-\frac{(W_i - w_h)^2}{2T_i^2}} \tag{14.84}$$

The relation which gives the coefficient W_i knowing w_i and w_h is:

$$P(W_i/w_i \text{ and } w_h) = \frac{1}{\sqrt{2\pi}\beta_i} e^{-\frac{(W_i-\alpha_i w_i)^2}{2\beta_i^2}} \frac{1}{\sqrt{2\pi}T_i} e^{-\frac{(W_i-w_h)^2}{2T_i^2}} \tag{14.85}$$

with:

$$\beta_i^2 = \frac{S_i^2 B_i^2}{S^2 + B_i^2} \tag{14.86}$$

and:

$$\alpha_i = \frac{S_i^2}{S_i^2 + B_i^2} \tag{14.87}$$

This follows a Gaussian distribution with a mathematical expectation:

$$W_i = \frac{T_i^2}{B_i^2 + T_i^2 + Q_i^2} w_i + \frac{B_i^2}{B_i^2 + T_i^2 + Q_i^2} w_h \tag{14.88}$$

with:

$$Q_i^2 = \frac{T_i^2 B_i^2}{S_i^2} \tag{14.89}$$

W_i is the barycentre of the three values w_i, w_h, 0 with the weights T_i^2, B_i^2, Q_i^2. The particular cases are:

- If the noise is large ($S_i \ll B_i$) and even if the correlation between the two scales is good (T_i is low), we get $W_i \rightarrow 0$.
- if $B_i \ll S_i \ll T$ then $W_i \rightarrow w_i$.
- if $B_i \ll T_i \ll S$ then $W_i \rightarrow w_i$.
- if $T_i \ll B_i \ll S$ then $W_i \rightarrow w_h$.

At each scale, by changing all the wavelet coefficients w_i of the plane by the estimate value W_i, we get a Hierarchical Wiener Filter. The algorithm is:

1. Compute the wavelet transform of the data. We get w_i.
2. Estimate the standard deviation of the noise B_0 of the first plane from the histogram of w_0.
3. Set i to the index associated with the last plane: $i = n$
4. Estimate the standard deviation of the noise B_i from B_0.
5. $S_i^2 = s_i^2 - B_i^2$ where s_i^2 is the variance of w_i
6. Set w_h to W_{i+1} and compute the standard deviation T_i of $w_i - w_h$.
7. $W_i = \frac{T_i^2}{B_i^2+T_i^2+Q_i^2} w_i + \frac{B_i^2}{B_i^2+T_i^2+Q_i^2} w_h$
8. $i = i - 1$. If $i > 0$ go to 4
9. Reconstruct the picture

14.6.4 Adaptive filtering from the wavelet transform

In the preceding algorithm we have assumed the properties of the signal and the noise to be stationary. The wavelet transform was first used to obtain an algorithm which is faster than classical Wiener Filtering. Then we took into account the correlation between two different scales. In this way we got a filtering with stationary properties. In fact, these hypotheses were too simple, because in general the signal may not arise from a Gaussian stochastic process. Knowing the noise distribution, we can determine the statistically significant level at each scale of the measured wavelet coefficients. If $w_i(x)$ is very weak, this level is not significant and could be due to noise. Then the hypothesis that the value $W_i(x)$ is null is not forbidden. In the opposite case where $w_i(x)$ is significant, we keep its value. If the noise is Gaussian, we write:

$$W_i = 0 \quad \text{if } \mid w_i \mid < kB_i \tag{14.90}$$

$$W_i = w_i \quad \text{if } \mid w_i \mid \geq kB_i \tag{14.91}$$

Generally, we choose $k = 3$.

With a filter bank we have a biunivocity between the image and its transform, so that the thresholded transform leads to only one restored image. Some experiments show us that uncontrolled artifacts appear for high level thresholding ($k = 3$). The decimation done at each step on the wavelet transform takes into account the knowledge of the coefficients at further resolutions. The thresholding sets to zero the intrinsic small terms which play their part in the reconstruction. With the lattice filter the situation is very different. No decimation is done and the thresholding keeps all significant coefficients. Where the coefficients are set to zero, we do not put zero, but we say that these values are unknown. The redundancy is used to restore them. Before the thresholding we have a redundant transform, which can be decimated, after the thresholding we get a set of coefficients from which we wish to restore in image.

If one applies the reconstruction algorithm, then it is not guaranteed that the wavelet transform of the restored image will give the same values for the coefficients. This is not important in the case where they are not significant, but otherwise the same values must be found. If $W_i^{(s)}$ are the coefficients obtained by the thresholding, then we require $W_i(x)$ such that:

$$P.W_i(x) = W_i^{(s)}(x) \tag{14.92}$$

where P is the non linear operator which performs the inverse transform, the wavelet transform, and the thresholding. An alternative is to use the following iterative solution which is similar to Van Cittert's algorithm:

$$W_i^{(n)}(x) = W_i^{(s)}(x) + W_i^{(n-1)}(x) - P.W_i^{(n-1)}(x) \tag{14.93}$$

for the significant coefficients ($W_i^{(s)}(x) \neq 0$) and:

$$W_i^{(n)}(x) = W_i^{(n-1)}(x) \tag{14.94}$$

for the non significant coefficients ($W_i^{(s)}(x) = 0$).

The algorithm is the following one:

1. Compute the wavelet transform of the data. We get w_i.

2. Estimate the standard deviation of the noise B_0 of the first plane from the histogram of w_0.

3. Estimate the standard deviation of the noise B_i from B_0 at each scale.

4. Estimate the significant level at each scale, and threshold.

5. Initialize: $W_i^{(0)}(x) = W_i^{(s)}(x)$

6. Reconstruct the picture by using the iterative method.

The thresholding may introduce negative values in the resulting image. A positivity constraint can be introduced in the iterative process, by thresholding the restored image. The algorithm converges after five or six iterations.

14.6.5 Hierarchical adaptive filtering

In the previous algorithm we do not use the hierarchy of structures. We have explored many approaches for introducing a non-linear hierarchical law in the adaptive filtering and we found that the best way was to link the threshold to the wavelet coefficient of the previous plane w_h. We get:

$$\begin{array}{rcl} W_i(x) = & w_i(x) & \text{if } \mid w_i(x) \mid \geq L \\ W_i(x) = & 0 & \text{if } \mid w_i(x) \mid < L \end{array}$$

and L is a threshold estimated by:

$$\begin{array}{lll} \text{if } \mid w_i(x) \mid \geq kB_i & \text{then } L = & kB_i \\ \text{if } \mid w_i(x) \mid < kB_i & \text{then } L = & kB_i t(\mid \dfrac{w_h}{S_h} \mid) \end{array}$$

where S_h is the standard deviation of w_h. The function $t(a)$ must return a value between 0 and 1. A possible function for t is:

- $t(a) = 0$ if $a \geq k$
- $t(a) = 1 - \frac{1}{k}a$ if $a < k$

14.7 Comparison using a multiresolution quality criterion

It is sometimes useful, as in image restoration where we want to evaluate the quality of the restoration, to compare images with an objective criterion. Very few quantitative parameters can be extracted for that. The correlation between the original image $I(i,j)$ and the restored one $\tilde{I}(i,j)$ gives a classical criterion. The correlation coefficient is:

$$C_{or} = \frac{\sum_{i=1}^{N}\sum_{j=1}^{N} I(i,j)\tilde{I}(i,j)}{\sqrt{\sum_{i=1}^{N}\sum_{j=1}^{N} I^2(i,j)\sum_{i=1}^{N}\sum_{j=1}^{N} \tilde{I}^2(i,j)}} \tag{14.95}$$

The correlation is 1 if the images are identical, and less if some differences exist. Another way to compare two pictures is to determine the mean-square error:

$$E_{ms}^2 = \frac{1}{N^2}\sum_{i=1}^{N}\sum_{j=1}^{N}(I(i,j) - \tilde{I}(i,j))^2 \tag{14.96}$$

E_{ms}^2 can be normalized by:

$$E_{nms}^2 = \frac{\sum_{i=1}^{N}\sum_{j=1}^{N}(I(i,j) - \tilde{I}(i,j))^2}{\sum_{i=1}^{N}\sum_{j=1}^{N} I^2(i,j)} \tag{14.97}$$

The Signal-to-Noise Ratio (SNR) corresponding to the above error is:

$$SNR_{dB} = 10\log_{10}\frac{1}{E_{nms}^2}\ \text{dB} \tag{14.98}$$

These criteria are not sufficient, they give no information on the resulting resolution. A complete criterion must take into account the resolution. For each dyadic scale, we can compute the correlation coefficient and the quadratic error between the wavelet transforms of the original and the restored images. Hence, we can compare, the quality of the restoration for each resolution.

Figures 14.21 and 14.22 show the comparison of three images with a reference image. *Data20* is a simulated noisy image, *median* and *wave* are the output images after respectively applying a median filter, and a thresholding in the wavelet space. These curves show that the thresholding in the wavelet space is better than the median at all the scales.

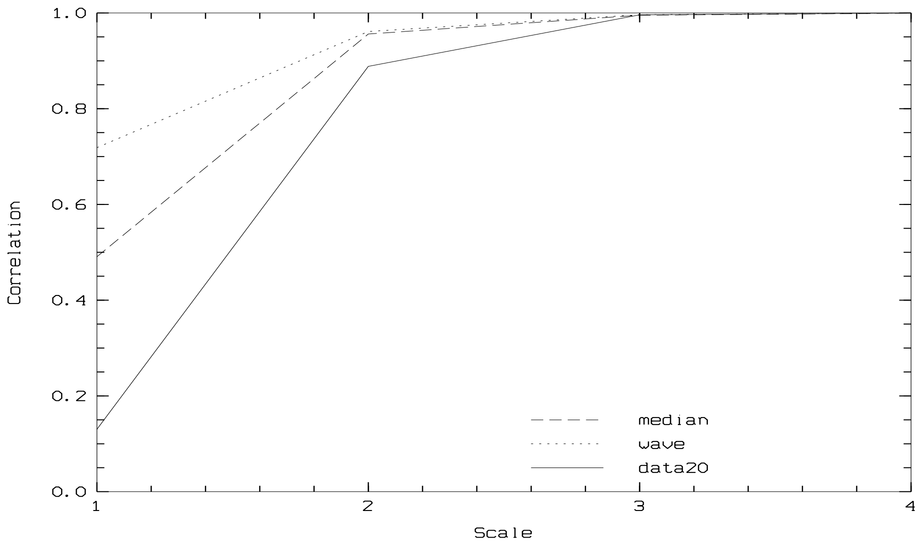

Figure 14.21: Correlation.

14.8 Deconvolution

14.8.1 Introduction

Consider an image characterized by its intensity distribution $I(x, y)$, corresponding to the observation of an object $O(x, y)$ through an optical system. If the imaging system is linear and shift-invariant, the relation between the object and the image in the same coordinate frame is a convolution:

$$I(x, y) = O(x, y) * P(x, y) + N(x, y) \tag{14.99}$$

$P(x, y)$ is the point spread function (PSF) of the imaging system, and $N(x, y)$ is an additive noise. In Fourier space we have:

$$\hat{I}(u, v) = \hat{O}(u, v)\hat{P}(u, v) + \hat{N}(u, v) \tag{14.100}$$

We want to determine $O(x, y)$ knowing $I(x, y)$ and $P(x, y)$. This inverse problem has led to a large amount of work, the main difficulties being the existence of: (i) a cut-off frequency of the PSF, and (ii) an intensity noise (see for example [6]).

Equation 14.99 is always an ill-posed problem. This means that there is not a unique least-squares solution of minimal norm $\| I(x, y) - P(x, y) * O(x, y) \|^2$ and a regularization is necessary.

The best restoration algorithms are generally iterative [24]. Van Cittert [41] proposed the following iteration:

$$O^{(n+1)}(x, y) = O^{(n)}(x, y) + \alpha(I(x, y) - P(x, y) * O^{(n)}(x, y)) \tag{14.101}$$

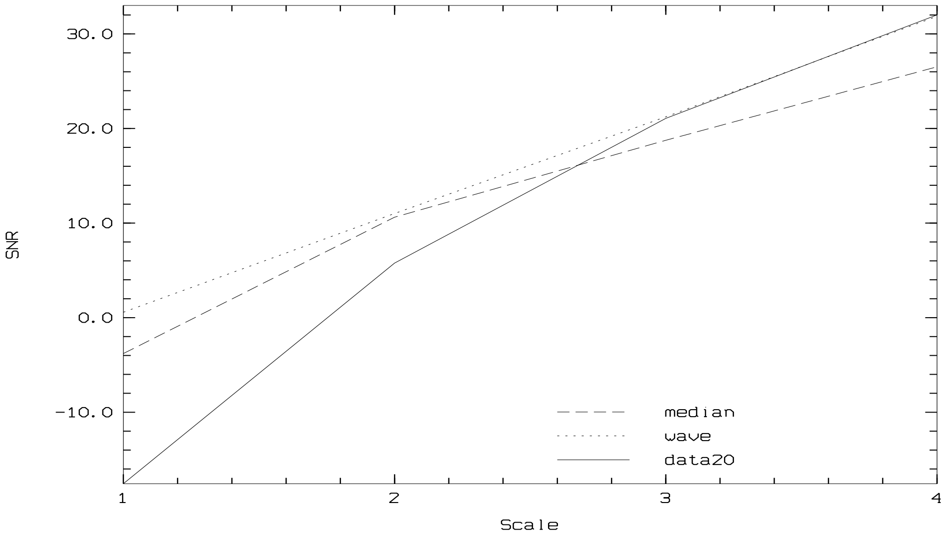

Figure 14.22: Signal to noise ratio.

where α is a converging parameter generally taken as 1. In this equation, the object distribution is modified by adding a term proportional to the residual. But this algorithm diverges when we have noise [12]. Another iterative algorithm is provided by the minimization of the norm $\| I(x,y) - P(x,y) * O(x,y) \|^2$ [21] and leads to:

$$O^{(n+1)}(x,y) = O^{(n)}(x,y) + \alpha P_s(x,y) * [I(x,y) - P(x,y) * O^{(n)}(x,y)] \quad (14.102)$$

where $P_s(x,y) = P(-x,-y)$.

Tikhonov's regularization [40] consists of minimizing the term:

$$\| I(x,y) - P(x,y) * O(x,y) \|^2 + \lambda \| H * O \|^2 \quad (14.103)$$

where H corresponds to a high-pass filter. This criterion contains two terms; the first one, $\| I(x,y) - P(x,y) * O(x,y) \|^2$, expresses fidelity to the data $I(x,y)$ and the second one, $\lambda \| H * O \|^2$, smoothness of the restored image. λ is the regularization parameter and represents the trade-off between fidelity to the data and the restored image smoothness. Finding the optimal value λ necessitates use of numeric techniques such as Cross-Validation [15] [14].

This method works well, but it is relatively long and produces smoothed images. This second point can be a real problem when we seek compact structures as is the case in astronomical imaging.

An iterative approach for computing maximum likelihood estimates may be used. The Lucy method [23, 24, 1] uses such an iterative approach:

$$O^{(n+1)} = O^{(n)}[\frac{I}{I^{(n)}} * P^*] \quad (14.104)$$

and

$$I^{(n)} = P * O^{(n)} \tag{14.105}$$

where P^* is the conjugate of the PSF.

14.8.2 Regularization in the wavelet space

14.8.3 Tikhonov's regularization and multiresolution analysis

If $w_j^{(I)}$ are the wavelet coefficients of the image I at the scale j, we have:

$$\begin{aligned} \hat{w}_j^{(I)}(u,v) &= \hat{g}(2^{j-1}u, 2^{j-1}v) \prod_{i=j-2}^{i=0} \hat{h}(2^i u, 2^i v) \hat{I}(u,v) \\ &= \frac{\hat{\psi}(2^j u, 2^j v)}{\hat{\phi}(u,v)} \hat{P}(u,v)\hat{O}(u,v) \\ &= \hat{w}_j^{(P)} \hat{O}(u,v) \end{aligned} \tag{14.106}$$

where $w_j^{(P)}$ are the wavelet coefficients of the PSF at the scale j. The wavelet coefficients of the image I are the product of convolution of object O by the wavelet coefficients of the PSF.

To deconvolve the image, we have to minimize for each scale j:

$$\parallel \frac{\hat{\psi}(2^j u, 2^j v)}{\hat{\phi}(u,v)} \hat{P}(u,v)\hat{O}(u,v) - \hat{w}_j^{(I)}(u,v) \parallel^2 \tag{14.107}$$

and for the plane at the lower resolution:

$$\parallel \frac{\hat{\phi}(2^{n-1} u, 2^{n-1} v)}{\hat{\phi}(u,v)} \hat{P}(u,v)\hat{O}(u,v) - \hat{c}_{n-1}^{(I)}(u,v) \parallel^2 \tag{14.108}$$

n being the number of planes of the wavelet transform (($n-1$) wavelet coefficient planes and one plane for the image at the lower resolution). The problem has not generally a unique solution, and we need to do a regularization [40]. At each scale, we add the term:

$$\gamma_j \parallel w_j^{(O)} \parallel^2 \ \min \tag{14.109}$$

This is a smoothness constraint. We want to have the minimum information in the restored object. From equations 14.107, 14.108, 14.109, we find:

$$\hat{D}(u,v)\hat{O}(u,v) = \hat{N}(u,v) \tag{14.110}$$

with:

$$\hat{D}(u,v) = \sum_j \mid \hat{\psi}(2^j u, 2^j v) \mid^2 (\mid \hat{P}(u,v) \mid^2 + \gamma_j) + \mid \hat{\phi}(2^{n-1}u, 2^{n-1}v)\hat{P}(u,v) \mid^2$$

and:

$$\hat{N}(u,v) = \hat{\phi}(u,v)[\sum_j \hat{P}^*(u,v)\hat{\psi}^*(2^j u, 2^j v)\hat{w}_j^{(I)} + \hat{P}^*(u,v)\hat{\phi}^*(2^{n-1}u, 2^{n-1}v)\hat{c}_{n-1}^{(I)}]$$

if the equation is well constrained, the object can be computed by a simple division of $\hat{N}$ by $\hat{D}$. An iterative algorithm can be used to do this inversion if we want to add other constraints such as positivity. We have in fact a multiresolution Tikhonov's regularization. This method has the advantage to furnish a solution quickly, but optimal regularization parameters γ_j cannot be found directly, and several tests are generally necessary before finding an acceptable solution. Hovewer, the method can be interesting if we need to deconvolve a big number of images with the same noise characteristics. In this case, parameters have to be determined only the first time. In a general way, we prefer to use one of the following iterative algorithms.

14.8.4 Regularization from significant structures

If we use an iterative deconvolution algorithm, such as Van Cittert's or Lucy's one, we define $R^{(n)}(x,y)$, the error at iteration n:

$$R^{(n)}(x,y) = I(x,y) - P(x,y) * O^{(n)}(x,y) \tag{14.111}$$

By using the *à trous* wavelet transform algorithm, $R^{(n)}$ can be defined by the sum of its n_p wavelet planes and the last smooth plane (see equation 14.33).

$$R^{(n)}(x,y) = c_{n_p}(x,y) + \sum_{j=1}^{n_p} w_j(x,y) \tag{14.112}$$

The wavelet coefficients provide a mechanism to extract from the residuals at each iteration only the significant structures. A large part of these residuals are generally statistically non significant. The significant residual is:

$$\bar{R}^{(n)}(x,y) = c_{n_p}(x,y) + \sum_{j=1}^{n_p} \alpha(w_j(x,y), N_j)\ w_i(x,y) \tag{14.113}$$

N_j is the standard deviation of the noise at scale j, and α is a function which is defined by:

$$\alpha(a,\sigma) = \begin{cases} 1 & \text{if } \mid a \mid \geq k\sigma \\ 0 & \text{if } \mid a \mid < k\sigma \end{cases} \tag{14.114}$$

The standard deviation of the noise N_j is estimated from the standard deviation of the noise in the image. This is done from the study of noise variation in the wavelet space, with the hypothesis of a white Gaussian noise.

We now show how the iterative deconvolution algorithms can be modified in order to take into account only the significant structure at each scale.

Regularization of Van Cittert's algorithm

Van Cittert's iteration is:

$$O^{(n+1)}(x,y) = O^{(n)}(x,y) + \alpha R^{(n)}(x,y) \tag{14.115}$$

with $R^{(n)}(x,y) = I(x,y) - P(x,y) * O^{(n)}(x,y)$. The regularization by the significant structures leads to:

$$O^{(n+1)}(x,y) = O^{(n)}(x,y) + \alpha \bar{R}^{(n)}(x,y) \tag{14.116}$$

The basic idea of our method consists of detecting, at each scale, structures of a given size in the residual $R^{(n)}(x,y)$ and putting them in the restored image $O^{(n)}(x,y)$. The process finishes when no more structures are detected. Then, we have separated the image $I(x,y)$ into two images $\tilde{O}(x,y)$ and $R(x,y)$. $\tilde{O}$ is the restored image, which does not contain any noise, and $R(x,y)$ is the final residual which does not contain any structure. R is our estimation of the noise $N(x,y)$.

Regularization of the one-step gradient method

The one-step gradient iteration is:

$$O^{(n+1)}(x,y) = O^{(n)}(x,y) + P(-x,-y) * R^{(n)}(x,y) \tag{14.117}$$

with $R^{(n)}(x,y) = I(x,y) - P(x,y) * O^{(n)}(x,y)$. The regularization by the significant structures leads to:

$$O^{(n+1)}(x,y) = O^{(n)}(x,y) + P(-x,-y) * \bar{R}^{(n)}(x,y) \tag{14.118}$$

Regularization of Lucy's algorithm

Now, define $I^{(n)}(x,y) = P(x,y) * O^{(n)}(x,y)$. Then $R^{(n)}(x,y) = I(x,y) - I^{(n)}(x,y)$, and hence $I(x,y) = I^{(n)}(x,y) + R^{(n)}(x,y)$. Lucy's equation is:

$$O^{(n+1)}(x,y) = O^{(n)}(x,y)[\frac{I^{(n)}(x,y) + R^{(n)}(x,y)}{I^{(n)}(x,y)} * P(-x,-y)] \tag{14.119}$$

and the regularization leads [39] to:

$$O^{(n+1)}(x,y) = O^{(n)}(x,y)[\frac{I^{(n)}(x,y) + \bar{R}^{(n)}(x,y)}{I^{(n)}(x,y)} * P(-x,-y)] \tag{14.120}$$

Convergence

The standard deviation of the residual is decreasing until no more significant structures are found. The convergence can be estimated from the residual. The algorithm stops when:

$$\frac{\sigma_{R^{(n-1)}} - \sigma_{R^{(n)}}}{\sigma_{R^{(n)}}} < \epsilon \tag{14.121}$$

14.9 The wavelet context in MIDAS

14.9.1 Introduction

A wavelet package concerning two dimensional imaging has been implemented in MIDAS. This package contains several wavelet transform algorithms, specific tools for the wavelet transform, visualisation commands, and three applications of the use of the wavelet transform: filtering, comparison, and deconvolution. These commands can be called if the wavelet context has been initialized by:

set/context wavelet

Table 14.1 shows the available commands.

14.9.2 Commands Description

TRANSF/WAVE

TRANSF/WAVE Image Wavelet [Algo] [Nbr_Scale] [Fc]

This command creates a file which contains the wavelet transform. The suffixe of a wavelet transform file is ".wave". It is automatically added to the name passed to the command. Several algorithms are proposed:

1. *à trous* algorithm with a linear scaling function. The wavelet function is the difference between two resolutions (see 14.4.3).
2. *à trous* with a B3-spline scaling function (default value). The wavelet function is the difference between two resolutions (see 14.4.3).
3. algorithm using the Fourier transform, without any reduction of the samples between two scales. The Fourier transform of the scaling function is a b3-spline and the wavelet function is the difference between two resolutions (14.4.5).
4. pyramidal algorithm in the direct space, with a linear scaling function (see section 14.4.4).
5. pyramidal algorithm in the direct space, with a b3-spline scaling function (see section 14.4.4).
6. algorithm using the Fourier transform with a reduction of the samples between two scales. The Fourier transform of the scaling function is a b3-spline the wavelet function is the difference between two resolutions (14.4.5).
7. algorithm using the Fourier transform with a reduction of the samples between two scales. The Fourier transform of the scaling function is a b3-spline. The wavelet function is the difference between the square of two resolutions (14.4.5).
8. Mallat's Algorithm with biorthogonal filters (14.4.2).

Commands	Description
TRANSF/WAVE	Image Wavelet [Algo] [Nbr_Scale] [Fc] creates the wavelet transform of an image
RECONS/WAVE	Wavelet Rec_Image reconstructs an image from its wavelet transform
HEADER/WAVE	Wavelet gives information about a wavelet transform
INFO/WAVE	Wavelet gives information about each scale of a wavelet transform
EXTRAC/WAVE	Wavelet Image_Out Scale_Number creates an image from a scale of the wavelet transform
ENTER/WAVE	Wavelet_in Image_in Scale_Number Wavelet_out replaces a scale of a wavelet transform by an image
VISUAL/WAVE	Wavelet [Visu_Type] visualizes a wavelet transform with default parameters
VISUAL/CUBE	Wavelet [output_file] [Disp] [Visu_Mode] [Display] visualizes a wavelet transform in a cube
VISUAL/CONT	Wavelet [Graphic_Number] [Visu_Mode] [Contour_Level] visualizes the contours of a wavelet transform
VISUAL/SYNT	Wavelet [output_file] [Display_Number] [Display] creates a visualization image from a wavelet transform
VISUAL/PLAN	Wavelet [Display] display each scale of the wavelet transform in a window
VISUAL/PERS	Wavelet [out_file] [Disp_N] [Visu_Mode] [Incr] [Thres] [Display] visualizes in perpsective a wavelet transform
FILTER/WAVE	I_In I_Out [Algo] [T_Filter] [Iter_Nbr] [N_Scale] [N_Sigma] [Noise] filters an image by using the wavelet transform
COMPAR/WAVE	Imag_1 Imag_2 [N_Scal] [N_Sigma] [T_Cor] [T_Snr] [Disp] [Init] compares two images in the wavelet space
PLOT/SNR	Tab_Snr plots the SNR table resulting from the comparison
PLOT/COR	Tab_Correl plots the correlation resulting from the comparison
TUTORIAL/WAVE	visualizes the wavelet transform of the galaxy NGC2997 with several algorithms
DIRECT/WAVE	Imag_In Psf Imag_Out [Nb_Scales] [γ_1,γ_2,γ_3,...] deconvolution with a multiresolution Tichonov's Regularisation
CITTERT/WAVE	Im_In Psf Im_Out [Resi] [Scal, Iter] [N_Sig, Noise] [Eps] [Fwhm] deconvolution by the regularized Van Cittert's algorithm
GRAD/WAVE	Im_In Psf Im_Out [Resi] [Nb_Scales] [N_Sigma, Noise] [Eps] [Max_Iter] deconvolution by the regularized one-step gradient algorithm
LUCY/WAVE	Im_In Psf Im_Out [Resi] [Nb_Scales] [N_Sigma, Noise] [Eps] [Max_Iter] deconvolution by the regularized Lucy's algorithm
TRAN1D/WAVE	Im_In Wave_Out [Num_Trans] [Num_Line] [Channel] [Nu] one dimensional wavelet transform
REC1D/WAVE	Wave_in Im_out [Num_Trans] [Channel] [Nu0] reconstructs a 1D signal from its wavelet transform

Table 14.1: Midas commands

The parameter *Algo* can be chosen between 1 and 8. If *Algo* is in $\{1,2,3\}$, the number of data of the wavelet transform is equal to the number of pixels multiplied by the number of scales (if the number of pixels of the image is N^2, the number of wavelet coefficients is Nbr_Scale.N^2). Algorithms 4, 5, 6, and 7 are pyramidal (the number of wavelet coefficients is $\frac{4}{3}N^2$), and the 8th algorithm does not increase the number of data (the size of the wavelet transform is N^2). Due to the discretisation and the undersampling, the properties of these algorithms are not the same. The 8th algorithm is more compact, but is not isotropic (see section 14.4.2). Algorithms 3, 6, and 7 compute the wavelet transform in the Fourier space (see section 14.4.5) and the undersampling respect Shannon's theorem. Pyramidal algorithms 4 and 5 compute the wavelet transform in the direct space, but need an interative reconstruction. Algorithms 1 and 2 are isotropic but increase the number of data. The 2D-discrete wavelet transform is not restricted the previous algorithms. Other algorithms exist (see for example Feauveau's one [11] which is not diadic). The interest of the wavelet transform is that it is a very flexible tool. We can adapt the transform to our problem. We prefer the 8th for image compression, 6 and 7 for image restoration, 2 for data analysis, *etc.*. The wavelet function can be derived too from the specific problem to resolve (see [35]).

The parameter *Nbr_Scale* specifies the number of scales to compute. The wavelet transform will contain *Nbr_Scale* $-$ 1 wavelet coefficients planes and one plane which will be the image at a very low resolution.

The parameter *Fc* defines the cut-off frequency of the scaling function ($0 < F_c \leq 0.5$). It is used only if the selected wavelet transform algorithm uses the FFT.

RECONS/WAVE

RECONS/WAVE Wavelet Rec_Image

Reconstructs an image from its wavelet transform. If the wavelet transform has been computed with the pyramidal algorithm in the direct space, the reconstruction is iterative.

HEADER/WAVE

HEADER/WAVE Wavelet

Gives information about a wavelet file and write them into keywords. The following keywords are modified:

- OUTPUTI[1] = number of lines of the original image
- OUTPUTI[2] = number of columns of the original image
- OUTPUTI[3] = number of scales of the wavelet transform
- OUTPUTI[4] = algorithm number
- OUTPUTR[1] = Frequency cut-off
- OUT_A = PYR, CUB or IMA

- PYR if it is pyramidal algorithm.
- CUB if there is no reduction of the sampling.
- IMA if the wavelet transform has the same size as the original image.

INFO/WAVE

INFO/WAVE Wavelet

This command gives the following information:

- Which wavelet tranform algorithm has been used.
- Name and size of the image.
- Number of scales.
- Min, Max and standard deviation of each scale.

EXTRAC/WAVE

EXTRAC/WAVE Wavelet Image_Out Scale_Number

Creates an image from a scale of a wavelet transform. The parameter *Scale_Number* defines this scale.

ENTER/WAVE

ENTER/WAVE Wavelet_In Image_Out Scale_Number Wavelet_out

Creates a new wavelet file *Wavelet_out* , by replacing the scale *Scale_Number* of the wavelet *Wavelet_In* by an image.

VISUAL/WAVE

VISUAL/WAVE Wavelet [Visu_Type]

Visualizes a wavelet transform with default parameters. The parameter *Visu_Type* can take the following values (see section 14.5):

- CUB = visualization in a cube. See figures 14.13 and 14.17.
- SYN = creation of one image from the coefficients (figures 14.15,14.19 and 14.20).
- PER = visualization in perspective (figures 14.14 and 14.18).
- PLAN = visualization of each scale in a window.
- CONT = plot one level per scale (figures 14.16).

the default value is PLAN.

VISUAL/CUB

VISUAL/CUBE Wavelet [output_file] [Disp] [Visu_Mode] [Display]

Creates a visualization file and load it in a window if the parameter *Display* equal to Y (Y by default) (see figures 14.13 and 14.17). The default name of the output file is *file_visu.bdf.* The image is loaded in the window number *Disp* (default is 1). *Visu_Mode* can take the value CO or BW (color or black and white).

VISUAL/CONT

VISUAL/CONT Wavelet [Graphic_Number] [Visu_Mode] [Contour_Level]

Plots one contour per scale of the wavelet transform on the graphic window number *Graphic_Number* (default 0). The contour are plotted in color if *Visu_Mode* equal to CO. The contour level L plotted at each scale j is defined by:

$$L = \frac{\sigma_1}{4^{j-1}} * Contour\_Level$$

where σ_1 is the standard deviation of the first scale. The *Contour_Level* default value is 3.

VISUAL/SYNT

VISUAL/SYNT Wavelet [output_file] [Display_Number] [Display]

Creates a visualization file and loads it in a window if the parameter *Display* equal to Y (Y by default) (see figures 14.15,14.19 and 14.20). The default name of the output file is *file_visu.bdf.* The image is loaded in the window number *Display_Number* (default is 1).

VISUAL/PLAN

VISUAL/PLAN Wavelet [Display]

Creates an image from each scale of the wavelet transform, and displays the scales in windows if the parameter *Display* equal to Y (Y by default). The names of the created images are *scale_1.bdf* for the first scale, *scale_2.bdf* for the second, ... The size of the window is limited to 512. If the image size is greater, scrolling can be done in the window by using the command VIEW/IMAGE.

VISUAL/PERS

VISUAL/PERS Wavelet [out_file] [Disp] [Visu_Mode] [Incr] [Thres] [Display]

Visualizes the wavelet transform of an image in perspective. A file is created and loaded in the window if the parameter *Display* equal to Y (Y by default) (figures 14.14 and 14.18). The default name of the output file is *file_visu.bdf.* The image is loaded in the window number *Disp* (default is 1). *Visu_Mode* can take the value CO or BW (color or black and white). *Incr* is a parameter which defines the number of lines of the image used. If *Incr* = 3, only on line on 3 are used (default value is 1). *Threshold* is a parameter which defines the maximum value which is taken into account. All the values superior to this maximum are set to the maximum. The maximum value M_j at the scale j is:

$$M = \sigma_j * \text{Threshold}$$

The default value is 5.

FILTER/WAVE

FILTER/WAVE I_In I_Out [Algo] [T_Filter] [Iter_Nbr] [N_Scale] [N_Sigma] [Noise]

Filters an image in the wavelet space (see section 14.6). The used algorithm for the transform depends on the parameter *Algo* (see TRANSF/WAVE). *T_Filter* defines the type of filtering. It can take the values 1,2,3,4 (default value is 1):

1. Thresholding (see section 14.6.4).
2. Hierarchical Thresholding (see section 14.6.5).
3. Hierarchical Wiener Filtering (see section 14.6.3).
4. Multiresolution Wiener Filtering (see section 14.6.2).

If we threshold (*T_Filter* = 1 or 2), the reconstruction can be done iteratively, and the parameter *Iter_Nbr* specifies the number of iterations (by default it is 1, no iteration). *N_Scale* is the number of scales (default value is 4). *N_Sigma* is used only if we threshold. We consider that at a given scale j, the significant level L is:

$$L = \text{N\_Sigma}.\sigma_j$$

where σ_j is the standard deviation of the noise at the scale j. *Noise* is the standard deviation of the noise in the image. If *Noise* equal to 0 (0 is the default value), the standard deviation is estimated automatically in the program from the histogram of the image by a 3σ clipping.

COMPAR/WAVE

COMPAR/WAVE Imag_1 Imag_2 [N_Scal] [N_Sigma] [T_Cor] [T_Snr] [Disp] [Init]

Compares two images in the wavelet space (see section 14.7). At each scale the signal to noise ratio and the correlation is calculated. The results are stored in tables *T_Cor* and *Tab_Snr* (the default names for the two tables are "cmp_correl.tbl" and "cmp_snr.tbl"). *N_Scal* defines the number of scales for the wavelet transform (default is 3). *N_Sigma* is a parameter which allows us to select the wavelet coefficients which are taken into account in the comparison. Only the wavelet coefficients W_j of the first image who verifies the following relation are taken into account:

$$W_j > \text{N\_Sigma}.\sigma_j$$

where σ_j is the standard deviation of the scale j. The default value is 0, this means that all the wavelet coefficients are used. If *Disp* equal to Y (yes), the results are plotted in two graphics window (Y is the default value) by calling the two procedures PLOT/COR and PLOT/SNR. If *Init* equal to Y, the tables *T_Cor* and *Tab_Snr* are initialized, and if *Init* equal to N, columns are added to the tables. That allows us to compare several images to a reference image.

PLOT/SNR

PLOT/SNR [Tab_Snr]

Plots the SNR table resulting from the comparison. *Tab_Snr* is the table which contains the comparison result concerning the signal to noise ratio. The default value is "cmp_snr.tbl".

PLOT/COR

PLOT/COR [Tab_Correl]

Plots the correlation resulting from the comparison. *Tab_Correl* is the table which contains the comparison result concerning the correlation. The default value is "cmp_correl.tbl"

DIRECT/WAVE

DIRECT/WAVE Imag_In Psf Imag_Out [Nb_Scales] [γ_1,γ_2,γ_3,...]

Deconvolves an image by the method described in section 14.8.3. γ_j are generally chosen such that $\gamma_j < \gamma_{j+1}$ because the regularization has to be stronger on high frequencies. The deconvolution is done by a division in the Fourier space. If all γ_j equal to 0, it corresponds to the Fourier quotient method. *Nb_Scales* defines the number of scales used in the wavelet transform. *Imag_In* is the file name of the image to deconvolve, *Psf* is the file name of the point spread function, and *Imag_Out* is the file name of the deconvolved image.

CITTERT/WAVE

CITTERT/WAVE Im_In Psf Im_Out [Resi] [Scal, Iter] [N_Sig, Noise] [Eps] [Fwhm]

Deconvolves an image by the method described in section 14.8.4. *Resi* is the file name of the output residual (default value is *residual.bdf*). *Noise* is the standard deviation of the noise in the image. If *Noise* equals to 0 (which is the default value), an estimation of the noise is done automatically from a 3σ clipping. *N_Sig* is the parameter used to define the level of significant structure in the wavelet space (4 is the default value). *Eps* is the convergence parameter (0.001 is the default value). *Iter* is the maximum number of iterations allowed in the deconvolution. *Fwhm* (Full Width at Half Maximum) allows us to limit the resolution in the restored image. The default value is 0 (no limitation).

GRAD/WAVE

GRAD/WAVE Im_In Psf Im_Out [Resi] [Nb_Scales] [N_Sigma, Noise] [Eps] [Max_Iter]

Deconvolves an image by the method described in section 14.8.4.

LUCY/WAVE

LUCY/WAVE Im_In Psf Im_Out [Resi] [Nb_Scales] [N_Sigma, Noise] [Eps] [Max_Iter]

Deconvolves an image by the method described in section 14.8.4.

TUTORIAL/WAVE

TUTORIAL/WAVE

Visualizes the wavelet transform of the galaxy NGC2997 with several algorithms.

Two commands are proposed for one dimensional signals:

TRAN1D/WAVE

TRAN1D/WAVE Im_In Wave_Out [Num_Trans] [Num_Line] [Channel] [Nu]

Computes the one dimensional wavelet transform of a spectrum or a line of an image. *Im_In* is the input image and *Wave_Out* is the output wavelet transform. The wavelet transform of a one dimensional signal is an image. *Num_Trans* is the wavelet transform chosen. 6 transforms are possible:

1. French hat
2. Mexican hat
3. *à trous* algorithm with a linear scaling function
4. *à trous* algorithm with a B1-spline scaling function
5. *à trous* algorithm with a B3-spline scaling function
6. Morlet's transform.

in case of Morlet's transform, the wavelet transform is complex. The modulus of the transform is stored in the first part of the output image, and the phase in the second part. Default value is 2. *Num_Line* is the line number in the input image which will be used (1 is the default value). *Channel* is the number of channels per octave (the default value is 12). This parameter is not used in the *à-trous* algorithm because *à-trous* algorithm is a diadic algorithm (1 channel per octave). *Nu* is the Morlet's parameter and is only used with Morlet's transform.

REC1D/WAVE

REC1D/WAVE Wave_In Im_Out [Num_Trans] [Num_Line] [Channel] [Nu]

reconstructs a one dimensional signal from its wavelet transform. *Wave_In* is the input image wich contains the wavelet transform, and *Im_Out* is the reconstructed signal. It is an image with one line. The reconstruction parameters must be the same as the wavelet transform parameters.

Bibliography

[1] H.-M. Adorf, "HST Image Restoration - Recent Developments", in P. Benvenuti and E. Schreier, Eds., Science with the Hubble Space Telescope, European Southern Observatory, 227–238, 1992.

[2] M. Antonini, M. Barlaud and P. Mathieu, "Image coding using vector quantization in the wavelet transform domain", *Proc. ICASSP*, Albuquerque USA, 1990.

[3] A. Bijaoui, "Algorithmes de la Transformée en Ondelettes, Application à l'imagerie astronomique", *Ondelettes et Paquets d'Ondes*, Inria, Rocquencourt, 1991.

[4] P.J. Burt et A.E. Adelson, "The Laplacian pyramid as a compact image code", IEEE Trans. on Communications, 31, pp. 532-540, 1983.

[5] J. Cohen, Astroph. Journal, 101 , 734, 1991.

[6] T.J. Cornwell, "Image Restoration", *Proc. NATO Advanced Study Institute on Diffraction-Limited Imaging with Very Large Telescopes*, 273–292, Cargèse, 1988.

[7] C.H. Chui, "An Introduction to Wavelets", *Wavelet Analysis and its Application*, Academic Press, Harcourt Brace Jovanovich, Publisher, 1992.

[8] A. Cohen, I. Daubechies, J.C. Feauveau, "Biorthogonal Bases of Compactly Supported Wavelets", *Comm. Pur. Appl. Math.*, Vol. 45, pp 485–560, 1992.

[9] I. Daubechies, "Orthogonal Bases of compactly Supported Wavelets", *Comm. Pur. Appl. Math.*, Vol. 41, pp 909–996 , 1988.

[10] I. Daubechies, *Ten lectures on Wavelets*, Philadelphia, 1992.

[11] J.C. Feauveau, "Analyse Multirésolution par Ondelettes non Othogonales et Bancs de Filtres Numériques", Thèse de Doctorat de l'université Paris Sud, 1990.

[12] B.R. Frieden, "Image Enhancement and Restoration", *Topics in Applied Physics*, Springer-Verlag Berlin Heidelberg New York, Vol. 6, pp 177–249, 1975.

[13] D. Gabor, "Theory of communication", Journal of I.E.E. **93** pp 429-441, 1946.

[14] N.P. Galatsanos and A.K. Katsaggelos, "Methods for Choosing Regularization Parameter and Estimating the Noise in Image Restoration and their Relation", IEEE Trans. on Image Processing, July 1992.

[15] G.H. Golub, M. Heath and G. Wahba, "Generalized Cross-Validation as a Method for Choosing good ridge Parameter", *Technometrics* 21 (2), 215-223, 1979.

[16] P. Goupillaud, A. Grossmann, J. Morlet, "Cycle-octave and related transforms in seismic signal analysis", *Geoexploration*, 23, 85-102, 1984-1985.

[17] A. Grossmann and J. Morlet, "Decomposition of Hardy functions into square integrable wavelets of constant shape", *SIAM J. Math. Anal*, Vol. 15, pp 723–736, 1984.

[18] E. L. Hall, "Image Enhancement and Restoration", *Computer Image Processing and Recognition*, Computer Science and Applied Mathematics, p 225, 1979.

[19] M. Holschneider, R. Kronland-Martinet, J. Morlet et Ph. Tchamitchian," The à trous Algorithm", *CPT-88/P.2215*, Berlin, pp 1–22, 1988.

[20] M. Holschneider, P. Tchamitchian, *Les ondelettes en 1989*, PG Lemarié, Springer Verlag, Berlin, p. 102, 1990.

[21] L. Landweber, "An iteration formula for Fredholm integral equations of the first kind", *Am. J. Math.*, Vol. 73, 615–624, 1951.

[22] J. Littlewood, R. Paley, *Jour. London Math. Soc.*, Vol. 6, p. 230, 1931.

[23] L. B. Lucy, "An Iteration Technique for the Rectification of Observed Distributions", Astron. Journal, 79, 745-754, 1974.

[24] A.K. Katsaggelos, Ed., *Digital Image Restoration*, Spinger-Verlag, 1991.

[25] S. Mallat, "A Theory for Multiresolution Signal Decomposition: The Wavelet Representation", *Proc. IEEE Trans on Pattern Anal. and Math. intel.*, Vol. 11, No 7, 1989.

[26] Y. Meyer, *Wavelets*, Ed. J.M. Combes et al., Springer Verlag, Berlin, p. 21, 1989.

[27] Y. Meyer, *Ondelettes*, Ed. Hermann, Paris, 1990.

[28] Y. Meyer, "Méthodes temps-fréquence et méthode temps-échelle en traitement du signal et de l'image", *Proc. Ondelettes et Paquets d'Ondes*, Inria, Rocquencourt, 1991.

[29] Y. Meyer, *Ondelettes: Algorithmes et Applications*, A. Colin, Paris, 1992.

[30] R. Murenzi, *Wavelets*, eds. J.M. Combes, A. Grossman, P. Tchmitchian, Springer Berlin, Heidelberg, New-York, 1988.

[31] M.B. Ruskai, G. Beylkin, R. Coifman, I. Daubechies, S. Mallat, Y. Meyer, and L. Raphael, *Wavelets and their Applications*, Jones and Barlett Publishers, Boston, 1992.

[32] C.E. Shannon, *Bell System Tech.*, Vol. 27, p.379, 1948.

[33] M.J. Shensa, "Discrete Wavelet Transforms: Wedding the à trous and Mallat Algorithms", *IEEE Rrans. on Signal Process.*, Vol. 40, 10, pp2464–2482, 1992.

[34] M.J.T. Smith and T.P. Barnwell, "Exact reconstruction technique for tree structured subband coders", *IEEE ASSP*, Vol. 34, pp. 434–441, 1988.

[35] J.L. Starck, and A. Bijaoui, "Filtering and Deconvolution by the Wavelet Transform", to appear in *Signal Processing.*

[36] J.L. Starck, A. Bijaoui, B. Lopez, and C. Perrier, "Image Reconstruction by the the Wavelet Transform Applied to Aperture Synthesis", to appear in *Astron. Astrophys.*, 1993.

[37] J.L. Starck, A. Bijaoui, "Multiresolution Deconvolution", submitted to *JOSA A*, 1993.

[38] J.L. Starck, "Analyse en Ondelettes et Imagerie à Haute Résolution Angulaire", Thèse de Doctorat de l'université de Nice, 1992.

[39] J.L. Starck and F. Murtagh, "Richardson-Lucy Image Restoration with Noise Suppression Based on the Wavelet Transform", ESO Data Analysis Workshop, Germany, 1993.

[40] A.N. Tikhonov, and V. Y. Arsenin, "Solution of Ill-Posed Problems", Winston, Washington, D.C., 1977.

[41] P.H. Van Cittert, *Z. Physik*, Vol. 69, p. 298, 1931.

Chapter 15

The Data Organizer

15.1 Introduction

Before being able to actually reduce and analyse a new set of observations, the observer has to prepare, sort out and arrange the data. That is, for instance, make a first quality check, classify data according to a set of rules and associate with each science frame a set of relevant calibration frames . This task can be cumbersome because of the complexity of the instruments and the large number and the diversity of the data files they produce. For instance, the EMMI instrument mounted on the New Technology Telescope allows a wide range of observing modes from wide-field imaging to high-dispersion spectroscopy, including long-slit, multiple-object spectroscopy and a dichroic mode where spectra are taken simultaneously in the blue and in the red arm of the instrument. The FITS files that are produced contain more than 50 different keywords, and making sense of this information without a proper tool may be very difficult.

The Data Organizer is built entirely on existing capabilities of the MIDAS Table File System. Therefore the astronomer does not have to learn any new computer jargon, change environments, or convert data formats. The output files created by the Data Organizer are MIDAS Tables and can therefore be used by any reduction package.

The concept of a Data Organizer tool is new and this first implementation may be subject to revisions as experience with processing of large amounts of data is obtained.

15.2 Overview of the Data Organizer

The current implementation of the Data Organizer consists of 6 commands which are listed in Table 15.2. In order to be able to execute these commands the context DO should be enabled first. This can be done using the command SET/CONTEXT DO.

15.3 The Observation Summary Table

The Data Organizer uses as input a list of FITS files or MIDAS images as well as a list of MIDAS descriptors which are considered to be relevant (e.g., exposure time, telescope

Command	Description
TUTORIAL/DO	on line tutorial
CREATE/OST	create an Observation Summary Table (OST)
CREATE/CRULE	create a classification rule for an OST
CLASSIFY/IMAGE	classify files by applying one or more classification rules
ASSOCIATE/IMAGE	associate suitable calibrations frames with scientific exposures
GROUP/ROW	group the rows of a table by the value of one of its column

Table 15.1: DO commands

setting, instrument mode). Each of these descriptors is mapped into one column of a table that is called the Observation Summary Table (OST), and the corresponding information for a given input file is stored into one of its rows.

15.3.1 Mapping of FITS keywords into MIDAS descriptors

The Data Organizer expects as initial input a list of MIDAS descriptors. These descriptors are the result of a translation of the FITS keywords following a scheme described in Vol. A, chapter 7 and automatically performed by MIDAS command INTAPE/FITS. For instance, the FITS keyword 'OBJECT' is translated into the MIDAS descriptor 'IDENT', the contents of the keyword 'EXPTIME' is stored into the 7th element of the descriptor 'O_TIME' and the ESO hierarchical keyword 'HIERARCH ESO GEN EXPO TYPE' is translated into the descriptor '_EGE_TYPE'

15.3.2 The Descriptor Table

The list of pertinent MIDAS descriptors should be stored into a MIDAS table containing the following columns:

```
DESCR_INAME  (Character Column)         contains the list of MIDAS
                                        descriptors to be mapped
                                        into the columns of the OST.

IPOS         (Integer Column)           contains for each descriptor
                                        the position of the element to be
                                        read.

DESCR_ONAME  (Character Column)         contains for each descriptor the
                                        label of the column of the OST
                                        in which its values will be stored.

OTYPE        (Character Column)         contains  for each descriptor the
                                        type of the column of the OST in
```

```
which its values will be stored.
I (integer), R(eal), D(double
precision), C*n (character string)
```

The following columns will be automatically created in the OST:

1. :FILENAME (containing the frame name)
2. :MJD (containing the Modified Julian Date of the exposure).

If a descriptor contains more than one element, the one at the position defined in the column :IPOS is taken. The table may be created using the commands CREATE/TABLE, CREATE/COLUMN, and the different parameters may be entered using the Table Editor (EDIT/TABLE). In the future we will supply on anonymous ftp templates for the different ESO instruments.

Table 15.2 shows a descriptor table created for NTT data obtained with the SUSI instrument. (Most of the examples in this document will be based on this set of 14 SUSI files.)

DESCR_INAME	IPOS	DESCR_ONAME	OTYPE
NPIX	1	NPIX_1	C*8
NPIX	2	NPIX_2	C*8
START	1	START_1	R
START	2	START_2	R
IDENT	*	IDENT	C*32
O_POS	1	RA	D
O_POS	2	DEC	D
O_AIRM	*	AIRMASS	R
O_TIME	7	EXPTIME	R
_EI_ID	*	INSTRUMENT	C*8
_EI_MODE	*	INST_MODE	C*8
_EIO2_ID	*	FILTER_NO	C*8
_EIO2_TYPE	*	FILTER_TYPE	C*8
_ED_NAME	*	DET_NAME	C*9
_ED_MODE	*	DET_MODE	C*1
_ED_PIXSIZE	*	DET_PIXSIZE	R
_ED_TEMPMEAN	*	DET_TEMPMEAN	R
_ED_AD_VALUE	*	DET_AD_VALUE	R

Table 15.2: A descriptor Table for SUSI exposures

15.3.3 Creating The Observation Summary Table

FILENAME	IDENT	DET_TEMPMEAN	MJD_LOC
susi0001.mt	BIAS	171.1	0.3587
susi0002.mt	BIAS	172.0	0.3666
susi0003.mt	FF B	172.1	0.3843
susi0004.mt	FF B	172.1	0.3869
susi0005.mt	FF B	172.1	0.3894
susi0006.mt	S295/R/5M	172.1	0.4582
susi0007.mt	S295/B/30M	172.0	0.4673
susi0008.mt	S0504/B/30M	172.0	0.4921
susi0009.mt	MS0955/B/30M	172.3	0.5238
susi0010.mt	BIAS 2d night	172.3	1.3055
susi0011.mt	BIAS 2d night	172.3	1.3087
susi0012.mt	BIAS 2d night	172.3	1.3100
susi0013.mt	FF U	172.1	1.3806
susi0014.mt	FF R 2d night	172.1	1.3898

Table 15.3: An Observation Summary Table (OST)

The command CREATE/OST allows the user to create an Observation Summary Table (OST). The following command:

```
CREATE/OST susi.mt ?  susi_descr susi_ost
```

will process all the FITS files whose names match the pattern susi*.mt, read in each of them the MIDAS descriptors listed in the column :DESCR_INAME of the table susi_descr and store the values of the elements specified in the column :IPOS into the table susi_ost. An extract of this table is shown in Table 15.3.

15.4 Classification of Images

A natural way of getting an overview of the data one has consists of classifying the files into a set of groups. One may for instance want to group the frames according to the exposure type or one may need to put together all the files observed in a given instrument mode.

15.4.1 Creation of the Classification Rules

An instruction for grouping flat field exposures could be: Select all files which have a descriptor IDENT matching one of the substrings 'FF', 'SKYFL', or 'FLAT'. With all

COLUMN	RULE
NPIX_1	
NPIX_2	
START_1	
START_2	
IDENT	*BIAS*
RA	
DEC	
AIRMASS	
EXPTIME	<=1
INSTRUMENT	
INST_MODE	
FILTER_NO	
FILTER_TYPE	
DET_NAME	
DET_MODE	
DET_PIXSIZE	
DET_TEMPMEAN	
DET_AD_VALUE	

Table 15.4: Formulation of a Classification Rule

relevant information being available in an Observation Summary Table, this can be accomplished by supplying a suitable logical expression to the general purpose command SELECT/TABLE. Because observers nomenclatures usually don't follow any standard conventions, the possibility is given to the users to write their own rules using an interface built on top on the Table Editor. This interface enables the user to interactively enter constraints on the existing fields of the OST. Relational operators (>, <, != or =) may be used as well as logical operators (&, |). Values or ranges of values have to be specified ("*" is the wildcard character, "~" ignores the case and the ".." specifies a range of values). Constraints applied to more than one column are ANDed and translated into an expression understandable by the **SELECT** command. This expression is stored into a descriptor of the OST. The command

```
CREATE/CRULE susi_ost bias
```

first creates a temporary table containing two columns labeled respectively :COLUMN and :RULE and then invokes the table editor. The user should then, for instance, formulate a rule for classifying BIAS exposures by entering constraints on columns of the OST (Table 15.4). Finally the command stores the derived selection criterion in the descriptor BIAS of the table susi_ost:

```
IDENT.eq."*BIAS*".AND.:EXPTIME.LE.1
```

This rule would lead to all files being classified as bias exposures which contain the string BIAS and the exposure time of which did not exceed one second. The temporary table file is automatically deleted when the descriptor of the OST has been created.

15.4.2 Classification of images

The classification process (command CLASSIFY/IMAGE) tags each frame with one or more character strings supplied by the user that classify the image in a unique way and will be stored into columns of the OST. For instance scientific exposures observed in SUSI using the filter number 639 may be described' with the two strings "SCI" and "FILT_639". The practical problem is however that the very large number of different possible optical elements mounted on instruments like EMMI/SUSI is not fixed in advance; for example, filters may be added or changed. Clearly, one do not want to write classification rules for all filter numbers or all combinations of grisms and gratings.
We have for those reasons introduced the concept of "*Wild Card Replacement Character*"; This character & may be used to build the output classification string in the following way: *substring&n* will be built by finding the *n-th* occurrence of the $*$ character in the classification rule, reading the content of the corresponding column and appending it to *substring*.

```
Midas 001> CREATE/CRULE susi_ost BIAS
     .. The Table Editor is invoked ..
Midas 002> READ/DESCR susi_ost.tbl BIAS
     :IDENT.EQ."*BIAS*".AND.:EXPTIME.LE.1
Midas 003> CREATE/CRULE susi_ost FF
     .. The Table Editor is invoked ..
Midas 004> READ/DESCR susi_ost.tbl FF
     :IDENT.EQ."FF*"
Midas 005> CREATE/CRULE susi_ost SCI
     .. The Table Editor is invoked ..
Midas 006> READ/DESCR susi_ost.tbl SCI
     :IDENT.NE."*BIAS*".AND.:IDENT.NE."*FF*"
Midas 007> CLASSIFY/IMAGE susi_ost BIAS EXPTYPE BIAS
Midas 008> CLASSIFY/IMAGE susi_ost FF EXPTYPE FF
Midas 009> CLASSIFY/IMAGE susi_ost SCI EXPTYPE SCI
Midas 010> CREATE/CRULE susi_ost OPATH
     .. The Table Editor is invoked ..
Midas 011> READ/DESCR susi_ost.tbl OPATH
     :FILTER_NO.EQ."#*".AND.:FILTER_TYPE.EQ."FILTER*"
Midas 012> CLASSIFY/IMAGE susi_ost OPATH OPATH FILT_&1
```

Table 15.5: MIDAS session for classifying SUSI exposures

Suppose you have created the classification rule WFIB for flagging all exposures obtained in the blue arm of EMMI using the wide-field imaging mode. Now you want to flag these exposures with a character string containing the filter number that has been used. The corresponding descriptor WFIB of the OST may in natural language be: "frame exposed in the blue arm using any filter and no grating" The translated selected criterion looks as follows:

```
:FILTB_ID.EQ."*".AND.:FILTB_TYP.EQ."FILTER*".AND.:GRATB_ID.EQ.""
```

The command

```
CLASSIFY/IMAGE ntt WFIB :OPATH FB&1
```

will flag all files in the OST ntt which satisfy this selection criterion and will store in the column :OPATH the character string obtained by appending the contents of column :FILTB_ID to the string FB. The classification given in the column :OPATH is obviously more convenient to use than the original selection criteria.

```
Note:
This implementation of the sofware expects, for the association process, that
the type of the exposures is stored in a column labeled :EXPTYPE and that frames
classified as scientific exposures are flagged with the string SCI.
```

15.4.3 An example of a Classification Process

We have defined a set of rules for classifying SUSI exposures according to the exposure type (BIAS, FF, SCI) and a set of rules for classifying them according to the optical path. (In this case, this consists simply in classifying the frames according to the filter number that was used.)
The MIDAS session shown in Table 15.5 creates these rules and applies them to our set of SUSI exposures (Table 15.3). The result of the classification process is illustrated in Table 15.6.
For convenience, one may also store all the classification parameters in a table (Table 15.7) and apply a set of classification rules in one step.
Note that instead of the interactive command CREATE/CRULE the equivalent command WRITE/DESCR may be used. The table must contain the following columns:

- :DESCR – character column
  Contains the names of the classification rules descriptors
- :OUTCOL – character column
  Contains the label of the output column of the OST in which the string flagging the exposures satisfying the corresponding rule should be stored.
- :OUTCHAR – character column
  Contains the character string for flagging the rows satisfying the corresponding rule.

In the session mentioned above, one can replace the four calls to the command CLASSIFY/IMAGE by:

```
CLASSIFY/IMAGE susi_ost susi_rule.tbl
```

IDENT	FILTER_NO	EXPTYPE	OPATH
BIAS		BIAS	
BIAS		BIAS	
FF B	#639	FF	FILT_639
FF B	#639	FF	FILT_639
FF B	#639	FF	FILT_639
S295/R/5M	#642	SCI	FILT_642
S295?B/30M	#639	SCI	FILT_639
S0504/B/30M	#639	SCI	FILT_639
MS0955/B/30M	#639	SCI	FILT_639
BIAS 2d night		BIAS	
BIAS 2d night		BIAS	
BIAS 2d night		BIAS	
FF U	#640	FF	FILT_642
FF R 2d night	#642	FF	FILT_642

Table 15.6: Observation Summary Table with file classifications appended in column :EXPTYPE and :OPATH as defined and applied in Table 15.5

DESCR	OUTCOL	OUTCHAR
BIAS	EXPTYPE	BIAS
FF	EXPTYPE	FF
SCI	EXPTYPE	SCI
OPATH	EXPTYPE	FILT_&1

Table 15.7: Classification Table for SUSI exposures (susi_rule.tbl)

15.4.4 Checking the quality of the data using the OST

At this stage of the process, it is already possible to make a preliminary statistical analysis of the observing run without having to inspect the files individually. One can, for instance, plot the r.m.s noise of the flat field exposures against their mean pixel value to check the general consistency of the frames. Or one may want to plot the mean temperature of the detector against the date of the exposures to check whether there are outliers.(Figure 15.1)

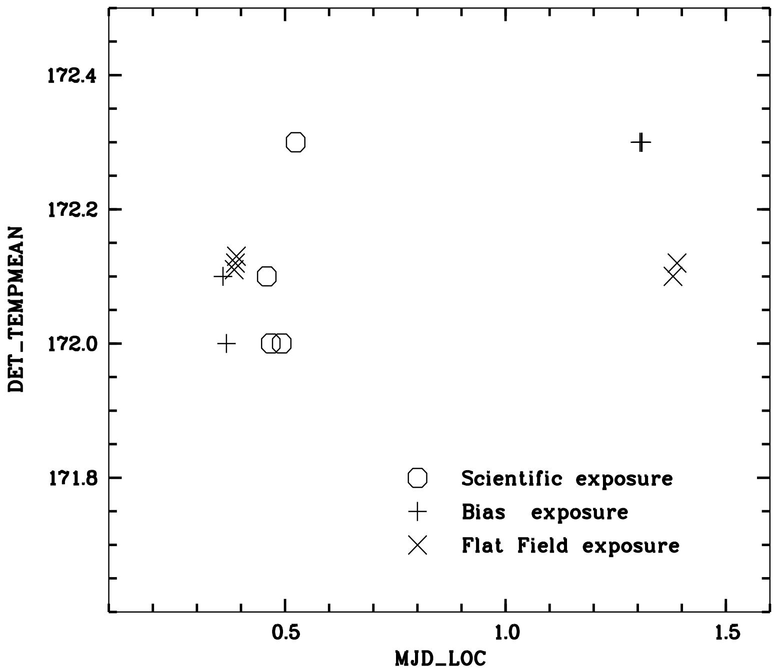

Figure 15.1: Trend analysis of the detector mean temperature with Modified Julian Date

15.5 Association of images

The association of scientific frames with suitable calibration images is achieved by selecting and ranking all calibration frames which for a given scientific exposure match a set of user defined selection criteria. One may expand the search by submitting a "second choice" set of criteria when too few frames match the original one. For instance, this may in natural language be expressed as: "Find for each scientific frame 2 dark exposures which have been observed within the same night. If not found, look for dark exposures observed within 3 days."

15.5.1 Creation of the Association Rules

The two sets of selection criteria ("first choice" and "second choice") have to be entered in a table containing four columns:

- :FUNCTION –character column (max 256 char)
  This column defines the different selection criteria and shall contain mathematical expressions in which the variables may be the columns of the OST. The expression may contain arithmetic as well as mathematical functions. On top of the operators and functions supported by the SELECT/TABLE command, a new function REFVAL is available. REFVAL(:col) will return the value of the column 'col' at the row corresponding at the scientific exposure that is being processed. Suppose

you try, for instance, to associate exposures to the image susi0006.mt (Table 15.3), REFVAL(:MJD_LOC) will be 0.4582.

- :RANGE_1 –character column (max 256 char)
  Contains the ranges defining the "first choice" selection criteria.
- :RANGE_2 –character column (max 256 char)
  Contains the ranges defining the second choice selection criteria.
- :WEIGHT –real column
  Defines the weighting factors for the different selection criteria. These weights are used to rank the frames that satisfy the selection criteria.

15.5.2 An example of selection criteria

Table 15.8 shows a table of association rules that has been created to associate bias exposures with scientific frames. A column labeled :MJD_LOC has been created in the OST and contains a number derived from the Modified Julian Date of the exposure.

- ABS(:MJD_LOC-REFVAL(:MJD_LOC)) is the absolute difference between the *time* of the scientific frame that is being studied and the *time* of a dark exposure
- ABS(:DET_TEMPMEAN-REFVAL(:DET_TEMPMEAN)) is the absolute difference between the mean temperature of the detector when the dark exposure was taken and the mean temperature of the detector when the scientific frame was observed.
- :NPIX_1-REFVAL(:NPIX_1) is the difference between the CCD window sizes in the X-direction of a dark exposure and the scientific frame while :NPIX_2-REFVAL(:NPIX_2) is the difference between the CCD window sizes in the Y-direction of a dark exposure and the scientific frame.

Each row of the column labeled :RANGE_1 defines the acceptable "first choice" ranges for each of the four functions mentioned previously, and the combination of these four criteria defines the "first choice" set of criteria. This may be expressed as:
Given a scientific exposure, select all dark exposures that satisfy:

- The elapsed time between the two exposures is less than 12 hours
- The detector mean temperature did not vary by more than 0.15 degrees between the two exposures
- The two frames have exactly the same size.

The column RANGE_2 defines in the same way the set of "second choice" criteria that can be expressed as:
Given a scientific exposure, select all dark exposures that satisfy:

- The elapsed time between the two exposures is less than 12 hours
- The detector mean temperature did not vary by more than 0.5 degrees between the two exposures
- The two frames have exactly the same size.

The graphs of Figure 15.2 illustrate the association of two dark exposures with each of two scientific images from our observing run: susi0006.bdf and susi0009.bdf. The association process when applying the set of criteria mentioned above will find two suitable candidates for susi0006.bdf satisfying the "first choice" set of criteria. No "first choice" candidates for susi0009.bdf will be found while five "second choice" candidates may be selected. Because more importance is assigned to the time difference (weight=0.8) than to the detector temperature difference (weight=0.2), the files susi0001.bdf and susi0002.bdf will be flagged as the "best" ones. (Note that the differences are computed in the units involved in the columns involved. In the given example they are degrees kelvin and days, respectively.)
A table of rules (Table 15.9) for associating flat-field exposures has been created following the same scheme. One criterion has to be added to the rules mentioned above: One has to make sure that one associates only frames obtained in the same instrument configuration (i.e in this case with the same filter). Figure 15.3 shows that only one flat field can be associated with the frame susi0006.bdf (for clarity, only the FFs obtained with the same filter as the scientific image are numbered) while three suitable flat-fields may be selected for susi0009.bdf

FUNCTION	**RANGE_1**	**RANGE_2**	**WEIGHT**
`ABS(:MJD_LOC-REFVAL(:MJD_LOC))`	`<=0.5`	`<=1`	`0.8`
`ABS(:DET_TEMPMEAN-REFVAL(:DET_TEMPMEAN))`	`<=0.15`	`<=0.5`	`0.2`
`:NPIX_1-REFVAL(:NPIX_1)`	`=0`	`=0`	`0`
`:NPIX_2-REFVAL(:NPIX_2)`	`=0`	`=0`	`0`

Table 15.8: Rules for the association of DARK exposures (cf. Fig. 15.3 and 15.2)

15.5.3 Association of Calibration Exposures

The association process creates an output table that contains the following columns:

FUNCTION	RANGE_1	RANGE_2	WEIGHT
ABS(:MJD_LOC-REFVAL(:MJD_LOC))	<=0.5	<=1	=0.8
ABS(:DET_TEMPMEAN-REFVAL(:DET_TEMPMEAN))	<=0.15	<=0.5	=0.2
:NPIX_1-REFVAL(:NPIX_1)	=0	=0	=0
:NPIX_2-REFVAL(:NPIX_2)	=0	=0	=0
:OPATH	=REFVAL(:OPATH)	=REFVAL(:OPATH)	=0

Table 15.9: Rules for the association of FF exposures (cf. Fig. 15.3 and 15.3)

- A character column labeled :SCI containing the names of the scientific frames. These names are extracted from the column :FILENAME of the OST
- A character column labeled 'exptype' (i.e DARK, FF, etc) containing the names of the associated exposures
- An integer column QUAL_'exptype' (i.e QUAL_DARK, QUAL_FF, etc) containing for each associated frame its quality flag. The calibration matching the "first choice" set of criteria get a quality of 1 while the exposures matching only the "second choice" set of criteria get a quality of 2.

The commands :

```
ASSOCIATE/IMA susi_ost bias bias_rule susi_asso C 2
ASSOCIATE/IMA susi_ost ff ff_rule susi_asso A 2
```

associates with each scientific frame two dark exposures by applying the rules stored in the table bias_rule, two flat field by applying the rules stored in the table ff_rule and stores the result in the table susi_asso (Table 15.10).

SCI	BIAS	QUAL_BIAS	FF	QUAL_FF
susi0006.mt	susi0001.mt	1	susi0014.mt	2
susi0006.mt	susi0002.mt	1		*
susi0007.mt	susi0002.mt	1	susi0003.mt	1
susi0007.mt	susi0001.mt	1	susi0004.mt	1
susi0008.mt	susi0002.mt	1	susi0003.mt	1
susi0008.mt	susi0001.mt	1	susi0004.mt	1
susi0009.mt	susi0001.mt	2	susi0005.mt	2
susi0009.mt	susi0002.mt	2	susi0004.mt	2

Table 15.10: Table of Associated Exposures

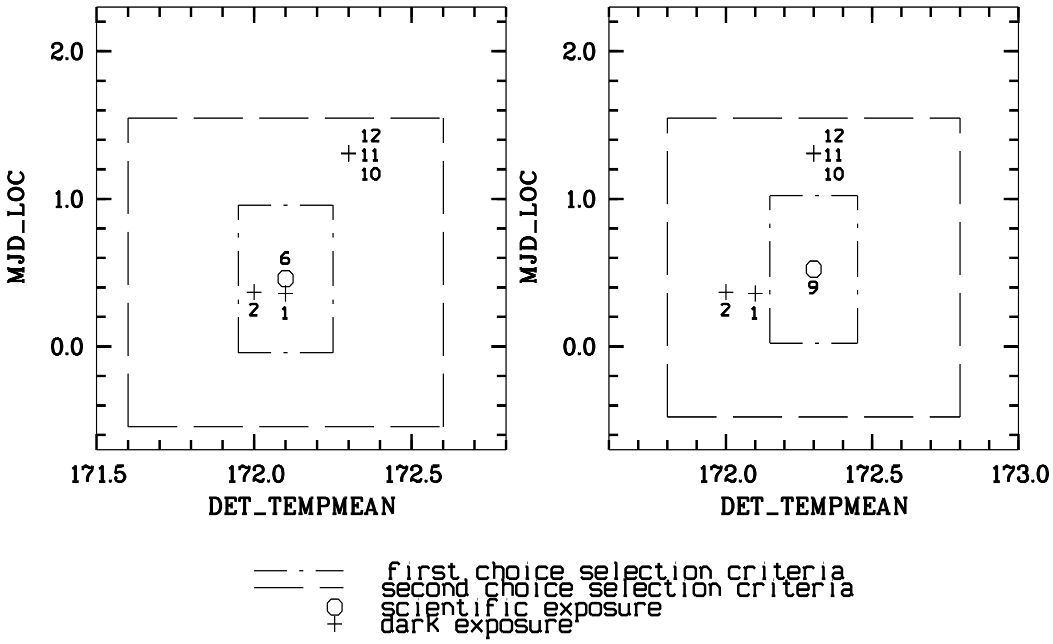

Figure 15.2: Association of DARK exposures with scientific frames

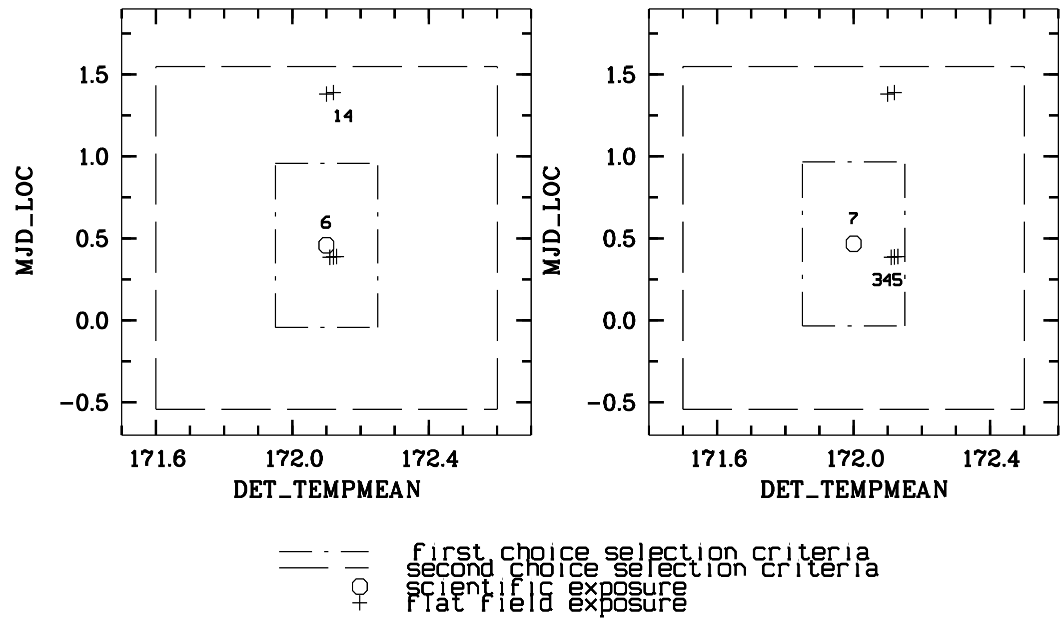

Figure 15.3: Association of FF exposures with scientific frames

15.6 Command Syntax Summary

Table 15.6 lists the commands of the DO context as well as their syntax.

Command	
TUTORIAL/DO	
CREATE/OST	file_specs [file_pref] intable outtable flag
CREATE/CRULE	table descr
CLASSIFY/IMAGE	table descr outcol outchar
	table classtable
ASSOCIATE/IMAGE	ost exptype rule_table outtable [flag] [nexp]
GROUP/ROW	table incol outcol

Table 15.11: DO Command List

Chapter 16

ASTROMET astrometry package

16.1 Introduction

This chapter briefly describes the astrometry context ASTROMET in MIDAS. The applications in this context correspond to Richards West's original POS1 program. The input/output are made compatible with MIDAS, but the algorithm has not been changed (as it proved to be extremely accurate, this would have been masochism).

******* IMPORTANT NOTE *******

Currently, the package has only been tested on system running SunOS. Correct installation and performance on platforms other than those running SunOS is explicitly NOT GUARANTEED. Performance, however, can be tested by executing the test described in the file directory `/midas/$VERS/contrib/astromet/tmp/pos1.exp`. The data for this test consists of two MIDAS tables: `ppm300.tbl` and `xy300.tbl`. Both tables can be found in the directory `/midas/demo/data` and can be retrieved by anonymous ftp.

16.2 Available Commands

The context ASTROMET can be enabled by the command `SET/CONTEXT ASTROMET`. While in the original implementation everything was done in one programme, in the context ASTROMET the software is split into three commands. These commands take care of the following:

- read the measurements and standard stars, and compute the transformation parameters (this step is performed by the command `ASTROMET/TRANSFORM`);
- edit the standard star table to remove/restore some stars (this step is performed by the command `ASTROMET/EDIT`);
- actually compute the converted coordinates (this step is performed by the command `ASTROMET/COMPUTE`).

Steps 1 and 2 can be iterated until a satisfactory result is obtained.

The whole process can also be executed in an iterative way by the command `ASTROMET/POS1`. This commands asks for all input parameters.

16.2.1 ASTROMET/TRANSFORM

This command computes the astrometric transformation parameters for the input data set using a least-square fitting algorithm. The input data is assumed to be two MIDAS tables with fixed columns names. The accuracy has been tested on standard stars, and in real life on many objects including comet Shoemaker-Levy 9, leading to very accurate impact times. `ASTROMET/TRANSFORM` only computes the transformation parameters. Other commands (see below) should be used to actually do the transformations, to edit the data, or to display the residuals.

The first input MIDAS table should contain the coordinates of the standard stars in RA and Dec. It should contain the following columns:

- `:PPM` Integer column (I6) containing the identifier of the standards;
- `:R_A` Real column containing the right ascension in degrees;
- `:DEC` Real column with the declination in degrees;
- `:MAG` Real column with the magnitude;
- `:PMA` Real column with the proper motion in RA in arcsec/year (NOT time_second);
- `:PMD` Real column with the proper motion in Dec in arcsec/year.

A possible way to produce this table is to use StarCat, with the MIDAS table output option, in the PPM catalogue.

The second MIDAS table should contain the XY measurements of the standards and should have, as a minimum, the following columns:

- `:IDENT` Character columns (A16) which must contain a character string; leading letters (*e.g.* PPM) are ignored;
- `:XCEN` Real column to contain the X measurement;
- `:YCEN` Real column to contain the Y measurement.

The remaining parameters are: the coordinate of the center, the epoch of the plate and the reference epoch of the catalogue, two instruments flag, a tolerance on the measurements, and a selection flag. These parameters are described in the help file of the command (`HELP ASTROMET/TRANSFORM`).

16.2.2 ASTROMET/EDIT

With this command the user can delete and undelete the astrometric standards used for the transformation, and optionally displays (slowly) the residuals. Input is the MIDAS table with the coordinates of the standard stars.

After having displayed the residuals, the programs ask for the identification of a star to be deleted/restored. This must be the sequence Number in the MIDAS table and NOT the identifier of the star. To finish the editing session type 0.

16.2.3 ASTROMET/COMPUTE

This command converts the measured X, Y coordinates to RA and Dec. It uses the transformation parameters computed by the command `ASTROMET/TRANSFORM`. The input is a table containing the XY coordinates which must have been processed previously by `ASTROMET/TRANSFORM` in order to have the descriptors with the transformation parameters set.

The user can choose among various options for doing the coordinate transformation. The output table will contain the computed coordinates. This table can be used as standard stars table in case a multi step astrometric calibration is needed.

`ASTROMET/COMPUTE` computes the RA,Dec coordinates from X,Y coordinates directly from the transformation parameters, while it obtains X,Y from the RA,Dec ones in an iterative manner. Also note that the equinox of the computed coordinates is the same as the standard star catalogue (which is NOT necessarily the same as its epoch) and is 2000.0 in case the PPM catalogue is used.

16.2.4 ASTROMET/POS1

This command corresponds to the original POS1 program and does an interactive astrometric reduction. The command can also be used to test the performance of the package and as a tutorial.

16.3 Command Overview

ASTROMET/COMPUTE	mes option out trail
	convert coordinates from the measured XY to RA, Dec or vice versa
ASTROMET/EDIT	std plot
	delete/undelete the astrometric standards
ASTROMET/POS1	
	interactive procedure for the POS1 astrometry package
ASTROMET/TRANS	std mes center pla,cat schmidt,blink tol xterm,yterm std
	compute the astrometric transformation parameters of a data

Table 16.1: ASTROMET commands

Appendix A

Command Summary

Below is an alphabetical list of all basic MIDAS commands. The following abbreviations have been used where appropriate for the command parameters:

input image:	*infr infr_1, infr_2...*
output image:	*outfr outfr_1, outfr_2...*
input tables:	*intab intab_1, intab_2...*
output tables:	*outtab outtab_1, outtab_2...*
display channel:	*chan chan_1*
device:	*device*

A.1 Core Commands

`@@ proc` *[par1] ... [par8]*
execute a MIDAS procedure
`@ proc` *[par1] ... [par8]*
execute a procedure in MID_PROC (MIDAS core procedures)
`@a proc` *[par1] ... [par8]*
execute a procedure in APP_PROC (MIDAS application procedures)
`@s proc` *[par1] ... [par8]*
execute a procedure in STD_PROC (MIDAS standard reduction procedure)

`ADD/ACAT` *[cat_name] frame_list*
add entries to an ASCII file catalog
`ADD/FCAT` *[cat_name] file_list [lowstr,histr]*
add entries to a fitfile catalog
`ADD/ICAT` *[cat_name] frame_list [lowstr,histr]*
add entries to an image catalog
`ADD/TCAT` *[cat_name] table_list [lowstr,histr]*

add entries to a table catalog

`ALIGN/CENTER` *inframe refframe incent_x,_y refcent_x,_y*
compute start coordinates for inframe to match with refframe center

`ALIGN/IMAGE` *intab reftab [option] [overlay_flag] [residual_flag]*
compute transformation coefficients for two rotated images

`APPLY/CONVERSION` *IMTB ima tab threshold*
convert a "mask" image to a table

`APPLY/CONVERSION` *TBIM tab ima npx1,npx2 sta1,sta2,stp1,stp2 bg,fg*
convert a table to a "mask" image

`APPLY/EDGE` *inframe outframe [thresh]*
do edge detection on an image

`APPLY/MAP` *outframe = inframe mapframe control_flags*
use an image frame like a Lookup Table

`APPLY/THIN` *inframe outframe*
apply thinning algorithm

`ASSIGN/DEFAULT`
assign default devices

`ASSIGN/DISPLAY` *[dev] [file_name]*
define output device for display

`ASSIGN/GRAPHICS` *[device] [option]*
define the graphic device for plot output

`ASSIGN/INPUT` *[dev] [file_name]*
define input device for writing

`ASSIGN/PRINT` *[dev] [file_name]*
define output device for printing

`AVERAGE/AVERAGE` *[in_specs] [out_specs] [out_opt] [draw_flag]*
compute average over subimage

`AVERAGE/COLUMN` *out = in [start,end] [SUM]*
average image columns

`AVERAGE/IMAGES` *out = in_specs [merge] [null] [av_option] [dat_intval]*
average images

`AVERAGE/KAPPA` *[in_specs] [out_specs] [out_opt] [draw_flag] [no_iter]*
compute average (kappa-sigma clipping) over subimage

`AVERAGE/MEDIAN` *[in_specs] [out_specs] [out_opt] [draw_flag]*
compute average (median value) over subimage

`AVERAGE/ROW` *out = in [start,end] [SUM]*
average image rows

`AVERAGE/WEIGHTS` *out = in_specs [merge] [null] [av_option] [dat_intval]*
average weighted images

`AVERAGE/WINDOW` *out = in_specs [meth] [bgerr,snoise]*
compute average of (consistent) pixel values

`BLINK/CHANNEL` *[cha1,cha2,..] [intval] [area_option]* blink between Image Display channels

`BYE` *[proc]*
terminate a MIDAS *session + return to the host system*

`CENTER/GAUSS` *[in_specs] [out_specs] [out_opt] [curs_specs] [wsize] [zw_option] [invert_flag]*
find intensity weighted center

`CENTER/IQE` *[in_specs] [out_specs] [out_opt] [curs_specs] [wsize] [zw_option] [invert_flag]*
find intensity weighted center + get angle of major axis

`CENTER/MOMENT` *[in_specs] [out_specs] [out_opt] [curs_specs] [wsize] [zw_option] [invert_flag]*
find intensity weighted center

`CHANGE/DIRECTORY` *direc*
change the default (current) directory for MIDAS

`CLEAR/ACAT`
deactivate the ASCII file catalog

`CLEAR/ALPHA`
clear the alpha-numerics memory of the image display

`CLEAR/BACKGROUND` *[no_secs]*
put Midas session into "foreground" mode

`CLEAR/BUFFER`
clear the command buffer + reset command numbers

`CLEAR/CHANNEL` *[chanl]*
clear + initialize memory channel

`CLEAR/CONTEXT` *[context]*
remove command definitions of a context

`CLEAR/DISPLAY`
reset image display

`CLEAR/FCAT`
disable automatic catalog functions for fit files

`CLEAR/GRAPHIC`
erase the screen of the graphic window or terminal

`CLEAR/ICAT`
disable automatic catalog functions for image frames

`CLEAR/ITT` *[chanl]*
bypass ITT on display of memory

`CLEAR/LUT`
bypass LUT *on image display*

`CLEAR/OVERLAY`
disable graphics/overlay plane of display

`CLEAR/SCROLL` *[chanl]*
reset scroll values

`CLEAR/TCAT`
  disable automatic catalog functions for table files
`CLEAR/ZOOM` *[chanl]*
  clear zoom
`CLOSE/FILE` *file_id*
  close an ASCII *file*
`COMPUTE/AIRMASS` *frame [long] [lat]*
`COMPUTE/AIRMASS` *alpha delta ST [exptime] [long] [lat] [date] [UT]*
  compute airmass (from sec z)
`COMPUTE/BARYCORR` *date UT alpha delta [longitude] [latitude]*
`COMPUTE/BARYCORR` *table.tbl [longitude] [latitude]*
`COMPUTE/BARYCORR` *image alpha delta [longitude] [latitude]*
  correct universal times and radial velocities to center of sun or barycenter of solar system
`COMPUTE/COLUMN` *res_frame.column = arithmetic_expression*
  do arithmetics on columns of an image
`COMPUTE/DIMAGE` *[outspec =] expression*
  compute arithmetic expression
`COMPUTE/DPIXEL` *[outspec =] expression*
  compute expression on pixel basis
`COMPUTE/DPLANE` *res_frame.plane = arithmetic_expression*
  do arithmetics on planes of images
`COMPUTE/HISTOGRAM` *result = table col [bin [min [max]]]*
  table-to-image or table-to-table histogram transformation
`COMPUTE/IMAGE` *[outspec =] expression*
  compute arithmetic expression
`COMPUTE/KEYWORD` *key = arithmetic_expression*
  compute values of a keyword
`COMPUTE/PIXEL` *[outspec =] expression*
  compute expression on pixel basis
`COMPUTE/PLANE` *res_frame.plane = arithmetic_expression*
  do arithmetics on planes of images
`COMPUTE/PRECESSION` *alpha delta equinox0 equinox1*
`COMPUTE/PRECESSION` *table.tbl equinox0 equinox1*
  precess equatorial coordinates from one epoch to another
`COMPUTE/REGRESSION` *table column = name[(ind)] [d_type]*
  compute result of a regression
`COMPUTE/ROW` *res_frame.row = arithmetic_expression*
  do arithmetics on rows (lines) of images
`COMPUTE/ST` *date UT [longitude]*
`COMPUTE/ST` *table.tbl [longitude]*
`COMPUTE/ST` *image [longitude]*
  calculate geocentric Julian date (JD) and local mean sidereal time (ST) from civil date and universal time (UT)

`COMPUTE/TABLE` *table column = expression*
compute arithmetic or string operations on table columns
`COMPUTE/UT` *date ST [longitude]*
`COMPUTE/UT` *table.tbl [longitude]*
`COMPUTE/UT` *image [longitude]*
calculate geocentric Julian date (JD) and universal time (UT) from civil date and local mean sidereal time (ST)
`COMPUTE/WEIGHTS` *input_specs [window_specs]*
determine weights for command AVERAGE/WEIGHTS
`COMPUTE/XYPLANE` *result_cube = expression*
compute arithmetic expression on xy_planes of cubes
`COMPUTE/XZPLANE` *result_cube = expression*
compute arithmetic expression on xz_planes of cubes
`COMPUTE/ZYPLANE` *result_cube = expression*
compute arithmetic expression on zy_planes of cubes
`CONNECT/BACK_MIDAS` *unit wait_specs b_char method*
connect "command syntax" to another MIDAS
`CONVERT/COORDS` *image coord_string [dsplay_flag]*
frame pixel ro world coordinates conversion (and vice versa)
`CONVERT/TABLE` *image = table x[,y] z refima [method] [par]*
`CONVERT/TABLE` *image = table x[,y] refima FREQ*
table to image conversion
`CONVOLVE/IMAGE` *frame psf result*
convolve image with point spread function
`COPY/AD` *ASCII_file dest_frame dest_desc [max_len]*
copy contents of ASCII *file to character descriptor of dest_frame*
`COPY/DA` *source_frame source_desc ASCII_file*
copy descriptor of source frame to ASCII *file*
`COPY/DD` *source_frame source_desc dest_frame dest_desc*
copy descriptors of source frame to destination frame
`COPY/DIMA` *source_frame source_desc dest_frame*
copy descriptor of source frame to new image
`COPY/DISPLAY` *[out_dev] [stop_flg] [ITTdef] [LUTnam] [prflag] [prmode]*
make a hardcopy of the display on output_device
`COPY/DK` *source_frame source_desc dest_key*
copy descriptor of source frame to keyword
`COPY/DKEY` *source_frame source_desc dest_key*
copy descriptor of source frame to keyword
`COPY/GRAPHICS` *[device] [plotfile]*
copy the existing plot file to the graphic device
`COPY/ID` *source_frame dest_frame dest_desc*
copy image data to descriptor of destination frame
`COPY/II` *source_frame dest_frame dest_format flags*
copy source frame to destination frame

`COPY/IT` *inframe outable [column]*
copy image into table
`COPY/KD` *source_key dest_frame dest_desc*
copy keyword to descriptor of destination frame
`COPY/KEYWORD` *source_key dest_key [M_unit]*
copy keywords of same type
`COPY/KI` *source_key dest_frame*
copy keyword to new frame
`COPY/KT` *keyword table [column ...] element*
copy keyword into table element
`COPY/LSDD` *list source_frame dest_frame*
copy list of descriptors of source frame to descriptors of dest_frame
`COPY/LSDK` *list source_frame*
copy list of descriptors of source frame to keywords
`COPY/LSKD` *list dest_frame*
copy list of keywords to descriptors of destination frame
`COPY/OVERLAY` *[chanl] [append_flag]*
copy all graphics and text from the overlay plane to image channel
`COPY/PSCR` *image [PS_file] [options] [prnt_flags] [scales] [LUT_specs]*
convert MIDAS *images (bdf-files) to PostScript format*
`COPY/TABLE` *intable outable [organization]*
copy source table to destination table
`COPY/TEIMA` *table column rows [npix] [coor] [prefix]*
Convert a 3D-table element to a MIDAS *image file*
`COPY/TI` *intable outimage*
copy table into image
`COPY/TK` *table [column ...] element keyword*
copy table element into keyword
`COPY/TT` *intable incolumn [outable] outcolumn*
copy a table column to an other existing table
`COPY/ZOOM` *[out_dev] [stop_flg] [ITTnam] [LUTnam] [prflag] [prmode]*
make a copy of the zoom window on output device
`CREATE/ACAT` *[cat_name] [dir_spec]*
create a catalog of files in the current directory
`CREATE/COLUMN` *table column [unit] [format] [type]*
create a table column
`CREATE/COMMAND` *comnd text*
create a "user" command
`CREATE/CURSOR` *[dspid] [wind_specs] [Xstation]*
create a cursor window
`CREATE/D_COMMAND` *comnd text*
create a directory "user" command
`CREATE/DEFAULT` *comnd def1 def2 ... def8*
create special defaults for MIDAS command

`CREATE/DISPLAY` *[dspid] [dspinfo] [meminfo] [alph_flag] [gsize] [Xstation]*
    create a display window
`CREATE/FCAT` *[catname] [dir_spec]*
    create a catalog of fit files in the current directory
`CREATE/FILTER` *frame [dim_specs] [frame_specs] [filt_type] [coefs]*
    create filter frame
`CREATE/GRAPHICS` *[graph_id] [graph_spec] [gsize] [Xstation]*
    create a graphics window
`CREATE/ICAT` *[catname] [dir_spec] [descr]*
    create a catalog of images in the current directory
`CREATE/IMAGE` *frame [dim_specs] [frame_specs] [func_type] [coefs]*
    create an image
`CREATE/LUT` *LUT_table H_specs S_specs I_specs cyclic_flag*
    create a colour lookup table
`CREATE/RANDOM` *name [dims] [starts,steps] [func_type] [coefs] [seed]*
`CREATE/RANDOM` *name = ref_frame [func_type] [coefs] [seed]*
    create a random image
`CREATE/ROW` *table row_position number_of_rows*
    add one or several rows at a given position of a table
`CREATE/TABLE` *table ncol nrow file [format_file] [organization]*
    create a table
`CREATE/TCAT` *[catname] [dir_spec] [descr]*
    create a catalog of tables in the current directory
`CREATE/VIRTUAL` *virtual table*
    create a virtual table from a physical table
`CREATE/ZOOM` *[dspid] [wind_specs] [Xstation]*
    create a zoom window
`CUTS/IMAGE` *frame [cut_specs]*
    display or set low + high cut values of an image frame

`DEBUG/MODULE` *[low_lev,hi_lev] [switch]*
    run MIDAS modules in debug mode
`DEBUG/PROCEDURE` *[low_lev,hi_lev] [switch]*
    run MIDAS procedures in debug mode
`DECONVOLVE/IMAGE` *frame psf result [no_iter] [cont_flag]*
    deconvolve image with point spread function
`DELETE/ACAT` *[catalog] [conf_flag] [range]*
    delete files with entry in ASCII file catalog
`DELETE/COLUMN` *table column_sel*
    delete table column(s)
`DELETE/COMMAND` *[comnd]*
    delete user defined command
`DELETE/CURSOR` *[disp]*

delete cursor window(s) on XWindow displays

`DELETE/DEFAULTS` *[comnd]*

delete special defaults for command

`DELETE/DESCRIPTOR` *frame descr_list [stop_flag]*

delete descriptors of frame

`DELETE/DISPLAY` *[disp]*

delete display window(s) on XWindow displays

`DELETE/FCAT` *[catalog] [conf_flag] [range]*

delete fit files with entry in catalog

`DELETE/FIT` *name [conf_flag]*

delete a fit file

`DELETE/GRAPHICS` *[grap]*

delete graphic window(s) on XWindow displays

`DELETE/ICAT` *[catalog] [conf_flag] [range]*

delete image frames with entry in catalog

`DELETE/IMAGE` *name [conf_flag]*

delete an image frame

`DELETE/KEYWORD` *key*

delete user defined keyword

`DELETE/LOGFILE`

delete current logfile

`DELETE/ROW` *table row_position number_of_rows*

delete one or several rows of a table

`DELETE/TABLE` *name [conf_flag]*

delete a table file

`DELETE/TCAT` *[catalog] [conf_flag] [range]*

delete table files with entry in catalog

`DELETE/TEMP`

delete temporary MIDAS files

`DELETE/ZOOM` *[disp]*

delete zoom window(s) on XWindow displays

`DISCONNECT/BACK_MIDAS` *unit [server_flag]*

disconnect from a background MIDAS

`DISPLAY/CHANNEL` *[chanl]*

display contents loaded in an Image Display channel

`DISPLAY/LUT` *[switch]*

en/disable display of current LUT

`DRAW/ANY` *[intens]*

draw manually or with input from file into the overlay channel

`DRAW/ARROW` *[in_spec] [coord_ref] [draw_opt] [intens] [nocurs] [key_flag]*

draw arrows in the overlay channel

`DRAW/CIRCLE` *[in_spec] [coord_ref] [draw_opt] [draw_specs] [nocurs] [key_flag] [arc_defs]*

draw circles in the overlay channel

`DRAW/CROSS` *[in_spec] [coord_ref] [draw_opt] [intens] [nocurs] [key_flag]*
draw crosses in the overlay channel
`DRAW/ELLIPS` *[in_spec] [coord_ref] [draw_opt] [draw_specs] [nocurs] [key_flag] [arc_defs]*
draw ellipses in the overlay channel
`DRAW/IMAGE` *frame [chanl] [scale] [center] [cuts] [over] [iaux] [fix]*
draw intensities of a line of an image into display channel
`DRAW/LINE` *[in_spec] [coord_ref] [draw_opt] [intens] [nocurs] [key_flag]*
draw straight line in the overlay channel
`DRAW/RECTANGLE` *[in_spec] [coord_ref] [draw_opt] [intens] [nocurs] [key_flag]*
draw rectangles in the overlay channel
`DRAW/SLIT` *[in_spec] [coord_ref] [draw_opt] [intens] [nocurs] [key_flag]*
draw IUE slits in the overlay channel

`ECHO/FULL` *[levla,levlb]*
show substitutions in MIDAS *procedure files*
`ECHO/OFF` *[levla,levlb]*
suppress display of input from MIDAS procedure files
`ECHO/ON` *[levla,levlb]*
display input from MIDAS procedure files
`EDIT/TABLE` *table [edit_option] [col] [row]*
interactive table editor
`EQUALIZE/HISTOGRAM` *frame descr itt_name*
perform histogram equalization
`EXECUTE/CATALOG` *com_string parm1 ... parm7*
execute a MIDAS *procedure or command for all entries in a catalog*
`EXECUTE/TABLE` *table command-string*
execute command on all rows of table
`EXTRACT/CTRACE` *[step] [frame] [plot_flag] [zw_option]*
extract a column from displayed image
`EXTRACT/CURSOR` *[subfr] [xpx,ypx] [loop_flag]*
extract a subframe via cursor
`EXTRACT/IMAGE` *out = in intval*
`EXTRACT/IMAGE` *out = in center_spec loff roff*
extract a subimage from an image frame
`EXTRACT/LINE` *out = in[..] step*
extract a 1-dim line from a 2-dim frame
`EXTRACT/REFERENCE_IMAGE` *in ref out thresh*
extract subimage according to reference image
`EXTRACT/ROTATED_IMAGE` *steps frame*
extract a rotated subimage from displayed image
`EXTRACT/RTRACE` *[step] [frame] [plot_flag] [zw_option]*
extract a row from displayed image

`EXTRACT/SLIT` *[in_option] [resframe] [slit_specs]*
extract a subimage defined by a fixed slit from image
`EXTRACT/TRACE` *[step] [frame] [plot_flag] [cut_option] [zw_option]*
extract a line from displayed image

`FFT/FINVERSE` *inr ini outr outi*
make inverse discrete Fourier transform
`FFT/FPOWER` *inr ini outr outi pow_spec*
make discrete Fourier transform and power spectrum
`FFT/FREQUENCY` *inr ini outr outi*
make discrete Fourier transform with frequency scaling
`FFT/IMAGE` *inr ini outr outi*
make discrete Fourier transform
`FFT/INVERSE` *inr ini outr outi*
make inverse discrete Fourier transform
`FFT/POWER` *inr ini outr outi pow_spec*
make discrete Fourier transform and power spectrum
`FILTER/COSMIC` *inframe outframe sky,gain,ron,[ns],[rc] [mask]*
remove cosmic ray events.
`FILTER/DIGITAL` *frame outframe [filter_specs] [subimage] [options]*
use digital filter on an image
`FILTER/GAUSS` *in out [radx,rady] [gauss_specs] [subima] [filtnam] [options]*
use Gaussian filter on an image
`FILTER/MAX` *frame outfram [xyradius] [subima] [options]*
apply maximum filter to an image
`FILTER/MEDIAN` *frame outfram [filt_specs] [flag] [subima] [options]*
smooth an image with median filter
`FILTER/MIN` *frame outfram [xyradius] [subima] [options]*
apply minimum filter to an image
`FILTER/SMOOTH` *frame outfram [filter_specs] [flag] [subima] [options]*
smooth an image by averaging
`FIND/MINMAX` *frame*
find min, max of frame and corresponding pixel numbers
`FIND/PIXEL` *frame lo,hi [inout_flag] [first_flag] [out_spec] [maxnpix] [area]*
find first/all pixel(s) with a value in/outside interval [low,high]
`FIT/FLAT_SKY` *outframe = inframe [in_specs] [order] [back_surface]*
`FIT/FLAT_SKY` *inframe [in_specs] [order] [back_surface]*
Approximate background of image by a surface
`FLIP/IMAGE` *frame [flag]*
flip an image around an axis

`GET/CURSOR` *[output] [option] [marker] [curs_specs] [zw_option]*
read cursor coords from display

`GET/GCURSOR` *[output_spec] [app_flag] [max]*
read and store cursor coordinates from the graphics display
`GET/IMAGE` *frame [input_source] [ITT_flag]*
read image from displayed image channel
`GET/ITT` *out_specs [chanl] [sect]*
read currently active ITT from image display
`GET/LUT` *out_specs [get_specs] [ITT] [format] [range]*
read currently active LUT from image display
`GROW/CUBE` *frame no_planes frame_list*
expand 2-dim/3-dim frame
`GROW/IMAGE` *out = in [start,step,no] [lincol_specs] [lincol_flag]*
expand single line into 2-dim image

`HELP` *[help_topic]*
display info about help_topic
`HELP/APPLIC` *[proc]*
display header information of application procedure
`HELP/CL` *[command]*
display help for commands only used in MIDAS procedures
`HELP/CONTRIB` *[proc]*
display header information of procedures in the Midas 'contrib' area
`HELP/KEYWORD` *key*
explain contents of given key
`HELP/QUALIF` *[qualif]*
display all commands with given qualifier
`HELP/SUBJECT` *[topic]*
display information related to given topic

`INDISK/ASCII` *in_file [out_file] [npix_string]*
read ASCII *file from disk + convert to Midas image*
`INDISK/FITS` *in_files [out_spec] [option] [check_flag]*
read FITS *files from disk*
`INFO/DESCR` *frame descr*
get type and size of descriptor
`INFO/IMAGE` *frame [option]*
get internal info of frame
`INFO/KEYWORD` *keyword*
get type, size, lockno and mod-time of keyword
`INFO/SETUP` *[setup]*
display all the information about a Setup
`INITIALIZE/DISPLAY` *[noLUT,maxLUTsz,minLUTsz] [ownLUT] [M_unit] [fonts] [colmode]*

initialize the image display

`INITIALIZE/SETUP` *[setup]*
initialize the variables of a Setup

`INSERT/IMAGE` *subframe modframe [startx,y,z]*
insert a subframe into another frame

`INTAPE/FITS` *file_specs file-id device [flags]*
read frames from magtape in FITS/IHAP format

`INTERPOLATE/II` *outima inima refima [s] [degree]*
interpolate Image to Image

`INTERPOLATE/IT` *outtab i,d inima [s] [degree]*
interpolate Image to Table

`INTERPOLATE/TI` *outima intab i,d refima [s] [degree]*
interpolate Image to Image

`INTERPOLATE/TT` *outtab i,d intab i,d [s] [degree]*
interpolate Table to Table

`ITF/IMAGE` *inframe table coli,colo scal outframe*
ITF correction

`JOIN/TABLE` *intab1 :X1,[:Y1] intab2 :X2,[:Y2] outtable [tolX,tolY]*
join table files

`LABEL/ANY` *filename*
write text labels stored in a file on the image display

`LABEL/DISPLAY` *labl [position] [mode] [option] [size] [key_flag]*
write a label on the image display

`LABEL/GRAPHIC` *label [x_pos,y_pos[,mm]] [angle] [size] [pos_ind]*
write a label_string on the graphics device

`LOAD/IMAGE` *frame_spec [chanl] [scale] [center] [cuts] [dirs] [fix]*
load image into display device

`LOAD/ITT` *in_specs [chanl]*
load intensity transfer table to Image Display

`LOAD/LUT` *in_specs [load_specs] [disp_flag] [format]*
load colour lookup table into Image Display

`LOAD/TABLE` *table x y [ident] [symbol] [size] [color] [conn_flag]*
load table into overlay channel of Image Display

`LOCK/KEYWORD` *key_list lockno*
lock keyword(s)

`LOG/OFF`
suppress logging

`LOG/ON`
enable logging

`LOG/TOF`

write top_of_form into logfile

`MAGNITUDE/CENTER` *[in_specs] [out_specs] [Fsiz,Nsiz,Bsiz] [out_opt] [center_params] [curs_specs] [zw_option]*
compute magnitude in center
`MAGNITUDE/CIRCLE` *[in_specs] [out_specs] [Fsiz,Nsiz,Bsiz] [out_opt] [center_params] [curs_specs] [zw_option]*
compute magnitude within circular aperture
`MAGNITUDE/RECTANGLE` *[in_specs] [out_specs] [pix_area] [out_opt] [center_params] [curs_specs] [zw_option]*
compute magnitude within square aperture
`MERGE/TABLE` *intable [intable ...] outtable*
merge table files
`MODIFY/AREA` *[source] [resfram] [degree] [constant] [drawflg]*
remove bad data from a circular pixel-area in an image
`MODIFY/COLUMN` *source_def res_frame [col_type] [column_coords]*
approximate values in a column
`MODIFY/CUTS` *[image] [cursor_spec]*
modify cut values of full frame or in cursor selected windows
`MODIFY/DISPLAY` *[target]*
toggle the state of a display window (mapped Window ¡-¿ Icon)
`MODIFY/GCURSOR` *frm_in frm_out y-coord xstart,xend no_curs,degree*
interactive modification of pixel values in a frame
`MODIFY/GRAPHICS` *[target]*
toggle the state of a graphics window (mapped Window ¡-¿ Icon)
`MODIFY/ITT` *[method] [value] [prflag]*
modify the currently active ITT
`MODIFY/LUT` *[method] [colour] [prflag]*
modify the currently active LUT
`MODIFY/PIXELS` *[source] [resfram] [arfacts] [xdeg,ydeg,niter] [drawflg] [noise] [ima_sec]*
approximate pixel-area in an image
`MODIFY/ROW` *source_def res_frame [row_type] [row_coords]*
approximate values in a row

`NAME/COLUMN` *table column [new-column] [unit] [format]*
redefines label/unit/format of a column

`OPEN/FILE` *filename flag file_control_key*
open an ASCII *file for reading or writing*
`OUTDISK/FITS` *in_files [out_spec] [option] [out_type]*
write Midas frames to FITS *files on disk*

`OUTDISK/SFITS` *in1[,in2[,...]] [out] [flags]*
convert set of MIDAS *files to a single FITS file on disk*
`OUTTAPE/FITS` *[catalog[,list]] device [flags] [density,block] [type]*
write to device in FITS *format*
`OVERPLOT/AXES` *[x_axis_spec] [y_axis_spec] [x_sc,y_sci[,x_off,y_off]] [x_lab] [y_lab]*
overplot a coordinate box with tickmarks and labels
`OVERPLOT/AXES` *[coord_str] [x_lab] [y_lab]*
overplot a coordinate box around a displayed frame
`OVERPLOT/COLUMN` *frame [x-coord] [y_start,y_end] [offset] [l_type]*
overplot a column of a frame on a graphic device
`OVERPLOT/COLUMN` *frame [x-coord] [y_start,y_end] [offset] [l_type]*
overplot a column of a frame on a graphic device
`OVERPLOT/CONTOUR` *frame [coord_str] [contours] [sm_par]*
overplot contour map of 2-dim. frame with smoothing option
`OVERPLOT/DESCRIPTOR` *frame [descr] [start,end] [offset]*
overplot the contents of a descriptor
`OVERPLOT/ERROR` *table [col1] [col2] col3 [direct] [bar]*
overplot table error column
`OVERPLOT/GRAY` *frame [coord_str] [gray_lev] [sm_par] [gray_ness] [options]*
overplot gray scale map of 2-dim. frame with smoothing option
`OVERPLOT/GRID` *grid*
overplot a grid on an existing coordinate box
`OVERPLOT/HISTOGRAM` *tab col [offset] [bin[,min[,max]]] [exc] [log] [opt]*
overplot histogram of a column in the table
`OVERPLOT/HISTOGRAM` *frame [offset] [log] [opt]*
overplot the histogram of an image
`OVERPLOT/KEYWORD` *[key_name] [start,end] [offset]*
overplot the contents of a keyword
`OVERPLOT/LINE` *[l_type] [x_sta,y_sta [x_end,y_end]]*
overplot a line on a graphic device
`OVERPLOT/ROW` *frame [y-coord] [x_start,x_end] [offset] [l_type]*
overplot a row (line) of a frame on a graphic device
`OVERPLOT/SYMBOL` *[s_type] [x_coord,y_coord] [s_size]*
overplot a symbol
`OVERPLOT/TABLE` *table [plane1] [plane2] [x_sc,y_sc[,x_off,y_off]] [symbols] [lines] [flag_dir]*
plot table data on selected plotting device
`OVERPLOT/VECTOR` *fram_a fram_b [coord_str] [scale_r] [pos_range] [sm_par] [head]*
overplot vector map from two 2-dim. images with smoothing option

`PLAYBACK/FILE` *name*
playback MIDAS *commands from an ASCII file*

`PLAYBACK/LOGFILE` *file*
    playback MIDAS *commands from a previous logfile*
`PLOT/AXES` *[x_axis_spec] [y_axis_spec] [x_sc,y_sc[,x_off,y_off]] [x_lab] [y_lab]*
    plot a coordinate box with large and small tickmarks and labels
`PLOT/AXES` *[coord_str] [x_lab] [y_lab]*
    plot a coordinate box around a displayed frame
`PLOT/COLUMN` *frame [x_coord] [y_sta,y_end] [x_sc,y_sc,x_off,y_off]*
    plot a column of an image on a plotting device
`PLOT/CONTOUR` *frame [coord_str] [x_sc,y_sc[,x_off,y_off]] [contours] [c_type] [sm_par]*
    plot contour map of 2-dim. image with smoothing option
`PLOT/DESCRIPTOR` *frame [descr] [start,end] [x_sc,y_sc[,x_off,y_off]]*
    plot a descriptor on plotting device
`PLOT/GRAY` *frame [coord_str] [x_sc,y_sc[,x_off,y_off]] [gray_lev] [sm_par] [gray_ness] [gray_opt]*
    plot gray scale map of 2-dim. image with smoothing option
`PLOT/HISTOGRAM` *tab col [x_sc,y_sc[,x_off,y_off]] [bin[,min[,max]]] [exc] [log] [opt]*
    plot histogram of a column in the table
`PLOT/HISTOGRAM` *frame [x_sc,y_sc[,x_off,y_off]] [exs] [log] [opt]*
    plot the histogram of an image
`PLOT/KEYWORD` *[key_name] [start,end] [x_sc,y_sc[,x_off,y_off]]*
    plot the contents of a keyword
`PLOT/PERSPECTIVE` *frame [coord_str] [alt,azi] [scal,offs] [sm_par] [xy_flag]*
    tree dim. representation of a 2-dim. frame, with smoothing option
`PLOT/ROW` *frame [y_coord] [x_sta,x_end] [x_sc,y_sc[,x_off,y_off]]*
    plot a row (line) of an image on a plotting device
`PLOT/TABLE` *table [plane1] [plane2] [x_sc,y_sc[,x_off,y_off]] [symbols] [lines] [flag_dir]*
    plot table data on selected plotting device
`PLOT/VECTOR` *frame_a frame_b [coord_str] [x_sc,y_sc[,x_off,y_off]] [scale_r] [range] [sm_par] [head]*
    plot vector map from two 2-dim. images with smoothing option
`PRINT/ACAT` *[cat_name] [lowno,hino]*
    print ASCII file catalog entries
`PRINT/DESCR` *frame [descr_list] [disp_flag]*
    print descriptor values
`PRINT/FCAT` *[cat_name] [lowno,hino]*
    print fit file catalog entries
`PRINT/HELP` *[help_topic]*
    print info about help_topic
`PRINT/HISTOGRAM` *table column [bin [min [max]]]*
    print statistics of a column
`PRINT/ICAT` *[cat_name] [lowno,hino]*

print image catalog entries

`PRINT/IMAGE` *frame_specs [pixel_specs] [hide_header_flag]*
print image data values

`PRINT/KEYWORD` *[key_list] [since]*
print contents of keywords

`PRINT/LOGFILE` *[page_specs]*
print contents of logfile

`PRINT/TABLE` *table [column ...] [elem1 [elem2]] [N] [width]*

`PRINT/TABLE` *table [elem1 [elem2]] [form] [N]*
print table values on the device/file specified via ASSIGN/PRINT

`PRINT/TCAT` *[cat_name] [lowno,hino]*
print table catalog entries

`PROJECTION/TABLE` *intable outable column_selection*
projection of one or more columns from a table

`READ/ACAT` *[cat_name] [lowno,hino]*
read ASCII *Catalog entries*

`READ/COMMANDS` *[proc]*
read commands from a procedure + store into command buffer

`READ/DESCR` *frame [descr_list] [disp_flag]*
display descriptor values

`READ/FCAT` *[cat_name] [lowno,hino]*
read fit file catalog entries

`READ/FILE` *file_id cbuf_key [maxrd]*
read an ASCII *file*

`READ/HISTOGRAM` *table column [bin [min [max]]]*
display statistics of table column

`READ/ICAT` *[cat_name] [lowno,hino]*
read Image Catalog entries

`READ/IMAGE` *frame_specs [pixel_specs] [hide_header_flag]*
display image data values

`READ/KEYWORD` *[key_list] [disp_flag] [since] [Midunit]*
display contents of keywords

`READ/SETUP` *setup*
read the contents of the variables related to a Setup

`READ/TABLE` *table [column_sel] [row_sel] [form]*
display table elements

`READ/TCAT` *[cat_name] [lowno,hino]*
read Table Catalog entries

`REBIN/II` *outima inima refima [func] [param] [intop]*
nonlinear rebin Image to Image

`REBIN/IT` *outtab i,d[,b] inima [func] [param] [intop]*
nonlinear rebin Image to Table

`REBIN/LINEAR` *in out [stepx,stepy] [offx,offy] [stx,sty] [fluxcons] [proc_mode]*
`REBIN/LINEAR` *in out [refframe] [fluxcons] [proc_mode]*
    rebin an image linearly
`REBIN/ROTATE` *in out [rot_specs] [ref_frame] [ref_flag]*
    rotate + rebin an image
`REBIN/SPLINE` *in out [stepx,stepy] [offx,offy] [startx,starty]*
`REBIN/SPLINE` *in out [refframe]*
    rebin an image using cubic splines
`REBIN/STATISTICS` *in out [xpix,ypix] [stat_val]*
    rebin image using statistical values as new pixel values
`REBIN/TI` *outima intab i,d[,b] refima [func] [param] [intop]*
    nonlinear rebin Table to Image
`REBIN/TT` *outtab i,d[,b] intab i,d[,b] [func] [param] [intop]*
    nonlinear rebin Table to Table
`REGRESSION/LINEAR` *table y x1,x2,...*
    linear regression on table columns
`REGRESSION/POLY` *table y x1[,x2] d1[,d2]*
    polynomial fit on table columns
`REGRESSION/ROBUST` *tab y x1[,x2,..,xn] [file] [out_col] [res_col]*
    Robust multi-variate regression by Least Median of Squares
`REGRESSION/TABLE` *table1 x1[,x2] table2 y1[,y2] degree tol [guess]*
    polynomial fit of variables in two tables (not yet implemented)
`RENAME/FIT` *old new [history] [overwrite]*
    rename a fit file
`RENAME/IMAGE` *old new [history] [overwrite]*
    rename an image frame
`RENAME/TABLE` *old new [history] [overwrite]*
    rename a table frame
`REPLACE/IMAGE` *in out [test/]low,hi=express1[,express2]*
    replace pixels according to intensity
`REPLACE/POLYGON` *in,intab out test/low,hi=value*
    replace pixels inside polygon
`REPORT/PROBLEM` *[errfile]*
    send error reports and comments to the person(s) in charge of MIDAS
`RESAMPLE/IMAGE` *in out stepping*
    resample an image
`RESET/DISPLAY`
    reset Xwindow display after Control C
`RESTORE/NAME` *[file_spec] [verbose] [history] [overwrite] [descr]*
    change file name according to descr. FILENAME
`RETRO/TAB` *table*
    retrofit 3-dim table to old 90NOV format
`ROTATE/1DIM` *in out nop_flag*
    rotate a 1-dim profile around its startpoint

`ROTATE/CLOCK` *in out [factor]*
  rotate an image by multiples of 90 degrees clockwise
`ROTATE/COUNTER_CLOCK` *in out [factor]*
  rotate an image by multiples of 90 degrees counter_clockwise
`RUN` *progr*
  execute a program inside the MIDAS environment

`SAVE/REGRESSION` *table name*
  save results of a regression
`SCROLL/CHANNEL` *[chanl] [scrolx,scroly]*
  scroll given ImageDisplay channel
`SEARCH/FCAT` *[cat_name] search_string [options]*
  search in fit file catalog for frame with matching descriptor IDENT
`SEARCH/ICAT` *[cat_name] search_string [options]*
  search in image catalog for frame with matching Ident field
`SEARCH/TCAT` *[cat_name] search_string [options]*
  search in table catalog for table with matching Ident field
`SELECT/TABLE` *table logical-expression*
  select table entries
`SET/ACAT` *[cat_name]*
  make given catalog the "active" ASCII file catalog
`SET/BACKGROUND` *[method] [echo] [sleep_time]*
  put Midas session into "background" mode
`SET/BUFFER` *[no_lines]*
  set up command buffer for MIDAS
`SET/CONTEXT` *cntxt*
  enable new context
`SET/CURSOR` *[curs_no] [curs_form] [curs_coords] [flag]*
  set cursor form and position
`SET/DISPLAY` *[colour_mode]*
  set up Image Display for RGB or pseudo colours
`SET/FCAT` *[cat_name]*
  make given catalog the "active" fit file catalog
`SET/FORMAT` *[format_specs]*
  set formats for replacement of Midas data
`SET/GCURSOR` *[curs_no] [curs_form]*
  set cursor form in graphics window
`SET/GRAPHICS` *option1[=value1] option2[=value2] ...*
  set plot characteristics
`SET/ICAT` *[cat_name]*
  make given catalog the "active" image frame catalog
`SET/ITT` *[chanl]*
  enable ITT *for Image display channel*

SET/LUT *[sect]*
    enable usage of colour lookup tables
SET/MIDAS_SYSTEM *option=value*
    set different modes and options for Midas
SET/OVERLAY
    enable graphics overlay
SET/REFCOLUMN *table column*
    define column as reference in table access
SET/TCAT *[cat_name]*
    make given catalog the "active" table file catalog
SHIFT/IMAGE *inframe outframe [x,yshift]*
    shift the pixels in an image
SHOW/ACAT *[cat_name] [display_flag]*
    show no. of entries in an ASCII file catalog
SHOW/BACK_MIDAS *unit [disp_option]*
    show info related to background MIDAS *sessions*
SHOW/CHANNEL *[chanl]*
    show info related to ImageDisplay channel
SHOW/CODE *command_string [option]*
    display the procedure which implements the command_string
SHOW/COMMAND *[comnd/qualif]*
    display MIDAS commands
SHOW/CONTEXT *[name]*
    show enabled contexts
SHOW/DEFAULTS
    display all special defaults
SHOW/DESCR *frame [descr_list] [flag]*
    show existing descriptors with name, type and size
SHOW/DISPLAY
    show current status of ImageDisplay + Graphics
SHOW/FCAT *[cat_name] [display_flag]*
    show no. of entries in a fit file catalog
SHOW/GRAPHICS *device_name*
    show the setup parameters for plotting
SHOW/ICAT *[cat_name] [display_flag]*
    show no. of entries in an image catalog
SHOW/KEYWORD *[keyword]*
    display contents of keyword data base
SHOW/TABLE *table*
    display table parameters
SHOW/TCAT *[cat_name] [display_flag]*
    show no. of entries in a table catalog
SHRINK/IMAGE *in_image out_image [option]*
    shrink 2 or 3-dim image to 1-dim image

`SORT/FCAT` *[cat_name]*
  sort a fit file catalog
`SORT/ICAT` *[cat_name]*
  sort an image catalog
`SORT/TABLE` *table keys*
  sort table according to (ascending) values
`SORT/TCAT` *[cat_name]*
  sort a table catalog
`STATISTICS/IMAGE` *[frame] [area] [bins] [lo,hi_exc] [option] [outtab] [plotflg] [format]*
  calculate statistics of a frame
`STATISTICS/TABLE` *table column*
  simple statistics on a table column
`STORE/FRAME` *key frame [indx] [exit_label]*
  store frame or entries of catalog into key
`SUBTRACT/ACAT` *[cat_name] frame_list*
  remove entries from an ASCII file catalog
`SUBTRACT/FCAT` *[cat_name] frame_list*
  remove entries from a fit file catalog
`SUBTRACT/ICAT` *[cat_name] frame_list*
  remove entries from an image catalog
`SUBTRACT/TCAT` *[cat_name] frame_list*
  remove entries from an table catalog
`SYNCHRONIZE/MIDAS`
  write keyfile and logfile to disk
`SYNCHRONIZE/TIME` *option*
  do time synchronization functions

`TRANSLATE/SHOW` *proc option*
  translate MIDAS *procedure and display resulting code*
`TRANSPOSE/CUBE` *inframe [outframe] [plane_spec]*
  rearrange the planes of a cube
`TRANSPOSE/IMAGE` *inima outima [diagonal]*
  transpose image
`TUTORIAL/EXTRACT`
  demonstrate some of the different EXTRACT commands
`TUTORIAL/FILTER`
  explain the usage of filters
`TUTORIAL/GRAPHICS` *option*
  explain the use of the graphics packages
`TUTORIAL/HELP`
  explain usage of the HELP command
`TUTORIAL/ITT` *[plotflag]*

explain the usage of ITT's

`TUTORIAL/LUT` *[plotflag]*
: show some standard LUT's and related MIDAS commands

`TUTORIAL/SPLIT`
: explain the usage of "split screen", i.e. multiple windows

`TUTORIAL/TABLE`
: explain usage of tables

`UNLOCK/KEYWORD` *key_list lockno*
: unlock keyword(s)

`$ comnd`
: execute a host system command

`VIEW/IMAGE` *[frame] [out_tab] [plot_option] [g,zhardcopy]*
: view an image with a "looking glass"

`WAIT/BACK_MIDAS` *[unit]*
: wait until command in background MIDAS *terminates*

`WAIT/SECS` *[no_of_secs]*
: suspend MIDAS *monitor for no_of_secs seconds*

`WRITE/COMMANDS` *[procnam] [par1] [par2] ... [par8]*
: save commands from command buffer + write into a procedure

`WRITE/DESCR` *frame descr data [flg]*
: store values into a descriptor

`WRITE/DHELP` *frame descr text*
: store help-text/comments for an existing descriptor

`WRITE/FILE` *file_id charbuf*
: write into an ASCII *file*

`WRITE/IMAGE` *frame [pix_specs] data [all_flg] [process_flg]*

`WRITE/IMAGE` *CURSOR,option data [process_flg]*
: update values of image frame

`WRITE/KEYWORD` *key data [flg]*
: write values into a keyword

`WRITE/OUT` *text_spec [section] [label]*
: display text on terminal

`WRITE/SETUP` *[setup]*
: modify the variables of a Setup

`WRITE/TABLE` *table column row_sel value*
: Store a value into a table

`XCORRELATE/IMAGE` *temp spec result shift* correlate 2 similar 1-dim frames over 2*(shift) bandwith

`ZOOM/CHANNEL` *[zoom_fact] [center]*
zoom image on image display
`ZOOM/OVERLAY` *[zoom_fact] [center]*
zoom image + overlay together

A.2 Application Commands

`ASSOCIATE/RANK` *table col1 col2 [action]*
rank-order correlation coefficient

`BIN/TABLE` *table col1 col2 [bin] [min] [max] [sigma]*
creates a table, bin.tbl with averages of col2 in bins of col1

`COMPARE/2SAM` *table col1 col2*
kolmogorov two-sample test
`COMPUTE/FIT` *image[,error] [= function[(refima)]]*
compute fitted image values
`COMPUTE/FIT` *table y[,error] [= function[(ind)]]*
compute fitted table values
`COMPUTE/FUNCTION` *image = function[(refima)]*
compute function values, result as image
`COMPUTE/FUNCTION` *table y = function[(ind)]*
compute function values, result in table column
`CREATE/FUNCTION` *fun1[,fun2...] [library_specs]*
define user functions for fitting
`CREATE/GUI` *[name]*
Creates graphical user interfaces
`CREATE/STAR` *in_frm in_tab out_frm [n_size] [frm_specs] [dmin,dmax] [radius]*
create the profile of a reference star by adding and averaging

`EDIT/FIT` *function*
interactive function editor

`FILTER/ADAPTIV` *frame outframe [maskframe] [type] [shape] size k noise*
adaptive filtering of an image
`FIT/IMAGE` *[nfeval[,prec[,metpar]]] [image[,wgt]] [funct]*

fit image values

`FIT/TABLE` *[nfeval[,prec[,metpar]]] table :dep,[:wgt] :ind [funct]*
fit table

`FTEST/VAR` *table col1 col2*
f-test for different variances

`GET/FIT` *table [image]*
create a table for fitting subimages

`IDENTIFY/CURSOR` *table ident x [y] [error]*
identify table entries from display

`IDENTIFY/GCURSOR` *table ident x [y] [error]*
identify table entries from graphic terminal

`INTEGRATE/APERTURE` *[in_specs] [out_tab] [radius]*
integrate the flux within an aperature

`INTEGRATE/LINE` *frame [y_coo] [x_sta,x_end] [n_cur,deg] [batch] [x-pos,range]*
integrate area in a (spectral) line

`INTEGRATE/STAR` *[in_specs] [out_tab] [parameters] [mode]*
computes flux, radius and background of stars previously centered

`KSTEST/1SAM` *table col distri coeffs*
kolmogorov one-sample test

`MODIFY/FIT` *table seq_no [name]*
modify fit parameters

`PRINT/FIT` *func_name*
print function parameters

`READ/FIT` *func_name*
display fitted function parameters

`REGISTER/SESSION` *session directory file table*
Register a session in the session manager

`REPLACE/FUNCTION` *fun1[,fun2...]*
replace user functions for fitting

`SAVE/FIT` *table seq_no [name]*
save results of a regression

`SELECT/FUNCTION` *name number[,...]*
    select function components
`SET/FIT` *par=value [par=value ...]*
    set parameters for the FITTING package
`SHOW/FIT`
    display parameters used in FITTING package
`SORT/COLUMN` *input output*
    column oriented sorting of the pixels of a frame.
`SORT/ROW` *input output*
    row oriented sorting of the pixels of a frame.
`STEST/MEAN` *table col1 col2*
    student t-test for different means

`TUTORIAL/ALIGN`
    explain the alignment of two images
`TUTORIAL/FIT`
    explain the modelling of table and image data by fitting non-linear functions.

A.3 Standard Reduction Commands

A.3.1 ccdred

`ALIGN/MOSAIC` *in_frm in_tab out_frm method,data [nxrsub,nyrsub] [xref,yref] [x_size,y_size]*
    Align the elements of the mosaiced frame
`BIAS/CCD` *[in_fram] [out_fram] [bs_fram]*
    Correct the input frame for the bias offset using a bias frame
`COMBINE/CCD` *exp [in_spec] [out_fram]*
    Combined a number of CCD *frames of the same exposure type*
`CREATE/MOSAIC` *in_cat out_frm out_tab nx_sub,ny_sub [not1,not2,...] [nocol,norow]*
    Mosaic a set of (infrared) ccd frames
`DARK/CCD` *[in_fram] [out_fram] [dk_fram]*
    Correct input frame for dark current offset using a dark current frame
`FIT/MOSAIC` *in_frm in_msk in_tab out_frm [match] [nxrsub,nyrsub] [xref,yref] [x_size,y_size]*
    Align and match the elements of the mosaiced frame
`FIXPIX/CCD` *[in_fram] [out_fram] [fix_table] [fix_meth]*
    Do a correction of bad pixels in the input frame
`FLAT/CCD` *[in_fram] [out_fram] [ff_fram]*
    Do a flat field correction of the input frame

FRCOR/CCD *[in_spec] [out_frm] [xboxmn,xboxmx] [yboxmn,yboxmx] [clip] [lowsig,higsig]*
    Make fringe correction frame(s0 from sky frames
FRINGE/CCD *[in_fram] [out_fram] [fr_fram] [fr_scale]*
    Do a fringe correction of the input frame
HELP/CCD *[keyword]*
    show the parameter setting of the current CCD *session*
ILLCOR/CCD *[in_spec] [out_frm] [xboxmn,xboxmx] [yboxmn,yboxmx] [clip] [lowsig,higsig]*
    Make flat field illumination correction frame(s)
ILLFLAT/CCD *[in_spec] [out_frm] [xboxmn,xboxmx] [yboxmn,yboxmx] [clip] [lowsig,higsig]*
    Apply correction to a flat field to remove illumination pattern
ILLUMINATION/CCD *[in_fram] [out_fram] [il_fram]*
    Do an illumination correction of the input frame
INIT/CCD *[name]*
    Initialize the CCD *package, optionally using the setting of a saved session*
LOAD/CCD *[intr]*
    Load instrument/detector specifications into the CCD *context*
MATCH/MOSAIC *in_frm in_tab out_frm method,data [match] [nxrsub,nyrsub] [xref,yref] [x_size,y_size]*
    Align and match the elements of the mosaiced frame
MKREDT/CCD *out_tab*
    Create CCD *empty table with columns for science and calibration frames*
OVERSCAN/CCD *[in_fram] [out_fram] [sc_area] [mode]*
    Correct the input frame for the bias offset in the overscan region
REDUCE/CCD *[in_spec] [out_frm]*
    Do the (partial) calibration of one or more frames
SAVE/CCD *name*
    save current CCD *session*
SET/CCD *keyw=value [...]*
    Define the values of parameters in the current CCD *session.*
SHIFT/MOSAIC *out_tab [curs_opt] [csx,csy] [clear_opt]*
    Get x and y shifts of the subraster in the mosaic frame
SHOW/CCD *[subject]*
    Show (part of) the setup of the CCD *package*
SKYCOR/CCD *[in_spec] [out_frm] [xboxmn,xboxmx] [yboxmn,yboxmx] [clip] [lowsig,higsig]*
    Make sky illumination correction frame(s)
SKYCOR/CCD *[in_spec] [out_frm] [xboxmn,xboxmx] [yboxmn,yboxmx] [clip] [lowsig,higsig]*
    Apply sky observation to flat field to remove illumination pattern

TRIM/CCD *[in_fram] [out_fram] [im_sec] [del_flg]*
: Extract the useful data from the ccd frame.

A.3.2 ccdtest

TEST1/CCD *in_cat [out_id] [meth] [option]*
: Combine bias frames stored in a catalogue and display it

TESTB2/CCD *in_frm [out_id] [row_ran] [col_ran]*
: Compute row and column average of a (averaged) bias frame

TESTB3/CCD *in_frm [out_id] [area] [size] [option]*
: Find the hot pixels in a (combined) bias frame

TESTB4/CCD *in_frm [out_id] [area] [size,fac]*
: Make a histogram of the pixel intensities and rebin the input frame

TESTB5/CCD *in_cat [out_id] [area] [size,fac]*
: Do the statistics of the bias frame is a catalogue.

TESTBA/CCD *in_cat [out_id] [meth] [row_ran] [col_ran] [area] [size,fac]*
: Do a series of tests of a catalogue of bias frames

TESTC/CCD *in_frm [rows] x_pix [colums] y_pix*
: Compute the horizontal and vertical charge transfer efficiency.

TESTD/CCD *in_cat [out_id] [dec_fac]*
: Do a test on a catalogue of dark current frames

TESTF1/CCD *in_cat [out_id] [meth] [area] [exp_ran] [option]*
: Combine the flat frame in the input catalogue and display

TESTF2/CCD *in_frm [out_id] [area] [thresh] [option]*
: Find the cold pixels in the combined low count flat.

TESTFA/CCD *in_cat [out_id] [meth] [area] [exp_ran] [thresh]*
: Do a series of tests on a catalogue of low count flat frames

TESTS/CCD *in_frm1 inf_frm2 [out_frm] n_exp [dec_fac]*
: Find the shutter error distribution

TESTT1/CCD *in_cat [out_id] [area] [option]*
: Display the linearity and transfer curves of pairs of flat frames.

TESTT2/CCD *in_tab [out_id] [select] [tim_int]*
: Fit the linearity cursves and determine the shutter offset

TESTT3/CCD *in_tab [out_id] [select]*
: Fit the transfer curve and determine the ADU *conv. factor and RON*

TESTTA/CCD *in_cat [out_id] [area] [tim_int] [select]*
: Do linearity and transfer tests on a catalogue of flat frames

A.3.3 do

ASSOCIATE/IMA *ost exptype rule_table outtable [flag] [nexp]*
: associates to scientific exposures a set of suitable calibration images

`CLASSIFY/IMAGE` *table descr outcol outchar*
  classify images according to one or several rules.
`CREATE/CRULE` *table rule*
  create an classification rule for a given Observation Summary Table
`CREATE/OST` *file_specs [file_pref] intable outtable flag*
  create an Observation Summary Table
`GROUP/ROW` *table incol outcol*
  group the rows of a table by the value of one of its column

A.3.4 echelle

`AVERAGE/TABLE` *frame table xy_col outcol [size]*
  Read pixels in a frame at positions defined by a table.
`BACKGROUND/ECHELLE` *in out [radx,rady,step] [degree] [smooth] [method]*
  estimate interorder scattered light of an echelle spectrum
`BACKGROUND/SMOOTH` *input output [radx,rady] [niter] [visu]*
  estimate interorder scattered light of an echelle spectrum
`CALIBRATE/ECHELLE` *[defmtd] [wlcmtd]*
  performs order definition and wavelength calibration
`CLEAN/ECHELLE`
  Clears contexts Echelle and Spec and removes process tables
`CONVERT/ECHELLE` *input output domain function param option*
  resample echelle orders
`DEFINE/ECHE` *[ordref] [width1,thres1,slope] [defmtd] [defpol]*
  define echelle order positions
`DEFINE/HOUGH` *[ordref] [nbord] [hwid] [hough_par] [thresh] [degx,degy] [hot_thres,step] [hough_setup]*
  define echelle order positions; automatic detection by Hough transform.
`DEFINE/SKY` *ima [nsky] [possky] [half_width]*
  defines limits of the sky windows
`DISPLAY/ECHELLE` *image [g_flag]*
  Optionally creates a display and graphic windows and scales an image to be displayed.
`ERROR/ECHELLE` *command keyword*
  Low-level error message generator for the echelle package
`EXTR/ECH` *input output [params] [method]*
  extract echelle orders
`EXTR/OPT` *input output slit,ord1,ord2 [ron,g,sigma] [table] [coeffs]*
  weigthed extraction of echelle orders
`EXTRACT/ORDER` *inp out sl,ang,off meth table coeff [ord1,ord2]*
  Extract echelle orders and produces a frame in space pixel-order
`EXTRACT/SKY` *in out [mode]*
  Extracts sky spectrum.

`FILTER/ECHELLE` *input output*
    filter echelle frame for cosmic ray hits and subtract background
`FLAT/ECHELLE` *[flat] [correct] [blaze]*
    subtract background from flat-field image and approximate blaze profile
`HELP/ECHELLE` *keyword [mode]*
    Provides short help on an echelle session keyword
`HOUGH/ECHELLE` *input [scan] [step,nbtr] [nbord] [flags] [hwid] [thres] [params]*
    perform Hough transform and orders detection on a flat-field frame
`IDENT/ECHEL` *[wlc] [lincat] [dc] [tol] [wlcloop] [wlcmtd] [guess,[shift]] [ccdbin]*
    perform wavelength calibration of echelle spectra
`INIT/ECHELLE` *[name]*
    initializes echelle parameters
`INITIAL/EMMI` *[ref] [grism]*
    Initializes the Echelle context for a given EMMI *configuration*
`KEYDEL/ECHELLE` *[table]*
    Deletes echelle session keywords
`LOAD/CALIBRATION`
    display wavelength calibration result
`LOAD/ECHELLE`
    display echelle orders (and optionally background) positions
`LOAD/IDENTIFICATION`
    display initial identifications.
`LOAD/SEARCH`
    Loads on display the position of the lines found by the SEARCH/ECHELLE *command.*
`MERGE/ECHELLE` *inframe outframe [params] [method]*
    merge echelle orders
`MERGE/OPTIMAL` *rebima weight out [delta]*
    Optimal weighted merging of echelle orders
`OFFSET/ECHELLE` *[image] [range] [cover] [ordtab] [mode]*
    Determines the offset along the slit between the order coefficients and a given echelle spectrum.
`OVERLAP/ECHELLE` *rebima order*
    Plots the overlap region between adjacent orders n and n+1
`PLOT/CALIBRATE` *[ord1,ord2]*
    plot dispersion relation in echelle reduction
`PLOT/ECHELLE` *frame [ord1,ord2] [printer]*
    plot extracted echelle orders.
`PLOT/IDENTIFICATION` *frame [ord1,ord2] [printer]*
    plot line identifications in echelle reduction
`PLOT/RESIDUAL` *[ord1,ord2]*
    plot dispersion residuals in echelle reduction

`PLOT/SPECTRUM` *in [start,end]*
    plots a rebinned spectrum in wavelength range
`PREPARE/BACKGROUND` *[step] [init] [back_tab] [order_tab] [descr]*
    low-level command; create table back.tbl
`PREPARE/WINDOW` *catalogue flat-bkg lhcuts*
    prepare echelle images for the command AVERAGE/WINDOW
`REBIN/ECHELLE` *input output sample*
    rebin echelle orders into wavelength
`REDUCE/ECHELLE` *input output [bkcor]*
    reduction of echelle spectra.
`REGRESSION/ECHELLE` *[defpol] [niter] [absres] [kappa]*
    fit 2-dim. polynomial to order positions (defpol limited to 5,5)
`REPEAT/ECHELLE` *[scalx,scaly] [response]*
    iterate on the response computation
`RESPONSE/ECHELLE` *[std] [fluxtab] [response]*
    compute instrument response
`RIPPLE/ECHELLE` *input output [params] [method] [option]*
    correct for the blaze function
`ROTATE/ECHELLE` *cat,ima root-name [mode] [flip_axis] [angle] [o_time]*
    rotate (and optionally flip) echelle images
`SAVE/ECHELLE` *name*
    saves current echelle session
`SAVINIT/ECHELLE` *ima,tab mode*
    saves/reads echelle session keywords as descriptor of an image/table
`SCAN/ECHELLE` *frame [scan-par]*
    update echelle keywords SCAN and IMSIZE.
`SEARCH/ECHELLE` *frame [width2,thres2]*
    search for emission lines
`SEARCH/ORDER` *[ordref] [w,t,s] [ordtab] [defmtd]*
    define echelle order positions
`SELECT/BACKGROUND` *[all]*
    interactive unselection of background reference positions
`SET/ECHELLE` *par=value [...]*
    set echelle keywords
`SHOW/ECHELLE`
    show echelle session
`SUBTRACT/BACKGROUND` *input bkg output [bkgmtd] [bkgvisu]*
    compute and subtract background from input frame.
`TUTORIAL/ECHELLE`
    demonstrates main commands of echelle package
`UPDATE/ECHELLE` *image*
    low-level command handling image geometry in world-coordinates
`UPDATE/ORDER` *image [offset]*
    Updates order definition coefficients and background table.

`VERIFY/ECHELLE` *file [type]*
check consistency of frame size against predefined values.

A.3.5 irac2

`ACUTS/IRAC2 [image] [load] [plot] [upper]`
Display an image with cuts mean-3*sig and mean+upper*sig

`CMASK/IRAC2 ffield clnffield lthrshold,hthrshold [dispflag]`
create a mask of bad pixels using a flatfield.

`DCOMB/IRAC2 [select] [seqname] [accsky] [align] output [trim] [tag]`
Sky subtract and combine dithered images.

`FFIELD/IRAC` *obj_frame ff_frame out_frame*
Flat Field an IRAC *frame*

`FOCUS/IRAC2 seqnum [focout] [create]`
Used to determine the best focus from a focus sequence.

`LAST` *[num]*
Gives very brief information on the most recent exposures

`MASK/IRAC2 inframe outframe`
replaces bad pixels by closest good pixel

`MKFLAT/IRAC` *lamp_on lamp_off flat_field*
Make a flat field

`OBSLIST/IRAC2 [start] [end]`
Lists a subsection of the IRAC2B *OST (Observation Summary Table)*

`OBSREP` *start end*
Print out a subsection of the IRAC2B *OST (Observation Summary Table)*

`QL/IRAC2 image1 image2 [outimage]`
Subtracts one IRAC2 image from another taking into account the detector integration times.

`RCOMB/IRAC2 select [align] output`
Combine frames created with the command RCOMB/IRAC2

`SEEING/IRAC2`
Determine the seeing, defined as the FWHM *of stellar images, of* IRAC2 images.

`SSUB/IRAC` *obj_frame sky_frame out_frame*
Sky Subtract an IRAC *frame*

A.3.6 irspec

`BADPIX/IRSPEC` *in out [l=load_opt.] [dir=clean_opt.] [debug=debug_opt.]*
Clean image of fixed pattern of bad pixels

`CALIBRATE/IRSPEC` *ima*

Apply on-line (mechanical) wavelength calibration.

`CALIBRATE/IRSPEC` *ima_ref mode=define*

Define and store in ima_ref precise calibration from sky/lamp lines

`CALIBRATE/IRSPEC` *ima ref=ima_ref*

Apply precise wavelength calibration to frame ima from parameters stored in ima_ref

`DEFINE/IRSPEC` *image table [mode] [threshold] [number] [load_option]*

Define fixed pattern of bad pixels and store it into a table.

`FLAT/IRSPEC` *in_flat in_dark out [l=load_opt.] [t=threshold] [v=vignetted_value]*

Create a normalized flat from an input flat frame.

`FLUX/IRSPEC` *in_ima response_ima out_ima [smooth=s1,s2] [shift=sh] [norm=normalize_option] [rect=rectify_option]*

Flux calibrate a spectrum (either 2D *or 1D) using a response frame* created using RESPONSE/IRSPEC

`MERGE/IRSPEC` *prefix_ima i1,i2[,i3] out_table [excl=#pixels_excluded] [corr=correct_option] [ref=#reference_image] [plot=plot_option] [format=i_format]*

Merge 1D *spectra into a table forcing connection of overlapping regions*

`RECTIFY/IRSPEC` *in out [l=load_opt.] [tilt=tilt_value] [ref=reference_row]*

Rectify tilted spectral lines.

`RESPONSE/IRSPEC` *in_ima flux_table out_response_ima [yrows=y1,y2,y3,y4] [obs=observation_mode] [norm=normalize_option] [rect=rectify_option]*

Create a response frame from a standard star 2D *spectrum and* a flux table.

`SKYSUB/IRSPEC` *ima_obj ima_sky out factor[,shift[,deltax,deltay]] [sky=sky_table] [force=force_sky_to_zero] [cuts=cuts_values] [debug=debug_option]*

Perform obj-sky correcting for variation and shift of sky lines.

`STANDARD/IRSPEC` *in_ascii_file out_table interp_method [degree=degree] [step=wavelength_step] [limits=wl1,wl2] [units=wavelength_units] [plot=plot_option]*

Create standard star flux table from a "flux ascii file".

`SUBTRACT/IRSPEC` *in_ima out_ima degree [exclude=area_to_exclude] [cont=continuum_image] [load=load_option]*

Fit and subtract, row by row, polynomial to a given image.

`TUTORIAL/CALIBR`

demonstration of wavelength calibration commands in IRSPEC

`TUTORIAL/IRSPEC`

General tutorial for the package IRSPEC

`TUTORIAL/SKYSUB`

Tutorial for sy substraction with IRSPEC *package*

A.3.7 long

`APPLY/DISPERSION` *in out [y] [coef]*
Apply the dispersion relation to a 1D *spectrum and generates a table*

`BATCH/LONG`
Prepare the Batch Reduction user interface

`CALIBRA/FLUX` *in out [resp]*
Correct an image for the instrumental response function

`CALIBRATE/LONG` *[tol] [deg] [mtd] [guess]*
Wavelength calibration of 1D *and long-slit spectra*

`CALIBRATE/TWICE`
Performs the wavelength calibration on a selected set of lines.

`CLEAN/LONG`
Clear context Long

`COMBINE/LONG` *catalog output [mtd]*
Average a catalog of images

`EDIT/FLUX` *[resp]*
Edit the instrumental response table

`ERASE/LONG`
Interactive rejection of dispersion relation nodes.

`ESTIMATE/DISPERSION` *wdisp wcent [ystart] [line] [cat]*
Estimate a linear approximation of the dispersion relation

`EXTINCTION/LONG` *in out [scale] [table] [col]*
correct spectra for interstellar or atmopsheric extinction

`EXTRACT/AVERAGE` *in out [obj] [sky] [mtd]*
extract a long-slit spectrum by averaging rows

`EXTRACT/LONG` *in out [sky] [obj] [order,niter] [ron,g,sigma]*
Optimal extraction of a long-slit spectrum

`GCOORD/LONG` *[number] [outtab]*
Get coordinates from the display window

`GRAPH/LONG` *[size] [position] [id]*
Creates a graphic window

`HELP/LONG` *[keyword]*
provides information about session keywords.

`IDENTIFY/LONG` *[wlc] [ystart] [lintab] [tol]*
Interactive calibration of lines in an arc spectrum

`INITIALIZE/LONG` *[session]*
Initialises parameters of context long

`INTEGRATE/LONG` *std [flux] [resp]*
Generates an intermediate response table from a standard star spectrum

`LINADD/LONG` *in w,bin [y] [mtd] [mode] [line]*
Adds entries to the table line.tbl

`LOAD/LONG` *image [scale_x,[scale_y]]*

`MAKE/DISPLAY`
    Creates a display window
`NORMALIZE/FLAT` *in out [bias] [deg] [fit] [visu]*
    Normalisation of flat-fields
`PLOT/CALIBRATE` *[mode]*
    Plot wavelength calibration identifications.
`PLOT/DELTA` *[mode]*
    Plot the fitted dispersion relation and allow interactive
    rejection of arc lines.
`PLOT/DISTORTION` *wave [delta] [mode]*
    Plot the fitted position of arc lines in wavelength/y-coordinate space.
`PLOT/FLUX` *[fluxtab]*
    Plot the flux table
`PLOT/IDENT` *[wlc] [line] [x] [id] [wave]*
    Plot interactive identifications
`PLOT/RESIDUAL` *[y] [table]*
    Plots residual after wavelength calibration
`PLOT/RESPONSE` *[resp]*
    Plots the response correction function
`PLOT/SEARCH` *[mode] [table]*
    Plot the results of SEARCH/LONG
`PLOT/SPECTRUM` *table*
    Plots a 1D *spectrum in table format, as supplied by APPLY/DISPERSION*
`PREPARE/LONG` *in [out] [limits]*
    Extracts sub-images from an image or a catalog.
`REBIN/LONG` *in out [start,end,step] [mtd] [table]*
    Rebin a long-slit spectrum using the row-by-row method
`RECTIFY/LONG` *in out [reference] [nrep] [deconvol_flag] [line]*
    rectify geometrically a distorted 2-D *spectrum*
`REDUCE/INIT` *partab*
    Initialises the batch reduction parameters
`REDUCE/LONG` *input*
    Batch reduction of long-slit spectra.
`REDUCE/SAVE` *partab*
    Saves the batch reduction parameters
`RESPONSE/FILTER` *std [flux] [resp]*
    Generate a response image by filtering based method.
`RESPONSE/LONG` *[plot] [fit] [deg] [smo] [table] [image] [visu]*
    Converts the response correction from table to image format.
`SAVE/LONG` *session*
    Saves session keywords
`SEARCH/LONG` *[in] [thres] [width] [yaver] [step] [mtd] [mode]*
    search for spectral features in a long-slit spectrum
`SELECT/LINE`

Select lines identified in all rows of an arc spectrum.

`SET/LONG` *key=value [...]*
: Assigns a value to long-slit session keywords

`SHOW/LONG` *[section]*
: Displays values of session keywords.

`SKYFIT/LONG` *input output [sky] [degree] [mode] [r,g,t] [radius]*
: fit polynomial to spatial flux distribution in windows of every column

`TUTORIAL/LONG`
: demonstrate commands of the package Long

`VERIFY/LONG` *file mode*
: Checks conformity of files in the long-slit context

`XIDENT/LONG` *[wlc] [ystart] [lintab] [tol]*
: Invoke the identification graphical user interface

A.3.8 optopus

`CREATE/OPTOPUS` *inp_file [out_tab] [fmt_file] [old_equinox]*
: create input table for HOLES/OPTOPUS command

`DRILL/OPTOPUS` *in_table [name]*
: write OPTOPUS *drill command file*

`HOLES/OPTOPUS` *[inp_tab] [out_tab] [HH,MM,SS.sss] [+/-DD,AM,AS.ss] [ac_flag] [p_flag] [old_eq,new_eq]*
: detetermine holes positions on Optopus plate.

`MODIFY/OPTOPUS` *[table]*
: plot positions of holes on plate and enable rejection of objects.

`PLOT/OPTOPUS` *[table] [label] [EW_flip_flag]*
: plot positions of holes on Optopus plate.

`PRECESS/OPTOPUS` *[inp_tab] [new_equinox]*
`p recess RA` *and DEC coordinates in table created by CREATE/OPTO.*

`REFRACTION/OPTOPUS` *[inp_tab] [out_tab] [year,month,day] [exp] [lambda1,lambda2] [start_st_slot,end_st_slot] [opt_st] [ast_flag]*
: correct for atmospheric refraction X *and Y coord. on Optopus plate.*

`RESTORE/OPTOPUS` *table*
: restore previously saved session parameters.

`SAVE/OPTOPUS` *table*
: save session parameters in descriptors

`SET/OPTOPUS` *option1[=value1] option2[=value2] ...*
: set Optopus context parameters

`SHOW/OPTOPUS`
: show session parameters.

`ZOOM/OPTOPUS` *[table] [zooming_factor]*
: blow up section of Optopus plate and enable rejection of objects.

A.3.9 pisco

`REDUCE/PISCO` *catalog table sky calibration [mode]*
perform complete reduction of polarimetric data

A.3.10 spec

`CENTER/HISTOGRAM` *image*
Median estimate and scale estimates of an image
`COMPUTE/PARAL` *ra dec st wave refw*
Computes parallactic angle and atmospheric differential refraction
`CONTINUUM/SPEC` *in out [radius/meth] [type] [smooth] [degree]*
Fitting of a spectrum continuum by smoothing splines
`CORRELATE/LINE` *table_1 table_2 [pixel] [cntr,tol,rg,st] [pos,ref,wgt] [ref_value] [outima]*
Cross-correlation between table columns.
`CUMULATE/HISTOGRAM` *in out*
Transforms a histogram image into the cumulated histogram
`DEBLEND/LINE` *infile [fitim] [fitpar] [method] [contin] [input] [intab]*
multiple component Gaussian fitting of spectral lines
`DISPERS/HOUGH` *[wdisp] [wcent] fr_specs [line] [cat] [mode] [range] [vflag]*
Determination of dispersion relations by HT
`EXTINCTION/SPECTRUM` *inframe outframe scale [table] [col]*
correct spectra for interstellar or atmopsheric extinction
`FILTER/RIPPLE` *frame outframe period [start,end]*
correct 1-dim. images for periodic ripple (Reticon)
`GRAPH/SPEC` *[size] [position] [id]*
Creates a long graphic window adpated for spectroscopy
`HST/SPEC` *fits_wave fits_flux table*
converts two HST *FITS files providing wavelength and flux information*
`MERGE/SPECTRUM` *spec1 spec2 out [interval] [mode] [var1] [var2]*
merge two 1D *spectra*
`NORMALIZE/SPECTRUM` *inframe outframe [mode] [table] [batch_flag]*
approximate continuum of 1-dim. spectra for later normalization
`OVERPLOT/IDENTIFICATION` *[table] [xpos] [ident] [ypos]*
overplot line identifications
`REFRACTION/LONG` *inim outim [mode]*
Differential atmospheric correction for slit spectra
`REGRESSION/ROBUST` *tab y x1[,x2,..,xn] [file] [out_col] [res_col]*
Robust multi-variate regression by Least Median of Squares
`ROTATE/SPEC` *cat [root] [meth] [flip] [angle] [mode]*
rotate (and optionally flip) a catalog of images
`SEARCH/LINE` *frame w,t[,nscan] [table] [meth] [type]*

search for spectral lines

`VERIFY/SPEC` *file dir keyw [type]*
low-level spec command checking the existence of calibration tables

A.4 Contributed Commands

A.4.1 astromet

`ASTROMETRY/COMPUTE` *mes option out trail*
Convert the coordinates from the measured xy ot RA,Dec vice versa

`ASTROMETRY/EDIT` *std plot*
Delete/undelete the astrometric standards

`ASTROMETRY/POS1`
Interactive procedure for the POS1 astrometry package

`ASTROMETRY/TRANS` *std mes center pla,cat schmidt,blink tol xterm,yterm std*
Compute the astrometric transformation parameters of a data

A.4.2 cloud

`COMPUTE/ABSORPTION` *inframe outframe [cm_table] [ap_table] [psframe]*
computes a synthetic 1dim. absorption spectrum

`COMPUTE/EMISSION` *outframe [em_table]*
computes a synthetic 1dim. emission spectrum

`CREATE/PSF` *[outframe] fwhm*
creates a 1-dim. image of a normalized gaussian

A.4.3 daophot

`ALLSTAR/DAOPHOT`
do simultaneous multiple-profile-fitting

`DAOMID/DAOPHOT` *table*
convert a DAOPHOT table into a MIDAS table

`DAOPHOT/DAOPHOT`
do precise photometry and astrometry in a 2-dim frame

`MIDDAO/DAOPHOT` *table*
convert a MIDAS table into a DAOPHOT table

A.4.4 esolv

`FROMOD/ESOLV` *[mode] [colour] [intable] [column]*
retrieve frames from optical disk

`MTABLV/ESOLV [table] col1 col2 l_frac col3`
find semidiameter of ellipses at given fraction of light
`STATPL/ESOLV table col1 select disp`
computes mean and sd of table file column given in p2
`TABFLV/ESOLV [table] ascii_file flag`
lists the contents of special table file
`TEXLV/ESOLV [table] tex_file`
prepare Tex file with selected parameters from ESOLV

A.4.5 geotest

`CREATE/ART_IMAGE` *frame frame dims [starts,steps] [func_type] [coefs]*
`CREATE/ART_IMAGE` *frame = ref_frame [func_type] [coefs]*
create artificial image
`CREATE/RAMP` *image [slope] [angle] [dimension]*
generate uniform sloping image, with mean flux per pixel of 100 units
`CREATE/SPC1 image [slope] [ampl] [period] [phase] [dim]`
generate sinusoidal, sloping 1-dimensional image
`CREATE/SPC2 image [period] [slope] [phase] [dimension]`
generate a discrete 1-dimensional image
`CREATE/SPC3 image psf_option centring table boxwidth-or-fwhm`
generate an artificial spectrum with lines
`CREATE/WAVE` *image [amplitude] [period] [dimension]*
generate 2-dimensional sinusoidal background image

A.4.6 invent

`ANALYSE/INVENTORY` *frame in_tab [out_tab] [ver_par] [deb_mode] [out_psf]*
verify the used table of objects and calculates the image parameters
`CLASSIFY/INVENT table`
classify the analysed objects into stars, galaxies and spurious objects
`SEARCH/INVENTORY` *frame table*
search objects in an image frame and store the parameters
`SET/INVENTORY` *par1 [par2]*
display and modify the values of the keywords used by Inventory
`SHOW/INVENTORY` *par1 [par2]*
display the values of the keywords used by the Inventory package

A.4.7 mva

`CLUSTER/TABLE` *intable outable [method]*
hierarchical clustering

`CMDS/TAB input_table output_table ncols._output_table`
    multidimensional scaling
`CORRES/TAB input output row/column_analysis ncolumn outable`
    correspondence analysis
`EDIST/TAB input_table output_table`
    standard distances
`KNN/TAB` *training_table no._of_gp.1_members test_table no._of_NNs*
    discriminant analysis
`LDA/TAB Input_table Output_table`
    Fisher's linear discriminant analysis.
`MDA/TAB input_table output_table eigenvectors`
    discriminant analysis
`MST/TABLE intable outtable grid_size`
    create minimal spanning tree for position table
`PARTITION/TABLE` *intable outable [no_of_class] [alg] [min. card] [s_value]*
    non-hierarchical clustering
`PCA/TAB in_tab out_tab option row/col_anal ncols_table eigenvectors`
    principal components analysis
`PLOT/TREE` *intable [col_ref]*
    plot output created by minimal spanning tree algorithm

A.4.8 pepsys

`CONVERT/PHOT`
`Helps you make a new table of observational data`
`MAKE/HORFORM`
`Make a blank FORM` *to fill in with horizon-obstruction data*
`MAKE/PHOTOMETER`
`generates or checks the instrumental table file for a photometer`
`MAKE/PLAN`
`generates a photometric observing plan`
`MAKE/STARFILE` *arglist*
`Helps you make a new file of program or standard stars`
`REDUCE/PHOT`
`Reduces tables of observational data`

A.4.9 romafot

`ADAPT/ROMAFOT` *int_tab [thres] [fac_int] [fac_sky] [fac_hol] [x_siz,y_siz]*
    derive trial values for fitting a new frame
`ADSTAR/ROMAFOT` *in_frame out_frame [reg_tab] [cat_tab] [x_dim,y_dim] [n_sub]*
    create an artificial image with subframes added at random positions

`ANALYSE/ROMAFOT` *frame [cat_tab] [int_tab] [sigma,sat]*
  INPUT MODE select all stars within selected subfields;
  OUTPUT MODE check at the results of the fit operation and select
`CBASE/ROMAFOT` *frame_1 frame_2 [out_tab1] [out_tab2]*
  create two tables for coordinate transformation
`CHECK/ROMAFOT` *cat_tab reg_tab err_mag*
  examine number of artificial stars recovered and check the accuracy
`CTRANS/ROMAFOT` *int_tab [tab_1] [tab_2] [pol_deg]*
  find transformation of coordinates and apply to an intermediate table
`DIAPHRAGM/ROMAFOT` *frame [regi_tab] [rego_tab] ap_rad*
  do aperture photometry with fixed diaphragm
`EXAMINE/ROMAFOT` *int_tab [hmin,hmax]*
  examine quality of fitted objects and flag badly fitted ones
`FCLEAN/ROMAFOT` *cat_tab inti_tab [into_tab]*
  selects windows in intermediate table present in catalogue table
`FIND/ROMAFOT` *frame [cat_tab]*
  select objects using the image display
`FIT/ROMAFOT` *frame [int_tab] [thres,sky] [sig,sat,tol,iter] [meth,[beta]] [fit_opt] [mean_opt]*
  determine characteristics of stellar images by non-linear fitting
`GROUP/ROMAFOT` *frame [area] [cat_tab] [int_tab] [thres] [wnd_max] [end_rad,sta_rad] [wnd_perc]*
  automatic grouping of objects
`MFIT/ROMAFOT` *frame [int_tab] [thres,sky] [sig,sat,tol,iter] [meth,[beta]] [fit_opt] [mean_opt] [mod_file]*
  determine characteristics of stellar images by non-linear fitting
`MODEL/ROMAFOT` *[mod_file]*
  compute (sub)pixel values for a model observation
`REGISTER/ROMAFOT` *int_tab reg_tab [wnd_opt] [obj_opt]*
  computes and store the absolute quantities in the registration table
`RESIDUAL/ROMAFOT` *in_frame out_frame diff_frame [reg_tab]*
  compute reconstructed image and difference with original image
`SEARCH/ROMAFOT 07-AUG-1989 RHW`
  do the actual search for objects above a certain threshold
`SELECT/ROMAFOT` *frame [int_tab] [wnd_size]*
  select objects and store the positions in intermediate table
`SKY/ROMAFOT` *frame [sky_tab] [area] [nrx,nry] [min,max]*
  determines intensity histogram and sky background in selected areas

A.4.10 surfphot

`COMPUTE/FCOEFF` *infram orient rin,rout,rstep outtab*
  compute fourier coefficients of azimuthal profiles in spiral galaxies

`COMPUTE/GRID` *angle*
  create image and table with distorted rect. and evenly spaced grid
`COMPUTE/SKY` *infram1 infram2 caltab method sky_factor*
  compute the sky background and restitute the frame
`FILTER/FILL` *inframe outframe rx,ry thresh*
  fill up low-flux (below threshold) pixels with nearby high-flux pixels
`FIND/PAIR` *intab1 intab2 outtab colums [errors] [coo_sys]*
  match (pair) two coordinates tables and produce an output table
`FIND/POSINC` *infram x_pos,y_pos rin,rout,rstep*
  find the position angle and inclination of a galaxy
`FIT/BACKGROUND` *outframe = inframe(s) [deg,it] [clp1,clpn] [skew] [outbck]*
`FIT/BACKGROUND` *outframe = inframe(s) [coef] [outbck]*
`FIT/BACKGROUND` *inframe(s) [deg,it] [clp1,clpn] [skew] [outbck]*
`FIT/BACKGROUND` *inframe(s) [coef] [outbck]*
  compute 2-dim. polynomial fit of the background
`FIT/ELL1 inframe outframe l_iso,h_iso x_cen,y_cen max_rad`
  fit an ellips with respect to predefined center
`FIT/ELL2 inframe pol_opt iso_tol iso_levels [center[ [radius] [sky_level]`
  fit an ellips with respect to predefined center
`FIT/ELL3 inframe outframe [step] [x,y] [low,high] [min,max] [opt]`
  fit ellipses to the isophotes of an object in a 2-dim. frame
`FIT/POSINC` *infram orient rin,rout,rstep region*
  fit the position angle and inclination to 2nd and 4th harmonic
`INTEGRATE/ELLIPS` *frame [ellips_par] [flag]*
  integrate pixel intensities within ellipse in 2-dim. image
`NORMALIZE/IMAGE` *infram outfram trunc_vals control_vals*
  normalize and truncate a frame
`REBIN/DECONVOLVE` *frame psf result zoom_x,zoom_y n_iter*
  rebin image linearly in space and simultaneously deconvolve it with psf
`RECTIFY/IMAGE` *in out table [nrep] [deconvol_flag]*
  rectify geometrically a distorted direct image
`SUBTRACT/SKY` *inframe outframe nx,ny*
  remove sky by subtraction of histogram-modeled substitute-sky

A.4.11 tsa

`AOV/TSA` *intab outima start step nsteps [order] [cover]*
  compute analysis of variance periodogramme
`BAND/TSA` *intab [maxobs] Evaluate frequency band for time series analysis*
`COVAR/TSA` *intab1 intab2 outtab start step nsteps scale*
  compute discrete covariance function for unevenly sampled data
`DELAY/TSA` *intab1 intab2 outtab start step nsteps [func,mode] [parm]*
  compute chi2-time lag function

`INTERPOLATE/TSA` *intab outtab func parm*
    Interpolate an unevenly sampled series using its covariance function
`NORMALIZE/TSA` *intab1 outtab column [mode]*
    Normalize mean and variance to 0 and 1
`ORT/TSA` *intab outima start step nsteps [order]*
    compute multiharmonic AOV *periodogram*
`POWER/TSA` *intab outima start step nsteps*
    Compute discrete power spectrum for uneven sampling by slow method
`SCARGLE/TSA` *intab outima start step nsteps*
    Compute Scargle periodogramme for unevenly spaced observations
`SET/TSA set global keywords for TSA` *context*
`SHOW/TSA show global keywords for TSA` *context*
`SINEFIT/TSA` *intab outtab freque order iter*
    fit sine (Fourier) series, subtract it from input and return residuals
`TABLE/TSA` *inascii [outtab] [type] [mxcol] convert ASCI table into MIDAS table*
`WIDTH/TSA` *inima [width] [centre] Evaluate line width and profile*

A.5 Procedure Control Commands

`BRANCH` *var comparisons labels*
    multi-way branching
`CROSSREF` *labl1 labl2 labl3 labl4 labl5 labl6 labl7 labl8*
    define cross reference labels for the 8 parameters
`DEFINE/LOCAL` *key_def data A lower_levels_flag*
    define the maximum no. of parameters for a procedure
`DEFINE/PARAMETER` *Pi default type/option prompt_str low_lim,hi_lim*
    define default, type and valid interval for parameter i
`DO` *loopvar = start end [step]*
    define a DO *loop (as in FORTRAN)*
`ENTRY` *proc*
    define begin of procedure in a file with different name than the procedure
`GOTO` *label*
    branch to command line containing label:
`IF` *par1 op par2 command_string*
    execute conditional statement
`INQUIRE/KEY` *key prompt_string*
    get terminal input in a MIDAS *procedure*
`LABL:`
    define a label, LABL *in this example*
`RETURN` *par1 par2 par3*
    return to calling procedure (or terminal) and optionally pass up to 3

parameters back

‘

A.6 Commands Grouped by Subject

In the following list, general MIDAS commands are given grouped in main application areas. Only the most common commands are listed to make the list easier to use. Commands used for special types of data reduction are given in appropriate chapters in the main part of this manual.

A.6.1 MIDAS System Control

@@	Execute a MIDAS procedure
BYE	Terminate the MIDAS session
CHANGE/DIRECTORY	Change the default (current) directory for MIDAS
CLEAR/CONTEXT	Clear current context level or all levels
COMPUTE/KEYWORD	Compute values of a keyword
CONNECT/BACK_MIDAS	Connect "command syntax" to another MIDAS
CREATE/COMMAND	Create a user command
CREATE/DEFAULTS	Create special defaults for MIDAS commands
DEBUG/PROCEDURE	Run MIDAS procedures in debug mode
DEBUG/MODULE	Run MIDAS modules in debug mode
DELETE/COMMAND	Delete a user defined command
DELETE/DEFAULTS	Delete special defaults for command
DELETE/IMAGE	Delete an image frame
DELETE/KEYWORD	Delete user defined keyword
DELETE/LOG	Delete log file
DISCONNECT/BACK_MIDAS	Disconnect from a background MIDAS
ECHO/FULL	Show substitutions in program files
ECHO/OFF	Suppress display of input from program files
ECHO/ON	Display input from program files
LOG/OFF	Disable logging
LOG/ON	Enable logging
PLAYBACK/LOG	Playback log file
READ/KEYWORD	Display contents of keywords
RENAME/IMAGE	Rename an image frame
RUN	Execute program inside MIDAS
SAVE/COMMANDS	Save commands from command window in a procedure
SET/CONTEXT	Set new context level
SET/FORMAT	Format for “number-to-string” conversion
SET/MIDAS_SYSTEM	Set different modes and options for MIDAS
WAIT/BACK_MIDAS	Wait until command in background MIDAS terminates

WAIT/SECS	Suspend MIDAS monitor for no_of_secs second
WRITE/COMMANDS	Store commands from a procedure into the command window
WRITE/KEYWORD	Store values into a keyword
WRITE/OUT	Write out text

A.6.2 Help and Information

HELP	Display help info for a command
HELP/...	Display info about various topics
INFO/...	Get information about frames, descriptors and specific setup
PRINT/HELP	Print help information
PRINT/LOG	Print log file
SHOW/COMMANDS	Display MIDAS commands
SHOW/DEFAULTS	Display all special defaults

A.6.3 Tape Input and Output

INDISK/...	Read data from disk in FITS or ASCII format
INTAPE/FITS	Read data from tape in FITS or IHAP format
OUTTAPE/FITS	Write data to tape in FITS format

A.6.4 Image Directory and Header

ADD/xCAT	Add one or more entries to a catalogue
COPY/DD	Copy descriptors from one file to another
CREATE/xCAT	Create a catalogue
DELETE/...	Delete a frame
DELETE/DESCRIPTOR	Delete a descriptor
INFO/DESCRIPTOR	Get type and size of descriptor
READ/DESCRIPTOR	Read descriptors
RENAME/...	Rename a frame
SORT/xCAT	Sort entries in a catalogue
SUBTRACT/xCAT	Remove an entry from a catalogue
WRITE/DESCRIPTOR	Write a descriptor
WRITE/DHELP	Write descriptor help

A.6.5 Image Display

BLINK/CHANNEL	Blink between channels
CLEAR/ALPHA	Clear the alpha-numerics memory
CLEAR/CHANNEL	Clear and initialize memory channel
CLEAR/DISPLAY	Reset monitor
CLEAR/LUT	Bypass LUT in screen segment on monitor
CLEAR/SPLIT	Disable split screen
CLEAR/ZOOM	Clear zoom

COPY/CHANNEL	Copy image memory channels
COPY/DISPLAY	Hard copy of image display
CREATE/CURSOR	Create cursor window
CREATE/ZOOM	Create zoom window
CREATE/DISPLAY	Create a display window (using Xwindow)
CUTS/IMAGE	Set display thresholds for image
DELETE/DISPLAY	Delete the display windows
DISPLAY/CHANNEL	Display image loaded into channel
DRAW/...	Draw rectangle and other figures in the overlay plane
EXTRACT/CURSOR	Extract a subframe from the frame currently displayed
EXTRACT/ROTATED	Extract a rotated subimage from displayed image
EXTRACT/TRACE	Extract interactively a line from an image
GET/CURSOR	Coordinates from display device by cursor
GET/IMAGE	Read currently loaded image from channel
INITIALIZE/DISPLAY	Initialize the image display
LABEL/DISPLAY	Write character string on display device
LOAD/CURSOR	Display cursor into display device (DeAnza only)
LOAD/IMAGE	Load image into display device
LOAD/ITT	Load an intensity transfer table
LOAD/LUT	Load a colour lookup table into display unit
LOAD/TABLE	Display table data on image display
MODIFY/LUT	Modify the currently active lookup table
SCROLL/CHANNEL	Scroll image on given channel
SET/CURSOR	Set cursor form and position
SET/DISPLAY	Define colour display control, size of screen etc. (DeAnza only)
SET/LUT	Enable use of colour lookup table
SET/SPLIT	Enable split screen (DeAnza only)
SHOW/CHANNEL	Show information related to channel
VIEW/IMAGE	Explore an image interactively
ZOOM/CHANNEL	Zoom image on display

A.6.6 Graphics Display

ASSIGN/GRAPHICS	Define plotter output device and replot
CLEAR/GRAPHICS	Clear graphic screen
COPY/GRAPHICS	Copy the plot file to the specific graphic device
CREATE/GRAPHICS	Create a graphic window (using Xwindow)
CUTS/IMAGE	Set plot thresholds (high and low) for image
DELETE/GRAPHICS	Delete the graphic windows
GET/GCURSOR	Coordinates from graphic device by cursor
LABEL/GRAPHICS	Plot text in an existing plot
OVERPLOT/ERROR	Overplot table error column

OVERPLOT/HISTOGRAM	Overplot histogram of table column or image
OVERPLOT/ROW	Overplot row/line of image data on previous plot
OVERPLOT/TABLE	Overplot table data on previous plot
PLOT/AXES	Plot a coordinate box with large and small tickmarks and labels
PLOT/CONTOUR	Contour plotting of an image
PLOT/DESCRIPTOR	Plot an entry in a descriptor
PLOT/HISTOGRAM	Plot a histogram of a table column or an image
PLOT/ROW	Plot row/line of an image
PLOT/PERSPECTIVE	Perspective plotting (3-dim.) of an image
PLOT/TABLE	Plot table data
SET/GRAPHICS	Set plot characteristics like scaling
SHOW/GRAPHICS	Show graphic characteristics

A.6.7 Image Coordinates

CENTER/...	Find center
GET/CURSOR	Coordinates from image display via cursor
GET/GCURSOR	Get coordinates from graphics device by cursor
READ/DESCRIPTOR	List reference coordinates
WRITE/DESCRIPTOR	Write reference coordinates

A.6.8 Coordinate Transformation of Images

ALIGN/IMAGE	Calculate linear transformation between 2 images
EXTRACT/IMAGE	Extract part of image
FLIP/IMAGE	Flip image in x and/or y
GROW/IMAGE	Repeat one scan line to make 2 dim images
INSERT/IMAGE	Insert a subimage into father image
REBIN/II	Logarithmic, exponential, $r^{1/4}$ frequency rebin
REBIN/LINEAR	Pixel rebinning of image
REBIN/ROTATE	Rotate an image any angle
REBIN/SPLINE	Rebin an image with cube splines
REBIN/WAVE	Rebin 1-D image to linear wavelength
RECTIFY/IMAGE	General geometric correction
ROTATE/CLOCK	Rotate clockwise 90 degrees
TRANSPOSE/CUBE	Rearrange planes of 3-dim data cube
TRANSPOSE/IMAGE	Transpose an image

A.6.9 Image Arithmetic

AVERAGE/AVERAGE	Compute simple average of all pixels in a subimage
AVERAGE/COLUMN	Compute average of image columns
AVERAGE/IMAGE	Calculate the average of images
AVERAGE/ROW	Compute average of image rows
AVERAGE/WINDOW	Compare images, then take the meaning

COMPUTE/COLUMN	Perform arithmetic expression on image column
COMPUTE/IMAGE	Compute arithmetic expression of images
COMPUTE/PIXEL	Perform arithmetic operations on images using pixel coordinates
COMPUTE/ROW	Compute arithmetic expression on image scan lines
COMPUTE/..PLANE	Do arithmatic on planes of a data cube

A.6.10 Filtering

CONVOLVE/IMAGE	Convolve image with given point spread function
CREATE/FILTER	Create filter image
DECONVOLVE/IMAGE	Deconvolve image with point spread function
FILTER/GAUSS	Use Gauss filter on image
FILTER/MAX	Apply maximum filter to an image
FILTER/MEDIAN	Median filter image
FILTER/MIN	Apply minimum filter to an image
FILTER/SMOOTH	Smooth an image
FFT/IMAGE	Compute discrete fourier transform of a complex input frame
FFT/INVERSE	Compute inverse discrete fourier transform of a complex input frame

A.6.11 Image Creation and Extraction

COPY/II	Copy image frames
CREATE/IMAGE	Create new image
CREATE/RANDOM	Create a new image from a random distribution
EXTRACT/CURSOR	Extract a subframe from the frame displayed
EXTRACT/IMAGE	Extract part of an image
EXTRACT/LINE	Extract a line from a frame
EXTRACT/ROTATED	Extract a rotated image
EXTRACT/SLIT	Extract subimage defined by fixed slit
EXTRACT/TRACE	Extract line from an image
INDISK/ASCII	Read ASCII file from disk
INDISK/FITS	Read FITS file from disk
INSERT/IMAGE	Insert a subimage into father image

A.6.12 Transformations on Pixel Values

FIT/FLAT_SKY	Correct an image for sky variations
ITF/IMAGE	Transform pixel values in an image
MODIFY/AREA	Remove bad pixel from circular area
MODIFY/CURSOR	Change pixel values in image by cursor
MODIFY/GCURSOR	Change pixel values in image by graphic cursor
MODIFY/PIXEL	Change pixel values in image

REPLACE/IMAGE	Modify pixel values in given intensity interval
REPLACE/POLYGON	Replace pixel values inside a polygon

A.6.13 Numerical Values of Image Pixels

FIND/MINMAX	Display (ands tore) max and min value
FIND/PIXEL	Find first pixel with a value inside or outside the interval
FIT/FLAT_SKY	Fit background image
INTEGRATE/APERTURE	Integrate flux inside aperture
INTEGRATE/LINE	Integrate pixel-values over area in image
MAGNITUDE/CIRCLE	Compute the magnitude of the specified object by integrating over the central area defined by a circular aperture
MAGNITUDE/RECTANGLE	Compute the magnitude of the specified object by integrating over the central area defined by a rectangular aperture
MODIFY/CURSOR	Change pixel values in image by cursor
MODIFY/GCURSOR	Change pixel values in image by graphic cursor
MODIFY/PIXEL	Change pixel values in image
PLOT/HISTOGRAM	Plot histogram of pixel values in image
PRINT/IMAGE	Print an image
READ/IMAGE	List pixel values into image
STATISTICS/IMAGE	Calculate statistics of an image
WRITE/IMAGE	Change pixel values in image (world coordinates)

A.6.14 Spectral Analysis

CALIBRATE/LINE	Calculate coefficients for wavelength calibration
CENTER/...	Compute center of line
CONVERT/TABLE	Make image from table values
EXTINCTION/SPECTRUM	Correct 1-D image for extinction
IDENTIFY/GCURSOR	Identify table entries from graphic display
IDENTIFY/LINE	Equate X positions to wavelengths
INTEGRATE/GCURSOR	Integrate line interactively
MODIFY/GCURSOR	Change data in line interactively
OVERPLOT/IDENT	Overplot line identifications
PLOT/IDENT	Plot line identifications
REBIN/...	Linear or non-linear image rebinning
RESPONSE/SPECTRUM	Make file for flux correction, response curve
SEARCH/LINE	Search calibration lines

A.6.15 Least Squares Fitting

COMPUTE/FIT	Compute fitted image or table

COMPUTE/FUNCTION	Compute function values of image or table
EDIT/FIT	Define function for fitting
FIT/IMAGE	Least squares fitting in image
FIT/TABLE	Least squares fitting in table
PRINT/FIT	Print fitted values
READ/FIT	Read fitted values
SELECT/FUNCTION	Select functions to be fitted
SET/FIT	Control execution of fitting
SHOW/FIT	Display control parameters

A.6.16 Table File Operations

BIN/TABLE	Create a table with averages of col2 in bins of col1
COMPUTE/HISTOGRAM	Compute histogram for a table column
COMPUTE/REGRESSION	Compute column from regresion coefficients
COMPUTE/TABLE	Compute arithmetic expression between columns
CONVERT/TABLE	Compute image from table data
COPY/TT	Copy keys from table to table file
CREATE/COLUMN	Create new column in a table file
CREATE/TABLE	Create a table file
DELETE/COLUMN	Delete column from an element in a table file
EDIT/TABLE	Change value of entry in table file
MERGE/TABLE	Merge two table files
NAME/COLUMN	Insert a label name for a column
PRINT/TABLE	Print table
READ/TABLE	List elements of a table file
REGRESSION/POLYNOMIAL	Compute regression between column in table file
SELECT/TABLE	Select a subtable
SHOW/TABLE	List table directory
SORT/TABLE	Order a table file
STATISTICS/TABLE	Computes low order statistics for a column

Appendix B

Detectors

B.1 CCD Detectors

The description of CCD reductions is divided into four parts. Section B.1.1 describes the ideas behind the relative calibration of 2–dimensional photometric CCD–images.

Section B.1.3 gives a description of the steps required to perform a relative calibration of the CCD and discusses the various MIDAS commands that are available to perform the required operations. Section B.1.9 provides some examples of typical reduction scenarios. Section B.1.10 is a summary of the relevant commands.

The procedures described in the manual will yield CCD–frames whose pixel values represent relative flux from one to the other. In order to proceed to calibrate the images in terms of magnitudes requires further steps not detailed here.

B.1.1 Introduction

The nominal output X_{ij} of a CCD–element to a quantum of light I_{ij} can be given as

$$X_{ij} = A_{ij} + B_{ij} \times I_{ij} + \text{non–linear terms} + \text{BIAS} \tag{B.1}$$

This equation does not account for charge transfer inefficiencies and other effects known to exist in CCDs.

The dark current and the cold columns contribute to the additive term A; the quantum– and transfer–efficiency enter into the multiplicative term $B \times I$. It is known that the response of the CCD is essentially linear so the non–linear terms are generally neglected. The Bias is normally added to the output electronically to avoid problems with digitising values near to zero.

The objective of the first step in reducing CCD–images is to determine the relative intensity I_{ij} of a science data frame. In order to do this, two more frames are required in addition to the science picture, namely:

- FLAT–frames to determine the term B_{ij}, and
- DARK–frames to describe the term A_{ij}.

FLAT–fields are made by illuminating the CCD with a uniformly emitting source. The Flat–field then describes the sensitivity over the CCD which is not uniform. For FLAT–field exposures and SCIENCE–frames we get from Equation (B.1)

$$\mathrm{FLAT\_FRM(i,j) = DARK(i,j) + B(i,j) \times ICONS + BIAS} \tag{B.2}$$

$$\mathrm{SCIE\_FRM(i,j) = DARK(i,j) + B(i,j) \times INT\_FRM(i,j) + BIAS} \tag{B.3}$$

where ICONS represents a flux from a uniform source, and BIAS is a constant signal which is added to the video signal of the CCD before being digitised.

The DARK–current is measured in the absence of any external input signal:

$$\mathrm{DARK\_FRM(i,j) = DARK(i,j) + BIAS} \tag{B.4}$$

Combining Eqs.(B.2), (B.3) and (B.4) we isolate:

$$\mathrm{INT\_FRM(i,j)} = \frac{\mathrm{SCIE\_FRM(i,j) - DARK\_FRM(i,j)}}{\mathrm{FLAT\_FRM(i,j) - DARK\_FRM(i,j)}} \times \mathrm{ICONS} \tag{B.5}$$

ICONS can be any number. If set to the average signal of the dark–corrected FLAT–frame or a subimage thereof:

$$\mathrm{ICONS} = \langle \mathrm{FLAT\_FRM - DARK\_FRM} \rangle \tag{B.6}$$

then the reduced intensity frame INT_FRM will have similar data values as the original SCIE_FRM.

B.1.2 Discussion

The mean absolute error of INT_FRM(i,j) yields with ICONS = 1:

$$(\Delta I)^2 = \left(\frac{\partial I}{\partial S}\right)^2 (\Delta S)^2 + \left(\frac{\partial I}{\partial D}\right)^2 (\Delta D)^2 + \left(\frac{\partial I}{\partial F}\right)^2 (\Delta F)^2 \tag{B.7}$$

(Only the first letter is used for abbreviations.)

Computing the partial derivatives we get

$$(\Delta I)^2 = \frac{(F-D)^2(\Delta S)^2 + (S-F)^2(\Delta D)^2 + (S-D)^2(\Delta F)^2}{(F-D)^4} \tag{B.8}$$

A small error in ΔI is obtained if ΔS, ΔD and ΔF are kept small. This is achieved by averaging Dark, Flat and Science frames. ΔI is further reduced if $S = F$, then Equation (B.8) simplifies to

$$(\Delta I)^2 = \frac{(\Delta S)^2 + (\Delta F)^2}{(F-D)^2} \tag{B.9}$$

This equation holds only at levels near the sky–background and is relevant for detection of low–brightness emission. In practice however it is difficult to get a similar exposure level for the FLAT_FRM and SCIE_FRM since the flats are usually measured inside the Dome. From this point of view it is desirable to measure the empty sky (adjacent to the object) just before or after the object observations.

B.1.3 Reduction Steps

The software now available is a package for doing relative calibrations of the pixels, for averaging frames, and for cleaning images. Cleaning in this context means removal of instrumental faults and other defects from the images. A list of the relevant commands is provided for reference in the section B.1.10 of this appendix.

The CCD–reduction involves the steps outlined below. Steps in paranthesis [...] are optional but see the following discussion to determine if they are optional in your particular case.

- [Remove irrelevant columns of all frames]
- [Bias correct all frames]
- compute an average DARK $\Longrightarrow$ DARK_AVG
- [remove defects from DARK_AVG]
- compute an average FLAT $\Longrightarrow$ FLAT_AVG
- dark subtraction $\Rightarrow$ FLAT_AVG = FLAT_AVG - DARK_AVG
- [remove defects from FLAT_AVG]
- [normalize the FLAT_AVG,(see Eq. (B.6)) $\Rightarrow$ FLAT_NRM]
- dark subtraction $\Rightarrow$ SCIE_FRM = SCIE_FRM - DARK_AVG
- [remove defects from SCIE_FRM]
- flatfield the scie-frame SCIE_FLAT = SCIE_FRM/FLAT_NRM
- compute the average of science frames $\Rightarrow$ SCIE_AVG

The sections below describe and discuss the various options which are available to perform the needed operations. These sections discuss bias corrections, techniques for computing averages, techniques for cleaning images and removing specific defects, and finally how to use the `COMPUTE` command for dark subtraction and flat fielding.

B.1.4 Removing Irrelevant Columns

With the ESO CCD systems currently in use, some columns or/and rows are added in electronically and contain no valid data. The command:

```
EXTRACT/CCD extr_coords [catalog]
```

will remove these columns from all frames in the referenced catalog. The default is the general catalog. This function will slightly speed up subsequent operations and will avoid possible problems with strange pixel values sometimes found in these columns.

B.1.5 Bias Corrections

A bias level is present in every CCD–image and arises from an electronic offset which is added to the signal from the CCD before being converted to digital values (the bias prevents a negative output signal). Under the present operating conditions the bias level has a value of about 200 data units.

A special purpose command has been generated to subtract the bias from a series of frames. Note, however, that this procedure is not necessary if you are not going to take a weighted average of frames. Only in this case, will the inclusion of a bias effect the weighting given to an individual frame.

To subtract a bias value for a series of frames, execute the command:

```
BIAS/CCD bias_value [catalog]
```

where `catalog` is the name of the catalog containing the series of frames. The default is the general catalog. To understand how to create and use catalogs, see the description of catalogs in Volume A, Chapter 3.

B.1.6 Averaging and Merging Frames

Three different techniques for averaging have been introduced. The first is a straight forward simple average without regard for statistical weights or possible non–statistical pixel values. The second average takes account of the fact that the images being averaged may have different statistical weights. The difference in statistical weights may arise, since one image may have been exposed through clouds or may have different noise characteristics for some reason or other. The third averaging technique not only takes account of possible different statistical weights and mean values, but also excludes from the resulting average those pixels which differ from the others in a non–statistical way.

The average functions mentioned above will normally produce a result frame whose world coordinates are the intersection of the father frames' world coordinates. It is possible to specify that the result be the union of the world coordinates of the father frames. In this case, the average functions will produce merged images.

Simple averaging — This function will take the given frames and produce a simple average of the pixel values. The inputs are either a list of frames names separated by commas, or a catalog reference. In the latter case, all frames in the catalog will be averaged to produce the result. The command is:

```
AVERAGE/IMAGE OUT = IN_1,IN_2,...,IN_n [M] [null]
AVERAGE/IMAGE OUT = catalog.CAT [M] [null]
```

where in the first case, the series of specific input frames will be averaged and in the second case all entries in the specified catalog will be averaged to produce the output frame. The optional parameter `M` determines whether or not the images will be merged in world coordinate space. The optional parameter `null` is used to specify the value that will be used to replace invalid pixels.

Weighted averaging — This function is very similar to the simple average function only each input frame must have a precalculated weighting factor. The average is calculated according to the following formula:

$$OUT = \frac{\sum_i IN_i \times W_i}{\sum_i W_i} \tag{B.10}$$

The function is called as:

```
AVERAGE/WEIGHT OUT = IN_1,IN_2,...,IN_n [M] [null]
AVERAGE/WEIGHT OUT = catalog.CAT [M] [null]
```

where the parameters have exactly the same meaning as in the simple average command. The weighting factor has to be calculated beforehand with the command `COMPUTE/WEIGHT` or has to be entered by hand in a real descriptor `WEIGHT`.

Averaging statistically consistent values — This command differs somewhat from the purely mathematical functions of the previous two commands in the sense that it tries to anticipate some of the astronomical content of the input data frames. The function requires that four auxiliary parameters (in descriptors) be defined for each input frame before it can process the frames. These parameters can be defined by the function `WRITE/DESCRIPTOR` and are:

- `LHCUTS(5),LHCUTS(6)` — low and high limits for valid pixels
- `O_TIME(7)` — exposure time
- `FLAT_BCK(1)` — mean background value.

The function calculates the following:

$$OUT = \frac{\sum_i (IN_i - B_i) \times W_i}{\sum_i W_i} \tag{B.11}$$

with the restriction that only those pixels satisfying the condition

$$|IN_i - B_i - \langle IN - B \rangle| < 2.0 \times \sigma(\mathrm{Background}) \tag{B.12}$$

are accepted into the average. The threshold for rejection of pixels is a selectable parameter in the command. The default is shown.

The function is referenced to as:

```
AVERAGE/WINDOW OUT = IN_1,IN_2,...,IN_n bgerr,snoise
AVERAGE/WINDOW OUT = catalog.CAT bgerr,snoise
```

Merging frames — Often in order to properly merge frames, it is necessary to adjust the world coordinates of the frames so that the individual objects in each frame have the same world coordinates in the various different frames. This then allows the averaging routines to proceed to combine the frames correctly. Notice that almost all the MIDAS routines operate in world coordinates.

B.1.7 Cleaning Images

This section describes three functions which have been created to cleanup defects in images. The first two functions for cleaning rows and columns are particularly useful for CCD–images. The third function is of a more general nature, but also quite useful for CCD–images.

Cleaning rows or columns — Since CCDs are discrete row and column devices, they often exhibit defects which effect an entire row or column. These can often be removed with the commands:

```
MODIFY/COLUMN INPUT OUTPUT [C or V] [columns]
MODIFY/ROW INPUT OUTPUT [C or V] [rows]
```

These commands require an input frame, and an output frame. The parameter `C` or `V` determines whether the routine computes and adds a constant to the indicated columns/rows of the input data (option `C`), or if the routine uses the adjacent pixel values to extrapolate data for the indicated columns/rows (option `V`). The rows or columns specification is not required if the input also specifies a table containing the rows or columns to be corrected. See the detailed command description for a complete description.

Cleaning areas and Removing objects — A general purpose MIDAS command has been created for cleaning areas of an image. This is particularly useful for removing the permanent defects which may exist in the CCD–frames and which cannot be removed by the `MODIFY/COLUMN` or `MODIFY/ROW` commands. This command uses fitted surfaces to replace pixel values in the specified areas. It would also be useful for removing stars and so forth from an image. The command is:

```
MODIFY/PIXELS INPUT OUTPUT [AREA] [DEGREE] [MARK]
```

See the detailed command description in Vol. A, Appendix A for a description of how to use this command and how to specify the various input parameters.

The input parameter can be specified as `CURSOR`, and then the currently displayed image will then be used to interactively clean a resulting image with the cleaning area specified by the cursor positions.

CCD specific cleaning — A visual inspection of CCD–frames reveals bad pixel areas which result from defective sensors. Often these elements produce an abnormally large amount of dark current and are so bright that charge spills over to neighbour elements causing streaking or hot spots. Since the CCD sensors are linked in the vertical direction, the surplus charge will tend to spread along the columns.

Many of these defects are permanent i.e. they appear on every CCD–frame at the same position. Thus the Dark–subtraction should (in principle) remove them. However, this is often not the case since the manifestations of these defects depend on exposure levels and temperature among other things. So after Dark–subtraction a hot spot residual may be present.

A second instrumental fault are the low–sensitivity columns. The origin of cold columns are probably defects in the horizontal shift register which collect and shift the charge after exposure. The DATA of COLD–COLUMNS is NOT LOST. Their pixel values are only lower by a constant value than the "normal", neighbouring columns.

The division of the SCIE– by the FLAT–frame doesn't always remove the cold columns. The sky levels of these frames usually do not match, thus the flatfielding gives different ratios for cold– and normal–columns. Even if the sky–background levels agree, cold lines lying over sources won't disappear after FLAT–field correction.

The MIDAS functions discussed above are designed to correct the effects of these cold columns, hot spots and other defects. In the case of cold columns, it is usually possible to recover the data with the `MODIFY/COLUMN` command using the `C` option. For the hot spots and other defects, the `MODIFY/PIXEL` command can be used to replace the bad areas with a best estimate surface. This technique will produce cosmetically clean images, but will not recover the data as the `MODIFY/COLUMN` does. To use the `MODIFY/PIXEL` command, the following procedure can be used once a CCD–image has been loaded on the Display monitor:

```
DRAW/RECTANGLE CURSOR table
```

where `table` is the name of table in which to store the start– and end–points of the rectangles. The resulting table is reusable and can be saved for future use. It may be useful to create a set of tables, since the size of the hot spots varies with temperature and exposure time. This table may then be used by the command:

```
MODIFY/PIXELS inputs output [par3] [par4]
```

to replace the pixels inside the given rectangles via a 2–dimensional surface interpolation. A $\kappa \times \sigma$–clipping is used to fit a surface to the underlying background which can be the sky or an extended source.

Selecting `inputs` = "inframe,intable" instead of `CURSOR` leads to the reading of the surface world coordinates x-st,x-end,y-st,y-end for 'inframe' from the MIDAS table 'intable.' The columns must be labelled `XSTART`, `XEND`, `YSTART`, and `YEND`.

B.1.8 Using the COMPUTE Command

The `COMPUTE/IMAGE` command can be used to perform the operation of dark subtraction and flat fielding. The `COMPUTE/IMAGE` command is a very general purpose command and is described in great detail in Vol. A, Appendix A. The functionality of `COMPUTE/IMAGE` for the current purpose is explained below.

To do a dark subtraction enter the following:

```
COMPUTE/IMAGE DENOM = FLAT_AVG-DARK_AVG
```

where `DENOM` is the result frame, `FLAT_AVG` is the average of flat fields, and `DARK_AVG` is the average of dark frames.

To flat–field a data frame, enter the following:

```
COMPUTE/IMAGE SCIE_FLAT = (SCIE_IN-DARK_AVG)/DENOM
```

where `SCIE_FLAT` is the flat fielded science frame, `SCIE_IN` is the input data frame, `DARK_AVG` is the average of dark frames, and `DENOM` is the dark subtracted flat field from above.

B.1.9 Examples and Hints

This section gives a few simple examples and some hints on the best way to do things.

The Simplest Case

This paragraph shows the very simplest possible example of how to do the relative calibration of a CCD–frame. It is assumed that frames are only taken in a single filter (R in this case). The frames available are the following:

- Darks

  DARK01 is a 20 minute frame taken before/after observing

  DARK02 *dito*

  DARK03 *dito*

  DARK04 *dito*

- Flats

  FLATR1 20 sec dome flat in R–filter

  FLATR2 *dito*

  FLATR3 *dito*

- Data

  N5126R 20 minute exposure of NGC 5126

  N7794R 20 minute exposure of NGC 7794

In order to process these frames, the following commands would be executed.

```
AVERage/IMAge DARKAV = DARK01,DARK02,DARK03,DARK04
AVERage/IMAge FLATAV = FLATR1,FLATR2,FLATR3
COMPute/IMAge DENOM = FLATAV-DARKAV
COMPute/IMAge N5126RF = (N5126R-DARKAV)/DENOM
COMPute/IMAge N7794RF = (N7794R-DARKAV)/DENOM
```

The first two commands compute the simple averages of the dark and flat field images. The third command computes the dark subtracted average flat field. The final two commands perform the flat fielding of the data frames. Note that there has been no correction for bad columns in either the flat field, the dark frames or the data frames. The next example shows how to do this.

A Case with some Bad Columns

In this case, we have the same frames available as in the previous example, but now we want to correct for bad columns. We first show how to do this interactively and then how to do it using a table.

```
AVERage/IMAge DARKAV = DARK01,DARK02,DARK03,DARK04
AVERage/IMAge FLATAV = FLATR1,FLATR2,FLATR3
$ COPY MID$DISK:[SYSMIDAS.PROC]BADCOL.TBL *.*
MODIfy/COLumn FLATAV,BADCOL FLATAV2 C
MODIfy/COLumn DARKAV,BADCOL DARKAV2 C
COMPute/IMAge DENOM = FLATAV2-DARKAV2
COMPute/IMAge N5126RF = (N5126R-DARKAV2)/DENOM
COMPute/IMAge N7794RF = (N7794R-DARKAV2)/DENOM
MODIfy/COLumn N5126RF,BADCOL CLR5126 C
MODIfy/COLumn N7794RF,BADCOL CLR7794 C
```

This example is like the previous one except that this time, the average dark and average flat fields are corrected for low sensitivity columns and the final data frames also. The table `BADCOL` contains the list of low sensitivity columns which are to be corrected.

B.1.10 CCD-Commands Summary

Table B.1 lists the available CCD reduction commands.

Command	**Function**
`AVERAGE par1-par3`	computes average of images
`AVERAGE/WEIGHT par1-par3`	computes average using normalized images
`AVERAGE/WIN par1-par6`	average of images using only consistent pixel values
`BIAS/CCD value catalog`	Subtracts a bias value
`COMPUTE/IMAGE`	General image arithmetic
`COMPUTE/WEIGHT`	Computes weighting factors for the average functions `AVERAGE/WINDOW` and `AVERAGE/WEIGHT`
`EXTRACT/CCD catalog`	Removes irrelevant columns
`MODIFY/COL par1-par3`	adjusts low–sensitivity CCD–columns
`MODIFY/ROW par1-par2`	adjusts low–sensitivity rows
`MODIFY/PIXEL par1-par5`	replaces areas with a surface
`MODIFY/SKY par1-par5`	computes a constant sky–background

Table B.1: CCD Reduction Commands

Appendix C

CES

The reduction of CES spectra is, especially for point sources, very straightforward, and the general instructions given in Chapter 6 for the standard reduction of spectra are fully adequate. The following merely adds a few instrument specific details and hints.

Cameras: Except where noted otherwise, the reduction of data obtained with the Short or the Long Camera is exactly the same.

Detectors: As far as the data reduction goes, the main difference between Reticon and CCD is the data format. Since the Reticon is a one-dimensional array, no attention has to (can) be paid to the extraction of the object spectra. The data acquisition system used with the Reticon permits multiple exposures to be combined into a pseudo two-dimensional spectrum. For operations on such data, all commands which work on image rows are useful, e.g., `COMPUTE/ROW, AVERAGE/ROW` or also `EXTRACT/IMAGE`.

Blemishes, etc.:

- In many echelle orders a prominent ghost appears which, with some bad luck, may sit right on top of the object spectrum. It cannot be 'flatfielded away' or otherwise corrected.
- The first 10–15% (in wavelength) of most spectra suffer from vignetting which divison by a flatfield may even enhance rather than remove. The most successful way of correcting it is with a flatfield standard star (see item flatfielding below).
- Reticon data suffer from a periodic ripple which flatfielding does not always remove adequately. For the 'normal' 4- or 8-pixel ripple, the command `FILTER/-RIPPLE` can be tried. However, occasionally, also very weird periods occur which can be identified in a Fourier transform (see the various `FFT/` commands). Spikes in the real part of the Fourier transform can be removed with the interactive command `MODIFY/GCURSOR` before the data is transformed back to the original pixel space. Alternatively, `FILTER/RIPPLE` can be used once the period is known. *Note* that these corrections have to be done *prior* to any rebinning (wavelength calibration in particular)!

- The dome flatfield lamp has an emission line at 670.7 nm, apparently due to lithium. Other deviations from a pure continuum are not known, but you may wish to watch for them (this advice applies also to the 'internal' FF lamp).
- Some of the strongest absorption lines appear not only in stellar but occasionally also in flatfield exposures!

Background determination: For Reticon data, separate observations are required. On CCD spectra, the various background components (bias, scattered light, ghosts, sky, etc.) can usually be estimated from the signal on either side of the object spectrum.

Unless you are sure that features in the background spectra are significant, only subtract the mean value as number or a strongly smoothed background spectrum in order not to add noise to your object spectra.

Flatfielding:

'Internal' FF lamp is perfect (apart from vignetting, see above) for the Reticon and usually fully adequate for CCD spectra over most of the wavelength range accessible in the blue pass of the CES. In the red, the phases of the fringes in flatfield and object spectra may be so different that division by such a 'flatfield' only makes things much worse.

Dome flats: Observers have reported that the position of fringes may depend slightly on telescope position.

Bright stars without disturbing spectral features and *if observed sufficiently close to the target in both position and time*, may be used for three purposes:

High spatial frequencies: For this application, the spectrum of the flatfield star must have been trailed so that its well exposed part (along the spatial axis) fully covers the relevant positions of object spectra. Division of the object spectrum by the standard spectrum (both assumed bias corrected) will also take care of low spatial frequencies and, perhaps, telluric features as described below.

Low spatial frequencies: Command `NORMALIZE/SPECTRUM` can be used to obtain an approximation to the continuum of the comparison star. Division of this curve into the extracted target spectra will be useful in correcting for vignetting problems and other residual curvatures (echelle ripple) of the flatfielded spectra.

Telluric lines can, with some luck, be removed by dividing the extracted object spectrum by a suitably normalised extracted flatfield star spectrum. Wherever possible, this should be done prior to wavelength calibration.

Wavelength calibration: If you have to be worried about artifacts introduced by the non-linear rebinning, try to be innovative and do not rebin your data at all! If this is not practical, the following details should be considered:

- For high precision, always flatfield your arc spectra.
- Almost the only relevant comparison source is a thorium lamp. Do not use the argon lines; the lamp contains argon only to start the gas discharge process.
- By far the best laboratory wavelengths are those by Palmer and Engleman (1983). Their list is availabe as MIDAS table `TH` in directory `MID_ARC`. Copy this table to your MID_WORK, use `SELECT/TAB` on column :WAVE and `COPY/TT` to reduce the size of the table to the range in wavelength of your spectra, then delete the first copy to recover disk space.
- The information given in the descriptor `O_COM` about the wavelength (i.e., 'CRVALX') of the central pixel (i.e., 'CRPIXX') and the mean channel width (i.e., 'CDELTX') usually is extremely reliable and therefore very useful in the interactive part of identifying the comparison lines.
- Select the threshold in the line searching step so that about 10-25 lines are detected. (As a rule of thumb, it is for normally exposed arc spectra both necessary and sufficient to use all lines which in column :INTENSITY of table `TH` are listed with a laboratory strength of about 3 units or more.) Only the `GAUSSIAN` option in command `SEARCH/LINE` will give useful line positions.
- Identify (`IDENTIFY/LINE`) about five lines interactively; they should be well distributed over the wavelength range you are interested in.
- Always start with a parabola for the approximation of the dispersion curve (`CALIBRATE/WAVE`), never user polynomials with degree > 3.
- For normally exposed arc spectra, the automatic identification should identify most of the lines which in table `TH` are listed (column :INTENSITY) with a laboratory strength of 3 units or more (but, of course, reject blends).
- With the Long Camera, the rms scatter of the computed wavelengths about a fitted second-order polynomial would usually be 1–2 10^{-4} nm (if not better). In spectra taken with the Short Camera, the corresponding value may be about two times higher.
- Rebin your spectra to a step in wavelength which is at least two times smaller than the detector pixel width.
- For the rebinning of such very high resolution spectra it is important that the descriptors `START` and `STEP` and the relevant variables of programs are of double precision (often applies also to the subsequent analysis of the calibrated spectra). If in doubt or in order to check possible problems, suppress the leading two digits from the laboratory wavelengths (e.g., `COMPUTE/TABLE`) and later re-introduce them in descriptor `START` of the calibrated spectra.

.

Flux calibration is not possible for CES spectra unless you managed to observe a standard star with flux data that is extremely well sampled in wavelength.

Appendix D

Echelle Reduction

This Appendix describes the reduction of Echelle spectra. It describes the steps required to produce one-dimensional spectra from two-dimensional images. Operating procedures are described in detail so that this manual can be used as a guide by users without MIDAS experience.

A tutorial example is available in the system, the user is recommended to run it (`SET/CONTEXT echelle`, then `TUTORIAL/ECHELLE`) while reading the relevant parts of this document. The algorithms related to the reduction method are described in Chapter 7.

D.1 Input Data

The information involved in a reduction session consists of user data and system tables. User data is a set of echelle images observed with the same instrument configuration, corresponding to a wavelength calibration image (`WLC`), a flat field image (`FLAT`) and astronomical objects. Optionally, this set will include the standard stars (`STD`) to be used in the absolute or relative flux calibration, and dark images (`DARK`).

D.2 Auxiliary Data

Auxiliary data include catalogues with comparison lines and tables with the fluxes of some standard stars.

The area `MID_ARC` contains the following calibration catalogues, thar.tbl, Thorium-Argon catalogue (for `CASPEC, ECHELLEC, EMMI`). hear.tbl, Helium-Argon catalogue (for `EFOSC`). as well as Th-Ar tables selected for blends at spectral resolutions 25000, 33000, 50000, 100000, respectively named thar25.tbl, thar50.tbl, thar50.tbl, thar100.tbl (these tables have been prepared by H.Hensberge, M.David and W.Verschueren from Antwerp University).

Standard stars for flux calibration are available in the area `MID_STANDARD`. Additional tables of southern spectrophotometric standards are available in /midas/calib/data/spec/ctio (Hamuy & al., 1992, PASP, **104** 533-552 and Hamuy & al., 1994, PASP **106** 566-589) and midas/calib/data/datatrans/hststd (HST Standards).

These tables are distributed on request in complement to the Midas releases. These tables are also available on anonymous ftp at the host **ftphost.hq.eso.org** (IP number 134.171.40.2). The files to be retrieved are located in the directory /midaspub/calib and are named README.calib and calib.tar.Z. Command `SHOW/TABLE` can be used to visualize the column name and physical units of the tables. Demonstration data required to execute the tutorial procedure `TUTORIAL/ECHELLE` are also located on this ftp server in the directory /midaspub/demo as echelle.tar.Z. FTP access is also provided on the World Wide Web URL:

`http://http.hq.eso.org/midas-info/midas.html`

The calibration directory contains other information such as characteristic curves for ESO filters and CCD detectors, which can be visualized with the Graphical User Interface XFilter (command `CREATE/GUI FILTER`).

D.3 Starting the MIDAS Session

You should first get an account and a disk area on the computers you want to work on (ask the system manager or see Vol A, Appendix D).

After login into the system, you can start the MIDAS environment by typing the **inmidas** command. On a workstation, you may provide a 2 characters strings for the identification of the terminal: in most of the cases, you need to visualise images or graphics, so give two digits; otherwise give a two letters strings (default is 00).

The reduction of echelle spectra requires a set of dedicated commands which are defined under the context `ECHELLE`. Therefore each reduction session starts with the command

```
SET/CONTEXT echelle
```

to activate the echelle specific commands and to set all parameters to default values. Frequent users can integrate this command as well as display and graphic window creation commands in the **login.prg** file located in the working directory or in the `MID_WORK` directory.

D.4 Reading the Data

Your tapes must be in FITS or IHAP format. Mount the tape on a tape drive (e.g. `MTA0`, see Appendix D in Vol. A) on the computer you are using and, then, start the data input with the command

```
INTAPE/format 1-nn A MTAx
```

where `format` defines the input format as either FITS or IHAP, `nn` number of files that you want to read and `x` is either 0 or 1 to select the tape driver. Images files will be loaded into the working area with names `A0001`, `A0002` and so on as specified above.

D.5 Display and graphic windows

In the early reduction stages it is recommended to create a display with the same number of pixels as the CCD frame, so that one can easily check for defects.

The following example is only useful on workstations:

```
CREATE/DISPLAY 0 520,337
CREATE/GRAPH
LOAD/LUT heat
```

The first command will create a display with identification 0 and axes of 520 and 337 pixels. The second command creates a graphic window with the default setting which can be used in most cases. The last command loads a look-up table, i.e. an intensity-to-color conversion table. Other graphic windows, e.g. for the final spectra, can be specified with:

```
CREATE/GRAPH id x0,y0,xsize,ysize
```

Then images are displayed by:

```
LOAD/IMAGE ech0011 cuts=500,2000
```

which loads the image ech0011 in the active channel, using cuts values 500 and 2000 to define the range of represented intensities. If the image is loaded later, the last settings will be used as default values (they are stored in descriptors LHCUTS and DISPLAY_DATA).

Perhaps the cuts are not adequate and as a consequence desired details do not appear. To measure the intensity of a few pixels, use command:

```
GET/CURSOR
```

On a workstation: with the mouse, press leftmost button to get the values, and the center one to exit.

On a DeAnza: switch a cursor on, and use ENTER button to get the values. To exit, set both cursors off and press ENTER. Make sure that RATE and TRACK switches are off.

D.6 On-line help

Most Unix platforms provide the Motif libraries which allow to compile the Midas Graphical User Interfaces, in particular the on-line help, activated by the command:

```
CREATE/GUI help
```

which provides updated on-line help files corresponding to the Volume C of the documentation.

D.7 A few useful commands

It will be useful to have a list of the images stored on the disk. The command:

```
CREATE/ICAT mycat *.bdf
```

creates a catalog stored on the disk as the file mycat.cat which contains the list of all images present on the disk. It is better to enable this catalog for automatic use by all other commands using:

```
SET/ICAT mycat
```

which allows in any command to refer to a catalog image by the simplified syntax #n where n is the entry number of the image in the catalog.

The following commands are also useful:

```
READ/ICAT mycat
```

lists the contents of an image catalog.

```
PRINT/ICAT mycat
```

prints the contents of an image catalog on the default printer. See command ASSIGN/PRINT to change the default.

```
COMPUTE/IMAGE out = expression
```

performs arithmetic on images, for example:

```
COMPUTE/IMAGE out = img1 - img2
```

computes the difference of images img1.bdf and img2.bdf and stores the result in out.bdf.

```
STATIST/IMA frame
```

performs statistic on an image. It is possible to select a sub-window in interactive mode or direct mode.

```
STATIST/IMA frame CURSOR
STATIST/IMA frame area_spec
```

The following commands are related to graphic facilities:

```
PLOT/IMAGE, OVERPLOT/IMAGE
PLOT/TABLE, OVERPLOT/TABLE
GET/GCURSOR
SEND/PLOT, ASSIGN/PLOT
```

It is also useful to handle tables, using commands:

```
READ/TABLE table [column]
SELECT/TABLE table conditional_expression
STATIST/TABLE table column
COMPUTE/TABLE table out_column = expression
```

D.8 Preprocessing

For the remainder of this Appendix it is assumed that a complete set of input frames is available.

D.8.1 Bias correction

The Bias is a constant offset (for each pixel) introduced by the readout electronics of the CCD. Its value is usually between 150 and 250 ADU (for CCD's currently in use at ESO).

As a first step, the short dark and pre-flashed dark exposures have to be averaged to produce the `BIAS` and pre-flashed `BIAS` frame. A method which performs the averaging and which at the same time discards bad pixels due to e.g. cosmic events is `AVERAGE/WINDOW` Before actually executing this command, prepare a catalogue containing all input files to be averaged via:

```
CREATE/ICAT catname dirspec
```

Choose appropriate values for the descriptors `FLAT_BKG` and `LHCUTS` and run the command

```
PREPARE/WINDOW inputspec flatbkg lhcuts
```

where `inputspec` is either a single file name or the catalogue with the list of input files, to be specified as catname.cat. This program writes the descriptors `FLAT_BKG` and `LHCUTS/R/5/2` into each input frame and will also read the descriptor `O_TIME` to get the exposure time, prompting for it if the descriptor is not present. After that the various files can be averaged via:

```
AVERAGE/WINDOW out = catname.cat bgerr,snoise
```

D.8.2 Dark-current and particle hits correction

In a similar manner, long dark exposures of the same exposure time can be averaged to produce the `DARK` frames. Please note that the output of `AVERAGE/WINDOW` is in units of the exposure time which is usually in seconds. Use the command `COMPUTE/IMAGE` to multiply the images by the exposure time and to add back the value for `FLAT_BKG`.

Long dark exposures, and object and standard star frames contain a huge amount of cosmic events. If multiple exposures exist the above mentioned method can be used to average the frames and to remove the spikes. If not one can obtain good results by applying a median filtering algorithm to the frames (W.K. Pratt, Digital Image Processing, John Wiley & Sons, 1978). The corresponding MIDAS command is:

```
FILTER/MEDIAN input output 1,1,threshold NA
```

with the threshold depending on the r.m.s. noise in the exposure. The noise in the (pre-flashed) BIAS and DARK frames can further be decreased by smoothing the images via e.g.

```
FILTER/SMOOTH input output 10,10
```

After having done all these steps the final (pre-flashed) BIAS and DARK frames can be subtracted using COMPUTE/IMAGE from the FLAT, OBJ and STD frames. Often, BIAS and DARK contain no large - scale structure. Then, it is preferable to substract a constant, thus avoiding extra noise addition.

It is also possible to use several spectra of the same target obtained under the same instrumental configuration. Offsets of the target resulting from the positioning of the target on the entrance slit of the spectrograph and variations of exposure time must be taken into account. The commands AVERAGE/IMAGE and AVERAGE/WEIGHT offer number of options to compare the images and filter particle hits.

D.8.3 Standard orientation

The frames, as they appear on the monitor with the command LOAD/IMAGE, are in general not oriented properly. All reduction programs assume that on the monitor the wavelength increases from left to right and that the spectral orders increase from top to bottom. This means that the frames have to be rotated in such a way that:

- orders are more or less horizontal, but slightly climbing from left to right.
- orders separation are narrower at the bottom than at the top.

The images have to be rotated, depending on the way the CCD chip has been mounted. To do this, prepare again a catalogue containing all files which have to be rotated using CREATE/ICAT. Next run the program

```
ROTATE/ECHELLE inputspec outframe mode
```

Note

The rotation is usually required for CASPEC, EFOSC, ECHELLEC, *and* EMMI *spectra. The command* ROTATE/ECHELLE *sets descriptor values* START and STEP *to 1.,1. for all images and updates the session keyword* CCDBIN *to the original steps of the images. These operations must be performed by the user if another command than* ROTATE/ECHELLE *is applied.*

As in PREPARE/WINDOW, inputspec is either a single input frame or a catalogue; outframe is either a single output frame or a two character identifier which will replace the first two characters of the name of each input frame; mode decides if the images are just rotated over -90 degrees (mode ROTATE, data obtained before April 1984) or rotated over 90 degrees and flipped in the x-direction (mode FLIP) for frames obtained later. This command will also read the exposure time from the descriptor O_TIME, prompting for it if the descriptor is not present.

D.9 Session Parameters

Input data observed with the same configuration of the instrument and the parameters needed for the reduction define a session.

Session parameters are listed by the command `SHOW/ECHELLE`. It is recommended to use this command to control the actual status before executing any further reduction step. The command `SHOW/ECHELLE` displays only those parameters which must be set according to the options and methods chosen for the reduction. This enables to know exactly which parameters are required and available to tune the reduction.

Current parameters are saved with the command `SAVE/ECHELLE name`, where `name` is the session name. This command will be used whenever you want to interrupt a session which then can be restarted at any other time.

Current parameter values are set to the default value with the command `INIT/ECHELLE`, or set to the values of a previously saved session with `INIT/ECHELLE name`. `INIT/ECHELLE` is required each time you start the reduction of a new set-up or instrument.

The command `HELP/ECHELLE` provides a short on-line information on the echelle keywords. The keyword name can be truncated to a significant root name. The information consists of a description, the default and actual value. Four different modes are available:

- "HELP/ECH" without parameter, displays a general help message about the echelle package.
- "HELP/ECH <keyword>" gives a short description of the echelle keyword <keyword>.
- "HELP/ECH method" provides a list of method keywords
- "HELP/ECH option" provides a list of option keywords

Relevant session parameters can be defined for each command in the usual way:

```
command/qualifier parameters
```

or can be defined in explicit form as

```
SET/ECHELLE param=value [param=value ...]
```

where `param` is the parameter name and `value` is the assigned value. The assigned values will be maintained until you save them for later reference. Current parameter values are re–assigned either by `INIT/ECHELLE` or by another `SET/ECHELLE` command.

The direct assignment is possible as well, as in the following examples:

```
WIDTH1 = 10
WLCLOOP(2) = 0.1
LINCAT = ''thar.tbl''
```

D.10 The Reduction Session

The reduction session consists of the following steps:

- Order definition: defines the orders location, usually from a Flat-Field
- Wavelength calibration
- Flat-Field correction
- Correction for the response curve
- Reduction of science spectra

Flat-field correction is optional as well as response curve correction, but instrumental effects show up as large variations on varying scales. There are two possibilities: either you have a standard star (STD) or you don't. If you did not take a STD exposure, because you were not interested in an absolute or relative flux calibration, you will now have serious problems in producing an acceptable spectrum. There are correction methods for the instrumental response with no standard star but they do not always provide a satisfactory solution and are time consuming.

A summary of the different operations performed by commands `RESPONSE/ECHELLE` and `REDUCE/ECHELLE` is provided in D.13.

D.10.1 Reduction using Standard Stars

In general it is necessary at this stage to assign values to the parameters `ORDREF, WLC, LINCAT`. The methods of order definition and wavelength calibration `DEFMTD, WLCMTD` must be chosen, as well as required parameter values, depending on the chosen methods.

Order Definition: Methods STD and COM

If `DEFMTD` is set to `STD or COM`, `WIDTH1` is the width of a single echelle order. To find this value load the `ORDREF` on the display using `LOAD/IMAGE` and use for example the commands `GET/CURSOR` or `EXTRACT/TRACE` and `PLOT/IMAGE TRACE`. A superior method which will be needed later in the reduction any way is to extract a column and to produce a plot on the hardcopy device. Let `x` be a column somewhere in the middle of the frame. The following sequence of commands will do the job:

```
SET/PLOT BIN=ON
PLOT/COL ima @x
SEND/PLOT ps2usr0
```

The default method for `DEFINE/ECHELLE` is `STD`. However, it happens that the order definition procedure mixes order and inter-order: this is likely to occur in the blue where the order spacing becomes less than the width of each order. Set the parameter `DEFMTD=COM`

Order definition: Method Hough

The method described above is recommended because it is quick and usually efficient. However an alternative method is available which enables you to process images of lower quality than required by `DEFINE/ECHELLE` and generally returns more accurate results. The price to be paid however is a higher demand for CPU time. The algorithm accepts frame which can be mildly contaminated by particle hits, bad columns, order gaps due to absorption lines, as found in `FLAT or STD`. If the frame is of very low quality for the order definition, as it may be the case for a science frame, an initial cleaning is required.

The algorithm assumes that:

- the interorder background is roughly constant over the frame.
- the slope of the orders is in the range]0.,0.5[

The method is enabled by the parameter:

```
SET/ECHELLE DEFMTD=HOUGH
```

The command can run in a fully automatic mode if all three parameters `NBORDI, WIDTHI, THRESI` are set to zero. To enforce the detection of a given number of orders, set the value of `NBORDI`. If the frame has a low contrast or if some areas of the interorder background are brighter than faint orders in the image, it may be useful to set `WIDTHI`. Given hw the half width of the orders and io the mean interorder distance, the optimal value of `WIDTHI` is in the range $]hw, io - hw[$. The threshold `THRESI` is normally estimated for each order independently. However the value can be enforced by giving a non null value to `THRESI`.

See also the help file of the commands DEFINE/HOUGH and HOUGH/ECHELLE for more details about the possibilities of this method.

If the image is overscanned, that is includes unsensitive areas in its lower and upper parts, it is better to avoid this areas by use of the command `SCAN/ECHELLE`.

The best way to use this command is to start with null value for all three parameters `NBORDI, WIDTHI, THRESI`. For example:

```
INIT/ECHELLE
SET/ECHELLE DEFMTD=HOUGH ORDREF= ...  (order reference frame)
SCAN/ECHELLE ORDREF CURSOR
DEFINE/HOUGH
```

The successive steps are the following:

- The image `ORDREF` is smoothed by `FILTER/MEDIAN`, a background value is estimated as the minimum of the smoothed frame in its central part and this value is subtracted from the frame. The result is stored in middummi.bdf. The frame middummi is displayed. this image must show a uniformly black background and a good contrast of the orders against the background. If this step does not provide a satisfying result, use another order reference image or clean it using general `MIDAS` commands. For a background estimate without previous order definition, consider the echelle command `BACKGROUND/SMOOTH`.

- The Hough transform of the middummi is computed and stored in middummh. A description of this step and of the cluster detection can be found in Ballester, 1991, Finding Echelle Orders by Hough Transform, Proceedings of the 3^{rd} ESO/ST-ECF Data Analysis Workshop, pp. 23-28. The image middummh.bdf must show clear accumulation peaks, roughly aligned along the columns and with a characteristic "butterfly" shape. The number of peaks is equal to the number of orders. This step is usually straightforward. It may happen however that bright straight lines in the Hough transform are generated by uncorrectly removed particle hits.

- The cluster detection is performed on middummh. The slope of detected orders must be rather constant, as well as the FWHM of the first bright orders. The FWHM measured on the first order is taken as reference. The displayed values provide an initial guess for WIDTHI. A visual check of the cluster detection is available by the low level command:

  ```
  HOUGH/ECHELLE middummi P5=NMV
  ```

  The result of the cluster detection is a table middummr.tbl containing slope, intercept, fwhm and peak value of the detected orders. To enforce the behaviour of cluster detection, set the parameter WIDTHI to a value big enough to remove correctly the detected features.

- The orders are followed individually and a threshold is estimated for each order. The threshold must vary continuously with the order number, and the detected positions must be in sufficient number for each order. The table order.tbl is created by this command. The measured positions can be visualised on the display window by the commands:

  ```
  LOAD middummi scale=...  cuts=...
  SELECT/TABLE order all
  LOAD/TAB order :X :Y :ORDER
  or
  LOAD/TAB order :X :Y
  ```

  Descriptors START and STEP of middummi must be equal to 1.,1. This step can be controlled by the parameter THRESI.

- A bivariate polynomial is fitted to the table and outliers are discarded by a kappa-sigma algorithm. The algorithm should not discard more than say, 10 percent of the positions and the resulting fit must be better than 0.3 pixels. The iterative regression and the kappa-sigma clipping are controlled by the low level command REGRESSION/ECHELLE.

Wavelength calibration

The parameter WIDTH2 is the width of the calibration lines. The parameter THRES2 defines the threshold above the local background to detect the comparison lines. A way to estimate this value is to load the WLC on the display. Then choose an order with few bright lines and extract this order using EXTRACT/TRACE. Produce a plot via PLOT/IMAGE TRACE and choose the threshold level such that 7-12 lines (per order) will be above this level. To produce a hardcopy of the WLC give the commands:

```
LOAD/IMAGE image
COPY/DISPLAY
```

Now we can execute

```
CALIBRATE/ECHELLE
or
CALIBRATE/ECHELLE DONE ! If the order definition is already done.
```

Note

For EFOSC reduction, the recommended method is TWO-D. In this method a bivariate polynomial is fitted to the initial identifications, instead of the monovariate polynomial involved in methods PAIR and ANGLE.

This command is the first step in the reduction. It finds the positions of the orders and extracts the comparison lines. The user is prompted for the identification of two calibration lines on the image display, repeated in overlapped regions of adjacent orders. For a detailed description of the wavelength calibration, see Section D.10.1. This identification is performed with the cursor. In case the orders are not overlapped, use the method ANGLE:

```
SET/ECHELLE WLCMTD=ANGLE
CALIBRATE/ECHELLE DONE
```

which performs the wavelength calibration differently, this time prompting for a minimum of four identifications anywhere in the spectrum.

Press Enter in the control box of the display and provide the asked order number and wavelengths.

If the automatic identification procedure does not converge to a satisfactory solution, use the level 1 command:

```
IDENT/ECHELLE [wlc] [lincat] [dc] [tol] [wlcloop] [wlcmtd] [guess]
```

to compute new dispersion coefficients. Use commands:

```
PLOT/RESIDUAL
SEND/PLOT VERSATEC
READ/HISTOGRAM line.tbl :RESIDUAL 0.08 -0.04,0.04
```

to display the results. You have to specify all parameters, because you are dealing with level 1 commands. The definition of the orders should give a standard deviation of less than 0.5 pixel and the dispersion coefficients should be fitted to better than 0.2 pixel.

The extracted orders in the pixel domain can be resampled into wavelengths with the command REBIN/ECHELLE. A method to verify the wavelength calibration consists of extracting and resampling the wavelength calibration frame and displaying the resampled image with a large scaling factor on the Y-axis. The variation of wavelength coverage from one order to the next should be smooth. Different regions of the resampled image and in particular the overlaps can be verified with the command PLOT/SPECTRUM.

```
EXTRACT/ECHELLE {WLC} &w
REBIN/ECHELLE &w &r
PLOT/SPECTRUM &r 4000,4050
```

Solutions are computed by default for each independent order (SET/ECH WLCOPT=1D). They can also be computed using a global bivariate polynomial (SET/ECH WLCOPT=2D).

Once a solution has been obtained for a particular set-up, using either of the echelles, you should save the results via

```
SAVE/ECHELLE name
```

For all the observations done with the same instrumental set-up, one can now use in CALIBRATE/ECHELLE this dispersion relation as a first guess to the final coefficients by specifying the parameter GUESS:

```
SET/ECHELLE WLCMTD=GUESS GUESS=name
```

The main advantage is that when using the GUESS option, no interaction is required to identify the lines in CALIBRATE/ECHELLE. After searching for the lines which are to be used for the determination of the dispersion relation the program applies as a first guess the coefficients used in the session previously saved and then starts the automatic identification.

The command CALIBRATE/ECHELLE has proved to be extremely flexible and reliable. However, the first-time users might run into some problems. The most frequent problems are listed below and a suggestion for their cure is given:

- *The wavelength calibration in a certain wavelength interval is wrong:* There is probably a wrong identification in the table. Give the command LOAD/IDENTIFICATIONS and proceed as indicated above.
- *The wavelength calibration is wrong and apparently no identification error was made:* There are too many lines per order. Try a higher threshold in the line detection.
- *There are *** NOT ENOUGH LINES *** per order to fit a polynomial of order three to get the dispersion coefficients:* The threshold was set too high. Try a lower one and use the GUESS option with the previous solution as the starting point. If, however, this problem is restricted to one or two orders this does not have to be serious because in those orders the dispersion coefficients will be deduced from a global fit to the whole set of orders. It is also possible to determine a bivariate solution (WLCOPT=2D)

Background Correction

Two background corrections can be performed with the package:

Interorder background: The scattered light in the interorder region is measured at a certain number of positions and the background map is provided by polynomial or spline interpolation. Interorder positions can be visualized with the commands:

```
SET/ECH BKGVISU=YES
LOAD/ECHELLE
```

and the interpolation and background subtraction is performed using `SUBTRACT/BACKGROUND`. This command also update the image descriptors to avoid background subtraction to be performed again by the `FLAT`, `RESPONSE` or `REDUCE` commands.

Sky background: Sky windows are defined on both side of the orders with the comamnd `DEFINE/SKY` and sky background is extracted using `EXTRACT/SKY`. The extracted sky can be subtracted from the extracted spectrum like in:

```
DEFINE/SKY object 2 CURSOR
EXTRACT/SKY object sky FILTER
SET/ECHELLE EXTMTD=AVERAGE SLIT=8
EXTRACT/ECH object ext
COMPUTE/IMAGE sub = ext - sky
```

Flat Field correction

After successful completion of the command `CALIBRATE/ECHELLE`, the next step in the reduction is the preparation of the `FLAT`s, i.e. the subtraction of the background level and the preparation of the file which is needed to normalise each `FLAT`.

The flat field correction is an optional step, it may not be needed in the case of objects exposed in the short wavelength range of the echelle (images centered at $\lambda < 5000$Å). If you do not want to perform a flat field correction, set the echelle keyword `FFOPT` to `NO` and skip this section. Otherwise, type:

```
SET/ECHELLE FFOPT=YES FLAT=...
```

It is advised to replace the default names ffcorr and blaze for the output files by some readily identifiable ones via the command:

```
SET/ECHELLE CORRECT=...  BLAZE=...
```

All paramters corresponding to the Background, Extraction and Rebin (sections 2, 4, 7) in the `SHOW/ECHELLE` must be set. See `MIDAS` User Manual, Vol. B, Chapter 8 (Echelle Spectra) for more details on the different background and extraction methods.

The command to reduce the `FLAT`s is simply:

```
FLAT/ECHELLE [flat] [correct] [blazecorr]
```

If you are following the tutorial, a plot will appear after a while in order to monitor the quality of the process. You will see a vertical trace of the raw flat field on top of the fitted background. After a short while a frame will appear on the image display showing the residuals of the fit at the grid points which have been used to determine the background. As explained in Section **Background definition** the background is fitted to points located in the interorder space.This residual map is stored in the intermediate frame `&Z`

The sequence described above should be applied both to the standard stars and objects with their corresponding flatfield and wavelength calibration images. Use the command `SAVE/ECHELLE name` to save the relevant files containing the result of the calibrations. The next step is the computation of the instrument response. It will involve the observed standard star(s). If you do not have one available, you could still use this procedure assuming that your object is itself a standard star with flux unity. If you are following the tutorial, the table `LTT1020` with absolute fluxes of this standard will be copied from the area `MID_CASPEC` into your work space by the command `SET/ECHELLE FLUX=LTT1020`. Other tables are also available or you can prepare your own table or use the table `UNITY`.

The following parameters have to be supplied: `SLIT`, `OFFSET`, `SAMPLE` and `FLUXTAB`, the identification of the standard star table which is to be used. To determine `SLIT` and `OFFSET` use the same procedure that was used on the `FLAT` to determine `WIDTH1`. To determine the `OFFSET` of the location of the orders for the object with respect to the flatfield the same column should of course be specified. `SAMPLE` determines the step with which the data in the final image will be sampled. When all the parameters are set, give the command:

```
RESPONSE/ECHELLE
```

This command performs the following tasks. It determines the background in the standard star and subtracts it. Then it divides by the `FLAT`, extracts the orders and rebins them. The counts are reduced to an exposure time of one second and the orders are binned into the same wavelength intervals as the corresponding standard star table (at present this means 12 Å intervals). The table values are divided by the observed counts to give the conversion to absolute flux units, the response. The response frame is lastly interpolated to the resolution required by the parameter `SAMPLE` using a low order polynomial. In order to cancel effects introduced by differences in the `FLAT` for the standard star and the object, the flatfield normalisation is applied. To check the accuracy of the calibration, the standard star is reduced as if it was an object. The response correction is applied to the `STD` and the individual orders are merged to form a one-dimensional spectrum. A description of these last steps is given in the part which describes the command `REDUCE/ECHELLE`.

The instrument response is called `RESPONSE` by default, but you could assign a different name, with the command

```
SET/ECHELLE RESPONSE=yourname
```

before executing `RESPONSE/ECHELLE`. If you do not want to correct for the instrument response, assign the value `NO` to the `RESPOPT` parameter as

```
SET/ECHELLE RESPOPT=NO
```

and skip the `RESPONSE/ECHELLE` command.

`REPONSE/ECHELLE` does not require any user interaction. It is advised, however, to monitor the intermediate results which appear at various stages on the screen and to inspect some of the intermediate files. Two steps are particularly delicate. The first one is the fitting of the background. As already mentioned before one should not specify too high an order for the 2D-polynomial. If `BIAS` and `DARK` frames have been subtracted a mere offset (degree 0,0) might be sufficient, of course depending on the exposure level. A careful inspection of the residuals of the background fitting as produced by `BACKGROUND/ECHELLE` will help in deciding the optimal choice. Small errors in the background fit will, at the edges of the orders be amplified due to the correction for the blaze. This may critically influence the results on the final, merged spectrum.

The second delicate step is the interpolation of the response frame to the required wavelength step. The sampling step of the table is 12 Å. This means about 10 calibration points per order. However discrepant pixels in the response frame which occur either at the end or beginning of an order can influence the fit. The effect is therefore again serious because the correction at the edges of the orders will be larger.

`RESPONSE/ECHELLE` will display on the monitor the resulting frame showing the response of the instrument at the resolution of the standard star spectrum specified by `SAMPLE`. Any discrepant pixels can be readily identified. To remove these, use the command

```
REPEAT/ECHELLE
```

(It is not a *nice* name but you are repeating some of the steps ...) This command will execute `MODIFY/AREA`. Identify, using the cursor, all pixels which should be excluded from the polynomial fit. When this is done, the command will execute the same set of steps as `RESPONSE/ECHELLE` and ultimately produce a new version of the `RESPONSE` file.

Even after this editing, the resulting response might still lead to unsatisfactory results. The reason is quite obvious. The sampling step of the standard star table is too large and therefore the number of reference pixels per order is too small, especially if one moves towards the blue end of the spectrum or if one has used the binned read-out mode. A possible way out of this problem is to use standard star tables sampled at about 3 Å instead of 12 Å . In the current version, standard star tables sampled at lower steps are obtained by interpolating the 12 Å tables. Note that the intrinsic resolution of the tables remains low and narrow (absorption) lines which are present in the standard star observation will not be present in the table. Test runs have shown that the response computed using a 3 Å sampling is superior to the one using the tables sampled at 12 Å . Absorption lines which are not present in the table give of course a wrong response correction but these regions can be easily edited using `REPEAT/ECHELLE`. Real high resolution tables will become available in future versions.

At this stage, everything is ready to reduce your observed spectrum. To reduce the object give the command `INIT/ECHELLE` to initialise the tables belonging to the object. Instead of specifying the wavelength step in Å one should now specify a reference file which determines the sampling of the final spectrum. This is normally the response file. Do not forget to specify also the parameter RESPONSE.

All parameters in sections Background, Extraction, Rebin, Flat-Field Correction, Response Correction, Merging as displayed by SHOW/ECHELLE must be checked.

Now type:

```
REDUCE/ECHELLE input output
```

where input and output are the input object frame and the one-dimensional output spectrum respectively. REDUCE/ECHELLE follows essentially the same steps as RESPONSE/ECHELLE. The background is determined and subtracted, the image is divided by the flatfield and the echelle orders are extracted and rebinned; the counts are normalised to an exposure time of one second. The orders are not yet corrected for the differential extinction between STD and OBJ. They are multiplied by the response and by the flatfield normalisation. As in RESPONSE/ECHELLE the determination of the background is a delicate step which needs careful monitoring.

Individual orders can be extracted using the level 1 command

```
MERGE/ECHELLE & name order1,order2 NOAPPEND
```

where & is a temporary file name (WARNING: This file will be deleted by the system when you initialise MIDAS again) created by REDUCE/ECHELLE. The parameter name is the root for the output file names to which the order number will be appended, parameters order1,order2 define the order range and NOAPPEND defines this "method of merging" orders. You will find all this in the tutorial. The individual orders can be plotted to check the accuracy of the method in the region where the orders overlap.

These steps, only four high level commands plus the corresponding SET/ECHELLE definitions, are the standard reduction. Depending on the type of observations, the operation can be optimised by running the commands in batch mode. Use the interactive command CALIBRATE/ECHELLE at a MIDAS work station and save the session status with the command

```
SAVE/ECHELLE name
```

where name is the session name. The rest of the reduction can be done completely in batch mode at some other time, probably at night, creating a procedure like

```
INIT/ECHELLE name1
SHOW/ECHELLE
FLAT/ECHELLE
RESPONSE/ECHELLE
REDUCE/ECHELLE input1 output1
REDUCE/ECHELLE ...
INIT/ECHELLE name2
...
```

where name1 ... are names of previously saved sessions, and input1 ..., output1 ... are the raw image and reduced spectrum, respectively. The command RESPONSE/ECHELLE can be excluded from the batch procedure given the interactive editing required.

D.10.2 Reduction without Standard Star

In the scheme described in the previous Section, the correction for the blaze function and for the chromatic response is done using standard stars. When the input data sets do not include standard stars it is possible to correct for the blaze function using a model of this effect as described in Chapter 7.

This alternative reduction scheme is called `IUE` method because of its resemblance to the IUE spectra reduction. The first steps of the reduction are identical to the standard method, i.e. one should start with `CALIBRATE/ECHELLE` and after that run `FLAT/ECHELLE` optionally. The command `RESPONSE/ECHELLE` should of course be skipped. The IUE-mode is set via

```
SET/ECHELLE RESPMTD=IUE
```

To reduce the data enter the command

```
REDUCE/ECHELLE input output
```

where `input` is the raw image,
command

```
MERGE/ECHELLE output name order1,order2 NOAPPEND
```

where `name` is the root of the name for output files with up to 4 characters, `order1,order2` defines the order range and `NOAPPEND` is a required command option.

D.11 Saving the Data on Tape

The resulting files can be saved on tape in FITS format for later use. Be sure to save the data before your disk reservation expires. Saving data on tape requires generating a catalogue with the filenames to be saved, this is done with the command

```
CREATE/ICAT catname dirspec
```

Files in this catalogue will be copied on tape as follows

```
OUTTAPE/FITS catname tapeunit
```

The output tape will be in FITS format, 1 block per record, and 1600 bpi density. Look into the command help for other options.

D.12 Instrument Description: CASPEC

In the present version we will be concerned with the standard configuration of CASPEC, the Cassegrain Echelle Spectrograph which is in operation at the $f/8$ Cassegrain focus of the ESO 3.6 m telescope. The instrument has been described in detail by D'Odorico and Tanné (1984) we will include here a summary of the main instrumental characteristics which are relevant for the analysis of the data.

In its standard configuration CASPEC uses a a 31.6 lines/mm echelle grating together with a 300 lines/mm grating cross disperser. A short focal length camera (f/1.46) focuses the beam into a thinned, back illuminated CCD consisting of 320 $\times$ 512 $30\mu m^2$ pixels. One pixel on the detector corresponds to an entrance aperture or 1.2 $\times$ 0.7 seconds of arc on the sky, the first dimension being in the dispersion direction. The table D.1 shows the change in resolution with the order number.

Order	**Resolution** (Å/pixel)	**Central Wavelength** (Å)
140	0.125± 0.01	4062.21
130	0.133	4374.95
120	0.143	4739.34
110	0.157	5170.49
100	0.174	5687.5
90	0.194	6319.2
80	0.217	7109.6
70	0.244	8120.

Table D.1: Resolution

In this configuration, the spectrograph records in a single CCD frame an $\sim$ 900 Å wide portion of the spectrum of objects with $V \leq 15$ with a resolving power of $\sim 20,000$ and a signal-to-noise ratio ≥ 10. This magnitude limit is set primarily by the readout noise of the chip (50 electrons rms) and by the maximum exposure time of $\sim$ 120 min before contamination by cosmic rays becomes a problem. Fainter objects can be observe at lower resolution by binning the CCD data.

Scattered light.

Assuming a plane grating, used in near–Littrow mode, the blaze function R at wavelength λ is approximated by

$$R(\lambda) = \frac{\sin^2 \pi\alpha X}{(\pi\alpha X)^2} \tag{D.1}$$

where α is a grating 'constant' with value between 0.5 and 1, and $X = m(1 - \lambda_c(m)/\lambda)$, in which m is the order number, and $\lambda_c(m)$ is the central wavelength of order m. Both parameters are related through the grating 'constant' k by $k = m\lambda_c(m)$. In table D.2 we include approximate values for the parameters k and α. These are mean values, given that the actual values for a given observation are a function of the order number and depend also on the instrumental set up.

Grating lines/mm	k	α
31.6	568746.	0.8
51.	344705.	0.8

Table D.2: Blaze parameters

D.13 Summary of reduction options

The two following diagrams summarise the successive steps performed by high-level procedures `RESPONSE/ECHELLE` and `REDUCE/ECHELLE` depending on the main options:

- Flat-field correction can be performed or not
- Instrumental response correction can be skipped, or performed using either a standard-star or IUE-like methods.

Command `RESPONSE/ECHELLE`	
No Flat-Field Correction	Flat-Field Correction
Check exposure time	Check exposure time
Subtract background from `STD`	Subtract background from `STD`
	Divide by the corrected flat-field
Extract STD spectrum	Extract STD spectrum
Rebin	Rebin
Divide by the exposure time	Divide by the exposure time
Adapt STEP for STD and flux table	Adapt STEP for STD and flux table
Convert the flux table into image	Convert the flux table into image
Divide the flux table by STD spectrum	Divide the flux table by STD spectrum
Filter the response (median,smooth)	Filter the response (median,smooth)
	Multiply by the blaze flat-field

Figure D.1: Response/Echelle

Command `REDUCE/ECHELLE`: Standard Star method	
No Flat-Field Correction	Flat-Field Correction
Check presence of exposure time	Check presence of exposure time
Subtract background from `OBJ`	Subtract background from `OBJ`
	Divide by the corrected flat-field
Extract orders	Extract orders
Divide by exposure time	Divide by exposure time
Rebin	Rebin
(Multiply by instrumental response)	(Multiply by instrumental response)
	Multiply by blaze function of flat-field
(Merge)	(Merge)

Figure D.2: Reduce/Echelle (1)

Command `REDUCE/ECHELLE`: `IUE`-like method	
No Flat-Field Correction	Flat-Field Correction
Extract raw spectrum	
Rebin raw spectrum	
Estimate the background of `OBJ`	Subtract background from `OBJ`
	Divide by the corrected flat-field
Extract background spectrum	Extract `OBJ` spectrum
Rebin background spectrum	Rebin `OBJ` spectrum
Subtract backgr. from raw spectrum	
	Multiply by blaze function of the flat-field
Correct for blaze by `RIPPLE/ECH`	Correct for blaze by `RIPPLE/ECH`
Merge	Merge

Figure D.3: Reduce/Echelle (2)

D.14 XEchelle

This chapter describes the use of the Graphical User Interface XEchelle. The interface XEchelle generates and sends MIDAS commands to the monitor. Therefore, all operations performed with the interface can also be performed manually by typing the commands on-line or writing a procedure. The interface is however a convenient additional layer of the software providing on-line help, visibility over the parameter values and avoiding syntax problems. These functions are particularly useful for parameter intensive procedures such as batch reduction.

Note

Using XEchelle requires that the described MOTIF GUIs have been installed on your system. Also this chapter must be considered as an operating manual of the graphical interface and assumes a general understanding of the Echelle context.

In this Section the following notations have been used:

Commands or keywords are indicated as:	`SET/ECH WLC=thar.bdf`
Push-buttons and Menus are surrounded by a box:	`[Identify]`
and Labels or names are written in bold face:	**Calibration Frame:**

D.14.1 Graphical User Interface

The Main window of the interface XEchelle, presented in Figure D.14.1 is created with the command:

Midas...> `CREATE/GUI ECHELLE`

To exit XEchelle you must select in the `[File]` menu the `[Exit]` option or push the `[Close Window]` button located at the lower-right corner of the main panel. The Main Panel of the interface includes the following elements:

a) A menu bar

a) A directory and file browser

a) A calibration and reduction steps area

c) A short help

c) A set of push-buttons

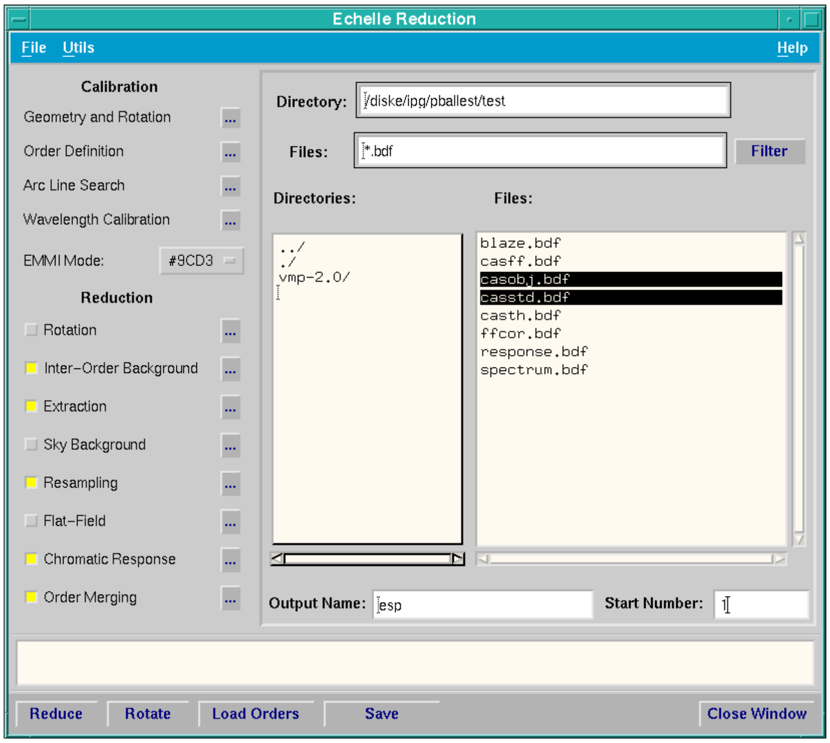

Figure D.4: Main window of the GUI XEchelle

Initializing Keywords

The parameter area is used to set and display the various parameters used by the context. These parameters may be either defined by the user or defaulted by the context. User defined parameters are set or changed by simply moving the mouse cursor to the relevant field and typing in the value. The corresponding `SET/ECHELLE` command is sent to MIDAS as soon as the mouse is moved out of the field. The command is echoed in the MIDAS monitor window.

Short Help

The short help is a small window located immediately below the parameter area, and is updated as the cursor moves on the different components of the interface to provide short information on the parameters and the possible actions.

Sending Commands

The buttons are located immediately below the short help. In colour terminals, the button labels have different colours which group them by functions, usually distinguishing between processing and verification commands.

On-line HELP facility

The interface incorporates an on-line HELP facility. A comprehensive description of the functions of each panel of the interface is obtained by clicking the `Help...` button in each panel, as in Fig. D.14.1. In the Main Panel, a `Help` menu provides general information on the context. The Help messages are displayed in a dedicated window.

Entering File Names

Input fields expecting file names can usually pop up a file list selection by clicking the `...` button located on the right side of the the text field.

Dialog Windows

In addition, XEchelle uses dialog windows to input specialized parameters necessary for the different reduction steps. These windows are popped up by pushing the three-dots `...` button corresponding to each calibration and reduction step on the left-hand side column of the Main Panel. These windows contain text fields, option menus and radio buttons. Values are given by moving the mouse inside the parameter fields or selecting an option by clicking on the relevant button. Push-buttons yielding to a dialog window are indicated by three dots in the label. For example, clicking on `Sky Background ...` pops up the Sky Background window. This window can be closed by clicking its `Cancel` button.

Saving and Loading Session Parameters

A number of parameters appear in the parameter area which correspond to the status of the package when the interface is created. These parameters can be changed by typing the new values in the field, or reading them from a previously saved session table. The option `Open` in the menu `File` may be used to read a session file. The name of the currently loaded session is indicated in the first text field **Session** at the top of the parameter area in the Main window. Selecting the option `Save` in the menu `File` will save the session keywords in this table. This option can be used to save intermediate reductions, whereas the option `Save As...` allows to specify a new table name.

Performing Batch Reduction

The Batch Reduction window allows catalogs of observations to be processed and a data reduction scheme to be defined dynamically. The different steps like Bias, Dark, Flat-Field correction, *etc.* are optional and must be set by clicking on the option buttons located on the left side for each reduction step in the Main Panel.

Figure D.5: Sky Background window

The input files for the reduction are selected in the file browser. For example on Fig. D.14.1, pushing the `Reduce` button will apply the reduction procedure to the two files **casobj.bdf** and **casstd.bdf**.

Appendix E

PISCO

E.1 Introduction

PISCO is the ESO **P**olarimeter with **I**nstrumental and **S**ky **CO**mpensation, a two–channel filter polarimeter using photon counting and offered on the ESO/MPI 2.2-m telescope. A description of its design and operation can be found in ESO Operating Manual No. 13 and references therein. Appendix A of that manual also discusses in detail the mathematical principles around which the MIDAS reduction program for PISCO data has been written.

E.2 Data Format

Formally, the raw data files produced by the PISCO data acquisition programme conform to the standards of FITS images or MIDAS bulk data frames and can, therefore, be treated with all MIDAS commands pertaining to images. By contrast, the content of these data files is very highly non-standard and is a mixture of spectrum-like images and tables (does not apply to calibration frames, cf. below).

As described in the PISCO Operating Manual, there are two types of data files:

Calibration frames contain 8 lines with 32 data points each which correspond to the 32 positions of the half-wave plate. In the first seven lines, seven measurements are stored; the eigth line holds their normalised mean.

Normal data frames also consist of 8 lines with 32 data points each. However, their contents is as follows:

Line 1: raw data of channel X in the direct mode of the compensating phase plate.

Line 2: raw data of channel Y in the direct mode of the compensating phase plate.

Line 3: raw data of channel X in the rotated mode of the compensating phase plate.

Line 4: raw data of channel Y in the rotated mode of the compensating phase plate.

Lines 5-7: results of on-line reduction as described in the Operating Manual. They are not used by MIDAS.

Line 8: not used at present.

E.3 Data Reduction

After having enabled the PISCO context (command SET/CONTEXT PISCO), the basic reduction of PISCO data involves merely one command, REDUCE/PISCO, which starts a routine (pisco.prg) solely consisting of other MIDAS commands. So, the whole procedure can be followed when the ECHO facility is enabled (see description for ECHO/ON or ECHO/FULL).
The essential steps are: (1) division of the data by the integration time (stored in descriptor O_TIME), (2) subtraction of the sky and the offset (using a sky measurement, if available, or a dark measurement), (3) calibration (division by calibration file) and (4) Fourier transformation. As is described in Appendix A of the PISCO Operating Manual, the last step, a simple Fourier transform using the MIDAS command FFT/POWER, is the core of the reduction of PISCO data.
The procedure is designed to handle an entire set of data by operating on an input catalog and producing a single output table in which each row holds the results for one data frame in the input catalog. There are no free parameters except for a *mode*, which can be set to 1, 2, or 3 depending on whether the two sky channels X and Y shall be reduced separately, together, or separately and together, respectively. Normally, the X and Y channel should be treated together.

column label	content
IDENT:	identifier, copied from descriptor IDENT of raw data file
UT:	universal time (hours), copied from descriptor O_TIME(5) of same data file
X_INTENSITY:	mean of 1^{st} line from data file, divided by integration time O_TIME(7), sky subtracted
Y_INTENSITY:	mean of 2^{nd} line from data file, divided by integration time O_TIME(7), sky subtracted
POLXY:	degree of polarisation in channels X+Y (mode=3)
ERRORXY:	error of :POLXY
ANGLEXY:	position angle of E-vector in channels X+Y (mode=3)
POLX:	degree of polarisation in channel X (mode=1)
ERRORX:	error of :POLX
ANGLEX:	position angle of E-vector in channel X (mode=1)
POLY:	degree of polarisation in channel Y (mode=2)
ERRORY:	error of :POLY
ANGLEY:	position angle of E-vector in channel Y (mode=2)

Table E.1: Format of output table produced by REDUCE/PISCO

The program attempts an error estimate from the noise level in the Fourier-transform. The polarisation angles still have an arbitrary zeropoint and need to be transformed to a standard system by means of observations of standard stars. Apart from the comparison with calibration data, the results of REDUCE/PISCO should be final.

The output table is a standard MIDAS table so that all applicable table commands can be used to further process the results. The columns of this table are shown in Table E.1.

Appendix F

IRSPEC REDUCTION

F.1 Introduction

IRSPEC is a cryogenically cooled grating spectrometer equipped with a relatively small (for optical CCD standards) 2D array of 62x58 pixels. It covers the 1 μm to 5 μm wavelength range and the two back-to-back gratings – #2 optimized for observations at $\lambda \leq 2.5$ μm and #1 for longer wavelengths – yield a dispersion of $\simeq 5$ Å/pixel in H and K (1.4-2.5 μm), $\simeq 10$ Å/pixel in L,M (3-5 μm) while in J (1-1.3 μm) one can choose between $\simeq 2$ Å/pixel (grating #2 order 2) or $\simeq 3.5$ Å/pixel (grating #1 order 3). In all cases the resolving power is quite high and each frame covers a small wavelength range; obtaining complete spectra of an atmospheric window requires $\sim$20 images at different grating positions. Along the slit the pixel size is $\simeq 2.2$ arcsec and the total, usable slit length is slightly less than 2 arcmin; small areas at the edges of the slit are vignetted.

Although there is no standard procedure for the observations, it is generally agreed that the most convenient strategy to subtract the strong sky emission – OH airglow up to 2.25 μm and thermal + molecular bands beyond – is to alternatively measure the 'object' and the 'sky' and subtract the frames (note that the 'sky' frame may also be taken with the object in another position along the slit). On chip subtraction of the sky emission - – the procedure normally used for optical CCD's – gives worse results due to the large intensity of the sky background combined with the relatively bad cosmetic quality of the infrared arrays. The object-sky telescope nodding is typically done every few minutes and one or more frames per telescope positions are produced and stored. At the end of the night, therefore, one has collected several hundred images which may also include data taken at many different wavelengths. This document describes a few procedures (commands) developed to make the handling of the many frames somewhat easier. For each command I did my best to provide you with long and exhaustive help files which are available on line. Here I just briefly describe the steps to follow to take full advantage of the commands available (remember to type SET/CONTEXT IRSPEC to access them).

F.1.1 A typical reduction.

You have one or more pairs of 'objects' and 'sky' frames for your astronomical target, the same for a 'standard' star. You also have one or more flats (halogen lamps up to 2.5 μm and sky frames beyond) and dark exposures. The first thing to do is creating a normalized flat (FLAT/IRSPEC) and to store it, together with the dark, with SET/IRSPEC. Then use SKYSUB/IRSPEC which does a 'clever' obj-sky subtraction aimed at minimizing the residual sky lines in the output frame. The output frame is also flat-fielded and corrected for the fixed pattern of bad pixels in the array (you can use the default pattern or redefine it using DEFINE/IRSPEC, see also BADPIX/IRSPEC). You can then correct the tilt of the spectral line using RECTIFY/IRSPEC. The 2D frame can then be wavelength calibrated using CALIBRATE/IRSPEC; you can take advantage of the on line mechanical wavelength calibration - – usually precise within 1 pixel – or use reference line(s) (often the OH airglow lines in the sky image) in case very accurate wavelengths are needed. Once you have produced the 2D spectrum of your 'standard' star, use RESPONSE/IRSPEC to determine the instrumental response – i.e. the conversion factor between counts/sec and flux units – after having defined the star intrinsic flux by means of STANDARD/IRSPEC. The resulting response frame can be used to flux calibrate the spectrum of your astronomical target using FLUX/IRSPEC, and this is the end of the reduction.

There are also two commands of more general interest which allow you to subtract, row by row, a polynomial fit of the continuum from an image (SUBTRACT/IRSPEC) and to 'cleverly' merge together 1D spectra at different wavelengths into a table (MERGE/IRSPEC).

A last comment concerning integration times and count units. The IRSPEC images are the average of a number (NDIT) of read-outs each with a given integration time (DIT). Therefore, the counts level does not depend on NDIT and whenever you are asked to use frames taken with the same integration time it means that they must have been taken with the same DIT.

F.1.2 Notes on specific commands

- - SET/IRSPEC flat=....
The InSb array used in IRSPEC is not linear and the relative response of its pixels (i.e. the flat-field) varies slightly with the level of the signal (i.e. the count-level) of the image. The flat used should therefore have a count-level close to the astronomical frames, better if within a factor <2. There is also a small dependence on wavelength which makes it advisable to have at least one flat per photometric band (J,H,K,L). Up to K ($\lambda < 2.5$ μm) the flats can be derived from short exposures of the halogen lamp; at longer wavelengths one can directly use the sky frames.

- - SET/IRSPEC dark=....
The dark frame is used in
SKYSUB/IRS
and
CALIBRATE/IRS ... mode=d
and must be taken with the same on-chip integration time (DIT) as the astronomical

frame(s). However, in SKYSUB/IRS you can avoid caring about the dark if – and only if – you use this command to make a straight subtraction (see last example of TUTORIAL/SKYSUB), in which case the dark frame is effectively not used. In case you do not have darks at a given DIT you may try to derive them by scaling darks taken with different (longer) DITs.

- - SKYSUB/IRSPEC

When you have more than one pair of object and sky frames it is not trivial to decide whether it is better to apply SKYSUB to each pair and then average them or first average the objects and the skies together and then apply SKYSUB (N.B. you must average, not sum the frames otherwise you will get into trouble with the flux calibration). The best approach is probably to first use the second method (SKYSUB on the averaged frames) and try the second if you have obvious problems with the cancellation of the sky lines.

- - CALIBRATE/IRSPEC

The wavelength calibration of IRSPEC data is much simplified by the fact that the pixel-wavelength dispersion is linear to very high accuracy, and by the very precise mechanical positioning of the grating. The on-line (mechanical) wavelength calibration that you retrieve using

CALIB/IRS image

is already quite accurate (usually within one pixel) and may be enough for many purposes. It can still be improved using by manually determining with CALIB/IRS ima_ref mode=d

the shift (error) on an image containing one or more reference lines (e.g. the OH airglow lines in a sky frame); with this you can easily achieve accuracies of 1/3 of a pixel. When using this mode remember to apply – using the above command – the same calibration to the object and to the standard otherwise you will not be able to run FLUX/IRSPEC. Small shifts between the object and standard frames can be corrected within FLUX/IRSPEC using the shift=... option.

- -RECTIFY/IRSPEC

The analytical formula used to derive the tilt angle at a given central wavelength is quite accurate and should produce rectified spectra within a fraction of a pixel. The errors introduced by assuming that the slit images are straight and parallel to each other are also small.

- - STANDARD/IRSPEC

I decided to force the users to create 'flux' files themselves containing their best guesses for the flux of the 'standard' stars because the use of the photometric points is not necessarily accurate, even though there is little more one could do. I hope that a list of spectrophotometric – or even just spectroscopic – standard stars for the infrared will be soon available. In the meantime, I apologize to the users for the inconvenience which, however, should make them aware of the problems one encounters to find a star with a precisely known flux distribution and with a featureless spectrum.

- - TUTORIAL/SKYSUB, TUTORIAL/CALIBRATE

These procedures are meant to give you a complete list of examples for the use of these relatively complicated commands. To work they need a number of small files (called irstut....) in the 'DEMO' area (.../midas/demo/data), contact your local wizard to have them installed.

Appendix G

Reduction of Long Slit and 1D Spectra

G.1 Introduction

G.1.1 Purpose

As described in Chapter L, the reduction of long slit spectra comprises three main steps:

1. Pre-reduction:
   – removal of cosmic rays
   – flat-fielding
   – dark subtraction.
2. Geometric corrections:
   – correction for distortions perpendicular to the dispersion direction (*e.g.*, due to differential refraction in the earth's atmosphere)
   – determination of the dispersion curve
   – resampling of the data to constant step in wavelength.
3. Final photometric operations :
   – sky subtraction
   – flux calibration.

Section G.3 is a summary of a typical session but without much explanation on the methods or general advice. First-time users may find it useful to follow it as it stands, step by step, in order to become acquainted with the plain technical operation of the available commands. Section G.4 presents the Graphical User Interface XLong. The interface XLong is a convenient way to gain visibility on the parameter values, avoid command syntax problems and understand the organisation of the context.

Note

Although this Appendix is intended to be self-contained, it cannot serve as a substitute for the HELP *information available for all the individual commands* *mentioned. Full advantage of the flexibility of the tools offered for the reduction of long slit spectra can be taken only on the basis of the command's detailed explanations in Volume A and more specifically in Chapter 6 (long-slit spectra) of Volume B.*

G.2 Retrieving demonstration and calibration data

Calibration tables are required to provide reference values of wavelength for Th-Ar arc lamp lines, atmospheric extinction or standard star fluxes.

A certain number of tables are distributed on request in complement to the Midas releases. These tables are also available on anonymous ftp at the host **ftphost.hq.eso.org** (IP number 134.171.40.2). The files to be retrieved are located in the directory /midaspub/calib and are named README.calib and calib.tar.Z. Command `SHOW/TABLE` can be used to visualize the column name and physical units of the tables. Demonstration data required to execute the tutorial procedure `TUTORIAL/ECHELLE` are also located on this ftp server in the directory /midaspub/demo as echelle.tar.Z. FTP access is also provided on the World Wide Web URL:

http://http.hq.eso.org/midas-info/midas.html

The calibration directory contains other information such as characteristic curves for ESO filters and CCD detectors, which can be visualized with the Graphical User Interface XFilter (command `CREATE/GUI FILTER`).

G.3 A Typical Session: Cook-book

To ensure easy reading, the following notations are used in this Section:

Commands to be entered by the user are indicated by typewritter style: `SET/CONTEXT long`
similarly, keywords by boxes: $\boxed{\text{WIDTH}}$
and file names by bold face type: **CALIB, line.tbl**

As a useful complement to this Section, an on-line tutorial is available using the command `TUTORIAL/LONG`. The tutorial procedure is provided in Section 1.10 of Chapter 6, Vol. B of the MIDAS User Guide. The tutorial is activated by the sequence:

Midas...> `SET/CONTEXT long`
Midas...> `TUTO/LONG`

The tutorial requires demonstration files which are not included in the standard release of MIDAS. If these files are not already installed, they can be obtained by anonymous ftp or by request from the MIDAS Hot-Line

G.3.1 Getting Started

In the following description of a typical session, the spectrum to be analysed is assumed to have previously been corrected for bias, dark current and flat field and to be oriented so that the direction of dispersion is along the X-axis (wavelength increasing horizontally to the right).

Set the context

As for several other MIDAS applications, the long slit package is stored in its own context. Thus, to enable on-line `HELP` and for the use of commands and keywords pertaining to the reduction of long slit spectra one has to type:

Midas...> `SET/CONTEXT long`

Initializing keywords

The keywords contain all necessary parameters (see Chapter 6, Sect. 1.9 of the Vol. B for a complete list). `SET/CONTEXT long` will set all relevant keywords to their default values. To override the defaults, three possibilities exist:

1. Terminal input (see also MIDAS Users Manual Sect. 3.2.3, Volume A), e.g.:

   Midas...> `WLC = ''`**ccd0045**`      ''`
   Midas...> `DCX(2) = 4`

   Note that adding some blanks at the end of a string to be entered into a character-type keyword may be a good idea in case the previous contents was a longer character string. Also it is mandatory to leave at least one blank on each side of the equal sign (=).

2. Using the command `SET/LONG`. This form is safer since keyword names are checked and the syntax is more flexible. The command:

   Midas...> `SET/LONG WLC =`**ccd0045** `DC=,4`

   is equivalent to the two commands above. It can be noted that no blanks are necessary at the end of the strings and that spaces before or after the assignment symbol (=) can be freely added or removed. Keyword names can also be truncated to their shortest significant part.

3. Restoring the values saved in a previous session with:

   Midas...> `INIT/LONG` **session**

where **session** is the name of a table created at the end of a previous session by the command:

Midas...> `SAVE/LONG` **session**

Generally, if parameters required for commands of the long slit package are omitted, their values will be taken from the associated keywords. Also, if a value is provided on-line to a command, it will change the value of the corresponding keyword. To display the current keyword values, type:

Midas...> `SHOW/LONG`

Since there is an important number of keywords involved in the package, the command `SHOW/LONG` is organized in sections, *e.g.*:

Midas...> `SHOW/LONG g` for general formats information
Midas...> `SHOW/LONG d` for dark, bias and flat-field corrections
Midas...> `SHOW/LONG w` for wavelength calibration keywords
Midas...> `SHOW/LONG r` for resampling keywords
Midas...> `SHOW/LONG e` for extraction keywords
Midas...> `SHOW/LONG f` for flux calibration keywords

Information can also be obtained on the keywords with `HELP/LONG`, as in:

Midas...> `HELP/LONG wlcmtd`

which provides the type, default value, description and current value of the keyword $\boxed{\text{WLCMTD}}$. A list of keywords and command names of the context Long is provided in Chapter 6 of the Volume B (Sect. 1.8 and 1.9).

G.3.2 Reading the Data

Observation data must be converted to MIDAS data structures before processing and the spectra must be oriented so that the dispersion direction is along the rows of the images, with the wavelength increasing from left to right. The command `INTAPE/FITS` can be used to read FITS files from a magnetic tape or from disk. The commands `ROTATE/CLOCK`, `ROTATE/COUNTER_CLOCK` and `FLIP/IMAGE` enable to perform any rotation by a multiple of 90 degrees and image flipping without interpolating the pixel values. The command `REBIN/ROTATE` can be used for any rotation but in the general case will interpolate the pixel values.

G.3.3 Pre-processing the spectra

Observation pre-processing consists of standard bias, dark and flat-field corrections. The package provides different commands to average images, extract sub-images or normalize flat-fields. General information related to preprocessing commands is available in the section Dark, Bias and Flat-Field of `SHOW/LONG` and can be obtained by:

Midas...> `SHOW/LONG d`

The command `PREPARE/LONG` extracts sub-images. The pixel positions (x_{low}, y_{low}) and (x_{upp}, y_{upp}) of the lower-left and upper-right corners of the extracted sub-image can be specified with the keyword $\boxed{\text{TRIM}}$. The command processes a catalog by default or an image if the name is provided with the extension **.bdf**. The sequence:

Midas...> `CREATE/ICAT flatcat ff*.bdf`
Midas...> `SET/LONG TRIM=10,20,500,550`
Midas...> `PREPARE/LONG flatcat exf`

generates a catalog **flatcat.cat** referencing the files **ff*.bdf** and extracts the sub-window `[@10,@20:@500,@550]` (see help file of `EXTRACT/IMAGE`). The output files will be called **exf*nnnn*.bdf** where *nnnn* is a four-digit number starting at 0001.

Images can be combined using an average or median method with the command `COMBINE/LONG`. This command processes a catalog of images and generates a combined image. The two commands:

Midas...> `CREATE/ICAT flatcat exf*.bdf`
Midas...> `COMBINE/LONG flatcat avflat AVERAGE`

generate a catalog **flatcat.cat** referencing the files **exf*.bdf** and average them to produce the image **avflat.bdf**. The option `MEDIAN` can be used to eliminate particle hits when combining dark exposures. Flat-field normalisation can be achieved by the command:

Midas...> `SET/LONG BIAS=190.  FDEG=2`
Midas...> `NORMALIZE/FLAT avflat normff`

which performs a bias correction of value $\boxed{\text{BIAS}}$ on the image **avflat.bdf** and interpolates the flat-field continuum by a polynome of degree $\boxed{\text{FDEG}}$ to produce the image **normff.bdf**. It can be noted that flat-field normalisation can require a different approach than the one implemented in the command `NORMALIZE/FLAT`. The Section 1.2.4 (Flat-Fielding) of Chapter 6 in Vol. B presents alternative methods.

G.3.4 Getting the Dispersion Solution

The determination of the dispersion solution involves an arc lamp spectrum $\boxed{\text{WLC}}$ and a line catalog $\boxed{\text{LINCAT}}$. The parameters required for line detection, identification and wavelength calibration are displayed by the command:

Midas...> `SHOW/LONG w`

Line Detection in the Comparison Spectrum

The first step consists of detecting the emission lines of the comparison spectrum to produce the line table **line.tbl**. It can be noted that the command `SAVE/LONG` will store all keyword values as descriptors of this table, so that a session can be saved only if the table **line.tbl** is available. The commands:

Midas...> `SET/LONG WLC=`**ccd0003** `THRES=20.  WIDTH=8 YWIDTH=3 YSTEP=1`
Midas...> `SEARCH/LONG`

initialize the line searching parameters and `SEARCH/LONG` performs the following operations:

- Average 2*$\boxed{\text{YWIDTH}}$+1 rows of image $\boxed{\text{WLC}}$ in the Y direction at a step $\boxed{\text{YSTEP}}$
- for each resulting row, detect emission lines above a local background computed by median estimate in a sliding window of size $\boxed{\text{WIDTH}}$ and with a peak intensity $\boxed{\text{THRES}}$ counts above the local background;
- determine line centers by fitting Gaussians to the line profiles.

The parameter $\boxed{\text{YWIDTH}}$ can be used to improve the signal-to-noise ratio of each row by averaging adjacent rows. The parameter $\boxed{\text{YSTEP}}$ is by default equal to 1 which corresponds to the standard row-by-row method but can be set to a larger value to accelerate the wavelength calibration. The detections are stored in the following columns of the table **line.tbl**:

Column label	:X	– X-positions of spectral lines
	:Y	– Y-position (row)
	:PEAK	– peak intensity of lines

Results can be checked with the command `PLOT/SEARCH`. For one-dimensional spectra,

Midas...> `PLOT/SEARCH`

plots the central row $\boxed{\text{YSTART}}$ of the spectrum as well as the position of the detected lines. This behavior is obtained for 2D spectra by the additional option 1D, as in:

Midas...> `PLOT/SEARCH 1D`

whereas typing the command without option provides a bi-dimensional graph of the X- and Y-positions of the detected lines.

Line Identification

In order to compute the dispersion relation, lines stored in table **line.tbl** have to be cross-identified with a reference catalogue LINCAT. For the first few spectral lines in the central row, this has to be done interactively, the rest is automatic. The sequence:

Midas...> `SET/LONG LINCAT=`**hear.tbl**
Midas...> `IDENTIFY/LONG`

plots the central row of the comparison spectrum WLC on the graphics device and activates the graphic cursor. The lines are selected by positioning the graphics cursor on them and clicking the left-hand mouse button or Return. On the session terminal you will be prompted for the wavelength (to be entered in the same unit as in LINCAT). The command will search for a corresponding entry in table **line.tbl**. If it is found, the wavelength you enter interactively will be stored in the column :IDENT of table **line.tbl**. An asterisk, " * ", will delete a previously made identification. To exit, click the middle mouse button on the graphics device.

At least 2 lines have to be identified interactively, but it is safer to identify a few more, especially if the dispersion relation is sensibly non-linear. They should be evenly distributed over the relevant range in wavelength. The identifications can be checked with:

Midas...> `PLOT/IDENT`

which plots the central row of the comparison spectrum and the interactive line identifications.

Determination of the Dispersion Coefficients

The command `CALIBRATE/LONG` determines the dispersion relation for each row of the spectrum. In addition, a bivariate dispersion relation is computed if the keyword TWODOPT is set to `YES`, as in:

Midas...> `SET/LONG TWODOPT=YES`
Midas...> `CALIBRATE/LONG`

The command `CALIBRATE/LONG` determines row-by-row polynomial solutions of degree DCX(1) stored in the table **coerbr.tbl** and fits a 2-D polynomial of degree DCX(1),DCX(2) to the lines with entries in columns :X, :Y, and :WAVE of table **line.tbl**. The coefficients of the 2D polynomial are stored in the keyword KEYLONGD. The program performs a final oulier rejection which eliminates all lines with residuals larger than $\boxed{\text{TOL}}$ pixels (if `TOL` is positive). However the minimum number of lines in any given row will be $\boxed{\text{DCX(1)}}$+2. If this number cannot be obtained, a polynomial of lower order will be fitted. Three more columns are added to the table **line.tbl**:

:WAVE	identified wavelength
:WAVEC	wavelength computed from polynomial fit
:RESIDUAL	(= :WAVEC – :WAVE) Residual of polynomial fit from the tabulated wavelength)

Note

For a proper calibration, images should preferably have positive step descriptors and the wavelength should increase from left to right.

Possible Graphical Verifications

Midas...> `PLOT/CALIBRATE`

plots the central row of the comparison spectrum and adds the line identifications obtained. This plot can be used for coarse consistency checks. A hardcopy may be handy for future, similar work.

Midas...> `PLOT/RESIDUAL`

displays the residuals (column :RESIDUAL in table **line.tbl**) as a function of wavelength. This command is used to judge the quality of the calibration.

Midas...> `PLOT/DELTA`

plots the dispersion relation and the residual value as a function of the wavelength. Note that `PLOT/RESIDUAL` and `PLOT/DELTA` display the rms of the residuals. It is also possible to display, for all rows, the residuals for a given line of wavelength "wavelength" with:

Midas...> `PLOT/DISTORTION` "wavelength"

The wavelength value must correspond to a value from the catalog $\boxed{\text{LINCAT}}$. The residuals are plotted in the graphic window with a scale such that the full extent of the graph corresponds to one pixel of the spectrum.

Refining the Dispersion Relation

Due to discontinuities in the line detections as well as in the calibration process, the dispersion relations computed by the row-by-row process could show row-to-row discontinuous variations. The command `CALIBRATE/TWICE`, provided that a sufficient number of calibration lines is available, can stabilize the solutions by performing a two-pass wavelength calibration which retains for the second pass only the lines that were consistently identified during the first pass.

G.3.5 Resampling in Wavelength

Resampling with the Row-by-Row Solution

The command `REBIN/LONG` can be used to apply the row-by-row dispersion solutions. To each row of the rebinned spectrum, this command will apply the closest dispersion relation. If a one-dimensional dispersion relation was estimated,*e.g.* by averaging the rows of the arc spectrum before wavelength calibration, the command `REBIN/LONG` will apply this unique dispersion relation all over the spectrum.

Midas...> `REBIN/LONG in out`

rebins each row of the input frame **in** separately, using the dispersion coefficients stored in table **COERBR**. The limits and step in the wavelength space are estimated automatically by the command `CALIBRATE/LONG` and can be checked with:

Midas...> `SHOW/LONG r`

Resampling with the Bivariate Solution

The command:

Midas...> `RECTIFY/LONG` nameI name0 [nrep] [deconvol]

geometrically rectifies a 2-D spectrum and rebin it to constant step in wavelength, using the dispersion coefficients stored in $\boxed{\text{KEYLONGD}}$.

Not Resampling the Data

The command `APPLY/DISPERSION` generates a result table containing the original flux counts of each pixel of the spectrum. This table can be plotted by `PLOT/SPECTRUM` as in the sequence:

Midas...> `APPLY/DISP ccd0020 sp20 @210`
Midas...> `PLOT/SPEC sp20`

which generates a table **sp20.tbl** containing the pixel values of the row 210 of the image **ccd0020.bdf** as well as the central wavelength of each pixel. This command is provided as a convenience for some applications when data resampling is not desirable. However the resulting spectral table cannot be processed further by the current package.

G.3.6 Estimating the Sky Background

The sky is measured in two regions usually located below and above the object spectrum and which pixel boundary limits are stored in the keywords LOWSKY and UPPSKY. The algorithm removes the cosmic rays from the sky spectrum before interpolation, using the CCD detector parameters RON for the read-out-noise, GAIN for the gain, SIGMA as threshold of the kappa-sigma clipping and 2*RADIUS+1 as size of the rejection window. The sky is fitted along the columns of the spectrum by a polynomial of degree SKYORD. Unique or independent spatial profiles are computed depending of the value of the parameter SKYMOD. Sky estimation parameters are displayed together with extraction parameters by:

Midas...> `SHOW/LONG e.`

The commands:

Midas...> `SET/LONG LOWSKY=50,180 UPPSKY=220,350`
Midas...> `SKYFIT/LONG ccd0056 sky56`

generate an image **sky56.bdf** of which size is identical to the input image **ccd0056.bdf** and corresponding to the interpolated sky background. Sky subtraction is performed with:

Midas...> `COMPUTE/IMAGE corr56 = ccd0056 - sky56`

G.3.7 Extracting the Spectrum

The limits of the object are defined in the keyword OBJECT. A simple row average is performed by:

Midas...> `SET/LONG OBJECT=189,196`
Midas...> `EXTRACT/AVERAGE corr56 ext56`

which is equivalent to the general MIDAS command:

Midas...> `AVERAGE/ROW ex56 = corr56 @189,@196`

An optimal extraction algorithm is also available, requiring the knowledge of the CCD detector parameters, a preliminary definition of the sky background, as well as the values of a polynomial order ORDER and a number of iterations NITER. The command:

Midas...> `EXTRACT/LONG ccd0056 ext56 sky56`

applies the Horne's optimal extraction algorithm to the image **ccd0056.bdf** and performs the sky subtraction to generate the output image **ext56.bdf**.

G.3.8 Flux Calibration

Parameters for instrumental response estimation and flux calibration are displayed by the command:

Midas...> `SHOW/LONG f`

The flux calibration consists of correcting for atmospheric extinction using a table $\boxed{\text{EXTAB}}$ and of comparing the one-dimensional reduced spectrum of a standard star $\boxed{\text{STD}}$ to a reference flux table $\boxed{\text{FLUXTAB}}$. The sequence:

Midas...> `SET/LONG FLUXTAB=fei110 EXTAB=atmoexan`
Midas...> `PLOT/FLUX`
Midas...> `EXTINCTION/LONG std23 ex23`

plots the flux table and corects the one-dimensional reduced spectrum **std23** for the atmospheric extinction. The instrumental response can be estimated by filtering with the command:

Midas...> `RESPONSE/FILTER ex23`
Midas...> `PLOT/RESPONSE`

or by polynomial or spline interpolation using the sequence:

Midas...> `INTEGRATE/LONG ex23`
Midas...> `RESPONSE/LONG fit=SPLINE`
Midas...> `PLOT/RESPONSE`

The obtained response is stored in the image **response.bdf** and can be applied to any reduced, extinction corrected spectrum using the command:

Midas...> `CALIBRATE/FLUX exspec spectrum`

G.3.9 End of the Session

At the end of the reduction session (or at any time during the session) all option and numerical parameters can be saved for later use, with:

Midas...> `SAVE/LONG` **name**

where **name** is the name of the output table to be created which comprises:

1. the table **line.tbl**,

2. the values of all session keywords saved as descriptors of **line.tbl**,
3. the table **COE"name".tbl**

This can be checked with the following commands:

Midas...> `READ/TABLE` name

should list the contents of the table **line.tbl** and:

Midas...> `READ/DESCRIPTOR` name.tbl *

should, among others, display the parameter values. To restore these values at the beginning of a session (or at any time during an ongoing session) type:

Midas...> `INIT/LONG` name

G.4 XLong

This chapter describes the use of the Graphical User Interface XLong. The interface XLong generates and sends MIDAS commands to the monitor. Therefore, all operations performed with the interface can also be performed manually by typing the commands on-line or writing a procedure. The interface is however a convenient additional layer of the software providing on-line help, visibility over the parameter values and options and avoiding syntax problems. These functions are particularly useful for parameter intensive procedures such as the batch reduction.

Note

Using XLong requires that the described MOTIF GUIs have been installed on your system. Also this chapter must be considered as an operating manual of the interface and assumes a general understanding of the Long context.

In this Section the following notations have been used:

Commands or keywords are indicated as:
`SET/LONG UPPSKY=190,220`
Push-buttons and Menus are surrounded by a box:
`Identify...`
and Labels or names are written in bold face:
Calibration Frame:

G.4.1 Graphical User Interfaces

The Main window of the interface XLong, presented in Figure G.4.1 is created by the command:

Midas...> `CREATE/GUI LONG`

and includes the following elements:

a) A menu bar

b) A parameter area

c) A short help

c) A set of push-buttons

Initializing Keywords

The parameter area is used to set and display the various parameters used by the context. These parameters may be either defined by the user or defaulted by the context. User defined parameters are set or changed by simply moving the mouse cursor to the relevant field and typing in the value. The corresponding `SET/LONG` command is sent to MIDAS as soon as the mouse is moved out of the field. The command is echoed in the MIDAS monitor window.

Short Help

The short help is a small window located immediately below the parameter area, and is updated as the cursor moves on the different components of the interface to provide short information on the parameters and the possible actions.

Sending Commands

The buttons are located immediately below the short help. In colour terminals, the button labels have different colours which group them by functions, usually distinguishing between processing and verification commands.

In order to "push" one of these buttons, you must position the mouse on top and "click" with the left button. In what follows, this operation will be called "to click" and will refer to the left-hand button unless indicated otherwise. Therefore, to exit XLong you must click the `Quit` menu of the menu bar and select the `Bye` option.

On line HELP facility

The interface incorporates an on line HELP facility. A comprehensive description of the functions of each button in the interface is obtained by clicking the right-hand mouse button over an interface button. A window appears with the description. The window remains on the screen when the mouse button is released and can be updated with a new message. Try for example to click with the right-hand mouse button on `Identify...` in the Main window, then with the same mouse button on `Calibrate...`.

Entering File Names

Input fields expecting file names can usually pop up a file list selction by clicking the right-hand mouse button in the text field. For example, clicking with the right-hand mouse button in the text field located in front of **Line Catalog:** in the Main Window pops up a selection list of all ***.tbl** files in your directory. A given file can be selected by clicking on the file name.

Dialog Windows

In addition, XLong uses dialog windows to input specialized parameters necessary for the different reduction steps. These windows contain text fields, option menus and radio buttons. Values are given by moving the mouse inside the parameter fields or selecting an option by clicking on the relevant button. Push-buttons yielding to a dialog window are indicated by three dots in the label. For example, clicking on `Rebin...` pops up the Rebinning window. This window can be closed by clicking its `Cancel` button.

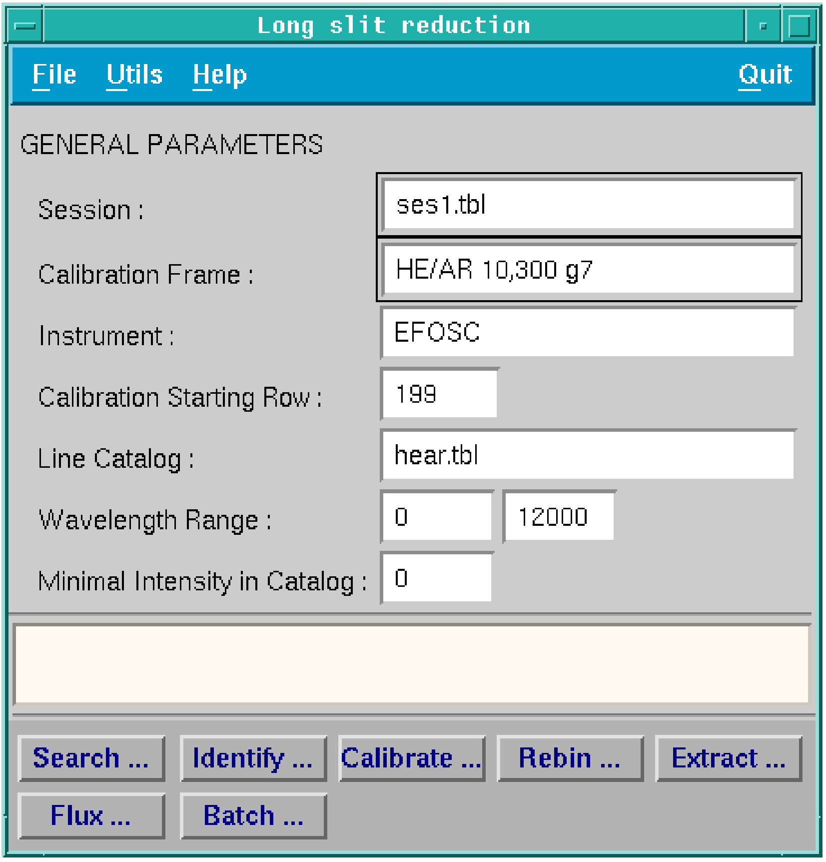

Figure G.1: Main window of the GUI XLong

G.4.2 Getting Started

Saving and Loading Session Parameters

A number of parameters appear in the parameter area which correspond to the status of the package when the interface is created. These parameters can be changed by typing the new values in the field, or reading them from a previously saved session table. The option `Open` in the menu `File` may be used to read a session file, as indicated in Figure reffig:xsave. The name of the currently loaded session is indicated in the first text field **Session** at the top of the parameter area in the Main window. Selecting the option `Save` in the menu `File` will save the session keywords in this table. This option can be used to save intermediate reductions, whereas the option `Save As...` allows to specify a new table name.

Detecting lines in the comparison (arc) spectrum

The Search window is popped up by clicking on the button `Search...` in the Main window. The different parameters controlling the SEARCH/LONG command can be updated in this dialog window. The file processed is selecting by clicking on `Search` in the Search window, which pops up a file selection list. When a file is selected, the SEARCH/LONG command is sent to the MIDAS monitor. The results verification command PLOT/SEARCH is activated by clicking on the `Plot` button in the Search window.

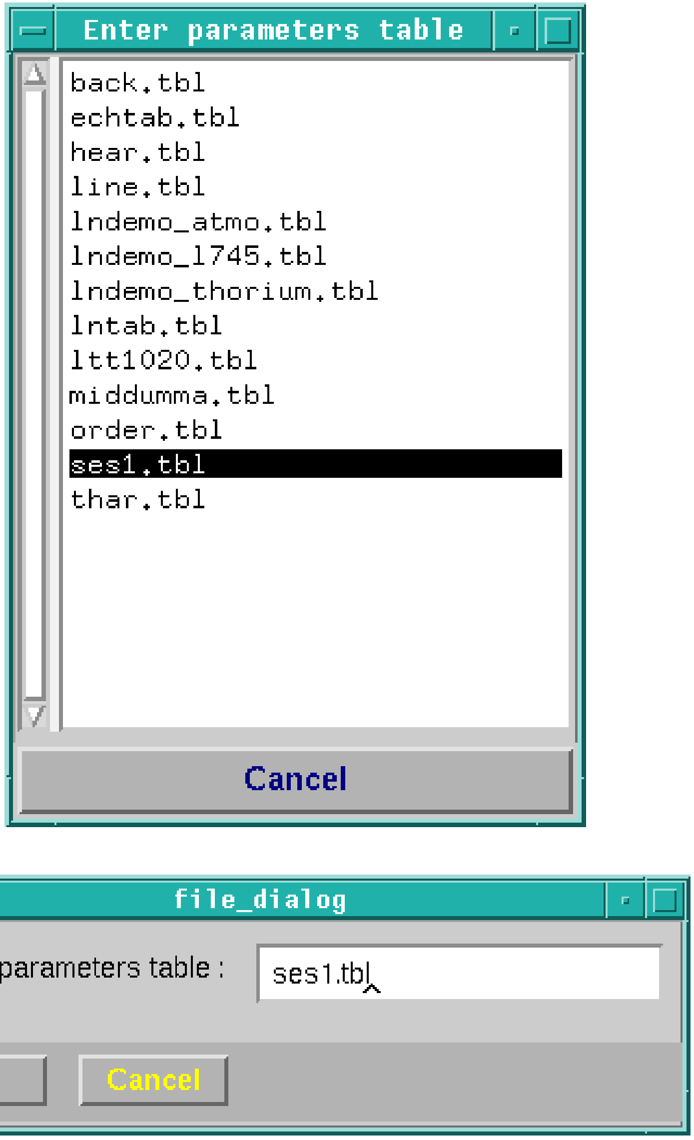

Figure G.2: Panels for Open and Save As... options of the menu File

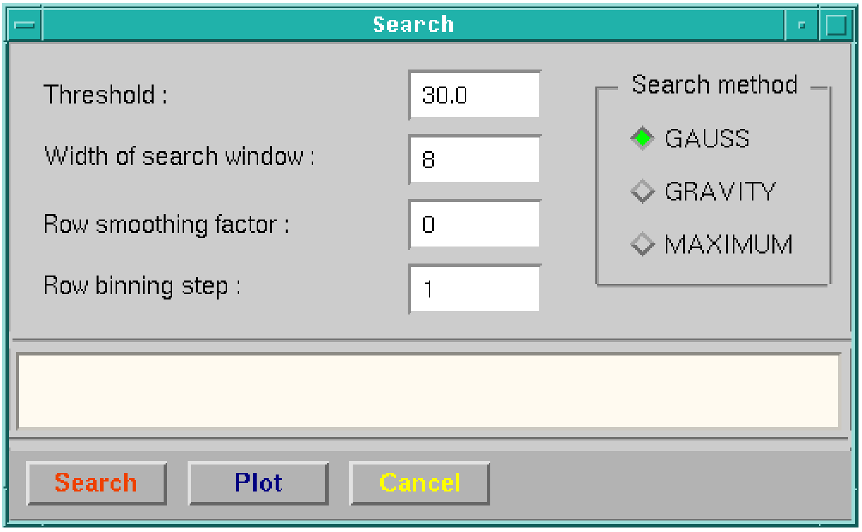

Figure G.3: Search Window

Identify the lines in one of the rows of the spectrum

Different parameters are used, indicated in the following fields of the Main window:

- the name of the table with the wavelengths of the comparison lines, indicated in the field **Line Catalog**.
- the range in wavelengths to be considered, indicated in **Wavelength Range**
- the **Minimal Intensity in Catalog** is used to perform a selection of the brightest lines of the catalog
- and the **Calibration Starting Row** will be used for the lines identifications. If this parameter is set to its default value 0, the central row will be considered.

After setting these parameters, click the `Identify...` button. to pop up the Identify window. The menu `Utils` in the menu bar provides graphic related commands. The identification starts by clicking on `begin`, which plots one row of the spectrum in the graphic window, indicates the position of the detected arc lines with blue vertical traces and gives a cursor.

When a detected line is clicked on, the interface expects the corresponding wavelength to be clicked in the wavelength list. The identified line turns green and an identification message appears in the MIDAS monitor. It is then possible to identify a new line. When a sufficient number of identifications has been performed, clicking with the middle mouse button in the graphic window desactivates the cursor. The full spectrum is plotted by default. Clicking again on `begin` deletes all identification, while more identification can be added by clicking on `continue`. Identifications can also be removed one by one with the button `delete`.

The x-axis and y-axis limits of the plot can be modified interactively with the menu `Frame`. The Begin button sets the x-axis to default values in order to start each time from a predictible configuration. The Continue button however will take the defined limits. The procedure to ideintify lines in zoomed areas of the spectrum is therefore:

- start with Begin
- if you identified no line, simply exit with the middle button of the mouse
- select new limits in x and y.
- make more identifications with Continue

Wavelength calibration

After having identified a few lines with the Identify window, click the `Calibrate` button in the Main window. The Wavelength Calibration window appears. The associated parameters can be set and a preliminary wavelength calibration is performed on the central row `YSTART` by clicking on `Calibrate`. The button `Edit` allows to remove calibration lines from the dispersion solution interactively, while `Calibrate all` estimates the dispersion relation for the complete spectrum. The button `Calibrate twice` performs a two-pass calibration.

Results can be checked with the buttons located at the bottom of the Wavelength Calibration window. `Dispersion` plots the dispersion relation, `Residuals` plots the residuals, `Spectrum` plots the lines identified during the calibration process. These three functions are by default performed on the central row of the spectrum. The button `Line Shape` plots the residuals as a function of the Y-position for a line selected in a wavelength list.

Rebin to wavelength scale

After completing the wavelength calibration, the `Rebin...` button in the Mainwindow allows to rebin spectra to wavelength. The Rebinning window offers three methods: linear,

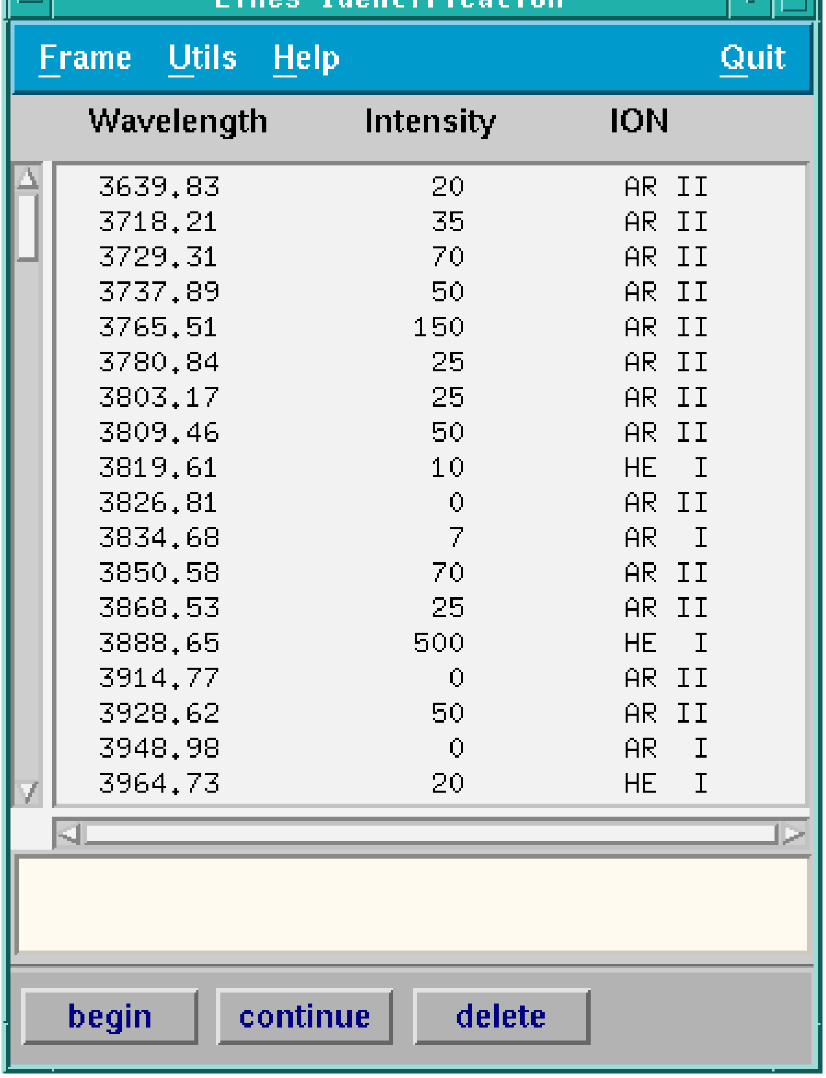

Figure G.4: Lines Identification Window

quadratic, or spline transformations. The first two provide flux conservation. The spline option does not incorporate a detailed conservation of flux. The button `Rebin rbr` activates the row-by-row solution corresponding to the MIDAS command `REBIN/RBR`. The button `Rebin 2-D` resamples by a bivariate polynomial solution and activates the command `RECTIFY/LONG`. The 2-D solution has been generated by the wavelength calibration process only if the option "Compute 2-D solution" in the Wavelength Calibration window has been selected before calibration. The button `Rebin table` allows to generate a table as output format and therefore avoid to resample the data. This table can be plotted with the button `Plot table`.

Clicking on one of the Rebin buttons pops up a list of **.bdf** files in the directory. Click the name of the selected file. A small prompt window appears requesting the name of the output file. The default is the input file name with the suffix `_reb`. The button `Plot table` uses the name of the output table generated by `Rebin table`.

The rebinning processes use the **Starting**, **Final** wavelength and **Step** parameters defined in the Rebinning window. If these parameters are not given, default values derived from the wavelength calibration coefficients are used. The progress in the rebinning process is reported in the MIDAS window.

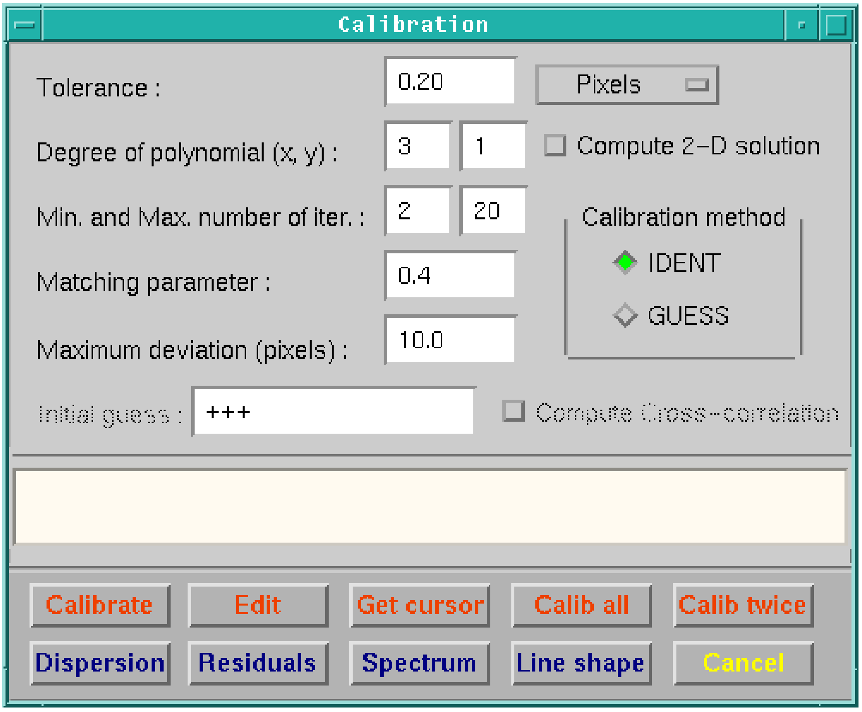

Figure G.5: Wavelength Calibration window

Extract

The Spectrum Extraction window is activated by clicking on the `Extract...` button in the Main window and contains options for fitting the sky and extracting spectra. The various options and parameters of this window are described in Sections G.3.6 and G.3.7 of this Appendix and in Chapter 6, Vol. B of the MIDAS User Guide.

The window provides two buttons `Get sky` and `Get object` for an interactive definition of the limits of the sky and the object. Clicking one of the these buttons activates a cursor in the display window. Two positions, corresponding sequentially to the lower and upper limits, are expected for the object. Four values are expected for the sky, defining two zones usually located below and above the spectrum. Positions must be clicked from the bottom to the top of the frame. When the expected number of positions has been selected, a `SET/LONG` command initializing the object or sky limit keywords (`OBJECT`, `LOWSKY`, `UPPSKY`) is sent to MIDAS.

The sky can be fitted on a spectrum by clicking on `Fit sky`. A file selection list pops up for selection of the input file. The name of the resulting sky file is defined in the text field **Sky (image or constant):** which includes the default value **sky**.

Spectrum extraction can be performed by simple row average or using an optimal extraction algorithm with the two buttons `Ext average` and `Ext weight`. Clicking on

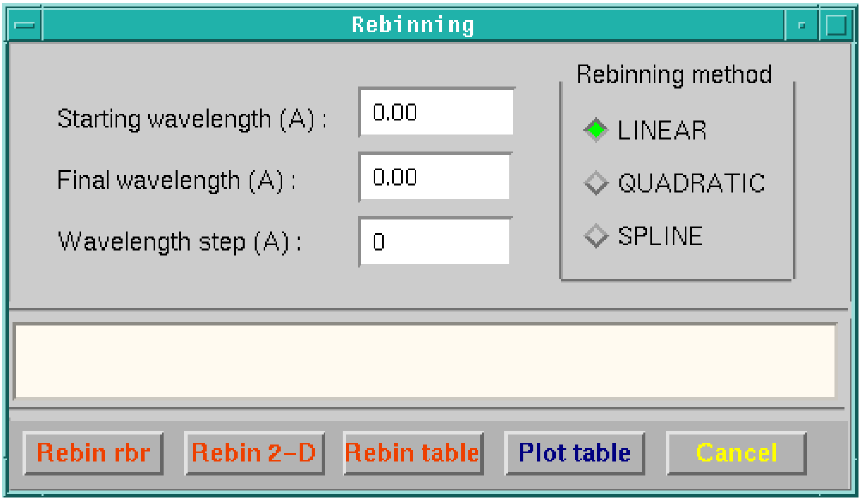

Figure G.6: Resampling Window

one of those buttons pops up a list of **.bdf** files in the directory. Click the name of the selected file. A small prompt window appears requesting the name of the output file. The default is the input file name with the suffix `_obj`.

Flux Calibration

The Flux Calibration window contains the options required for the atmospheric extinction and flux calibration. Click the `Flux...` button in the Main window to pop up the Flux Calibration window.

The `Extinct` button corrects the spectra for extinction and requires that the field **Extinction table:** contains a valid extinction table name. After clicking `Extinct` a selection list for **.bdf** files pops up. Click the file you want to correct. A small prompt window asks for the airmass. If the airmass appears in the file header, that value is used as default. The output is stored by default in a file with the original name plus the suffix `_ext`. Airmass and output file name can be modified before clicking on `OK` which activates the command `EXTINCTION/LONG`.

The `Integr` button allows the response table to be generated. The field **Flux table:** must be updated with the name of the standard star flux table. This table can be plotted by clicking on the button `Plot Flux`. After clicking `Integr` a file selection window appears requesting the name of the standard star image, which must be a one dimensional reduced, extinction corrected spectrum. Click the name. The name of the resulting intermediate response table is stored in the MIDAS keyword `RESPTAB` and by default set to **resp.tbl**. Values of this table can be interactively edited by clicking on `Edit`.

The response table must be interpolated to generate the final response curve, which name is provided in the field **Response curve:**. The Section **FITTING PARAMETERS** allows the different values and options to be selected (See Section G.3.8 and Chapter 6 Vol. B). The button `Fitting space` radio button allows the calibration curves to be plotted in two different ways. The first option `ratio/wave` is the standard plane used by MIDAS. The second option `magnitude/wave` plots Δm versus λ and generally has the advantage to require lower order curves to fit the response. The `Fitting type` button allows fitting the curves with either polynomials or splines.

Clicking on `Fit` activates the MIDAS command `RESPONSE/LONG`. The response curve can also be generated by filtering with the button `Filter`. The response curve can be plotted by clicking on the button `Plot resp`.

Reduced, extinction corrected spectra can be corrected for the instrumental response with the button `Correct`. Clicking on this button pops up a file selection list. Click the name of the spectrum. A small prompt window appears requesting the name of the output file. The default is the input file name with the suffix `_cor`. Clicking on `OK` sends a command `CALIBRATE/FLUX` to the MIDAS monitor.

G.4.3 Performing Batch Reduction

The Batch Reduction window allows catalogs of observations to be processed and a data reduction scheme to be defined dynamically. The different steps like Bias, Dark, Flat-Field correction, *etc.* are optional and must be set by clicking on the option buttons located on the left side of the window. The option button turns green when selected.

For each selected reduction step, the corresponding parameters must be provided, as indicated in the short help section. These parameters can be saved in a parameters file and be restored later, using the `File` pulldown menu.

The text fields associated to images, tables or catalogs allow a selection list to be displayed using the right-hand mouse button. Clicking over the desired name updates the corresponding field.

The input files can be given in two ways:

- Catalog name: by providing a catalog name in the field **Prefix/Catalog**
- Image numbers: by indicating the prefix of the images in the field **Prefix/Catalog** and providing the image numbers in the field **Numbers**, using dashes and/or commas. as in:
  Prefix/Catalog : red
  Numbers: 3-5,8
  In this example, the images **red0003, red0004, red0005, red0008** will be processed.

The dialog form accessed by pressing "Airmass ..." allows to modify the airmass values of the input images. If any of the input images do not have the corresponding airmass descriptor, the airmass dialog form is displayed automatically. The `Airmass...` button allows the modification of the airmass values of the input images. The `Apply` button executes the batch reduction.

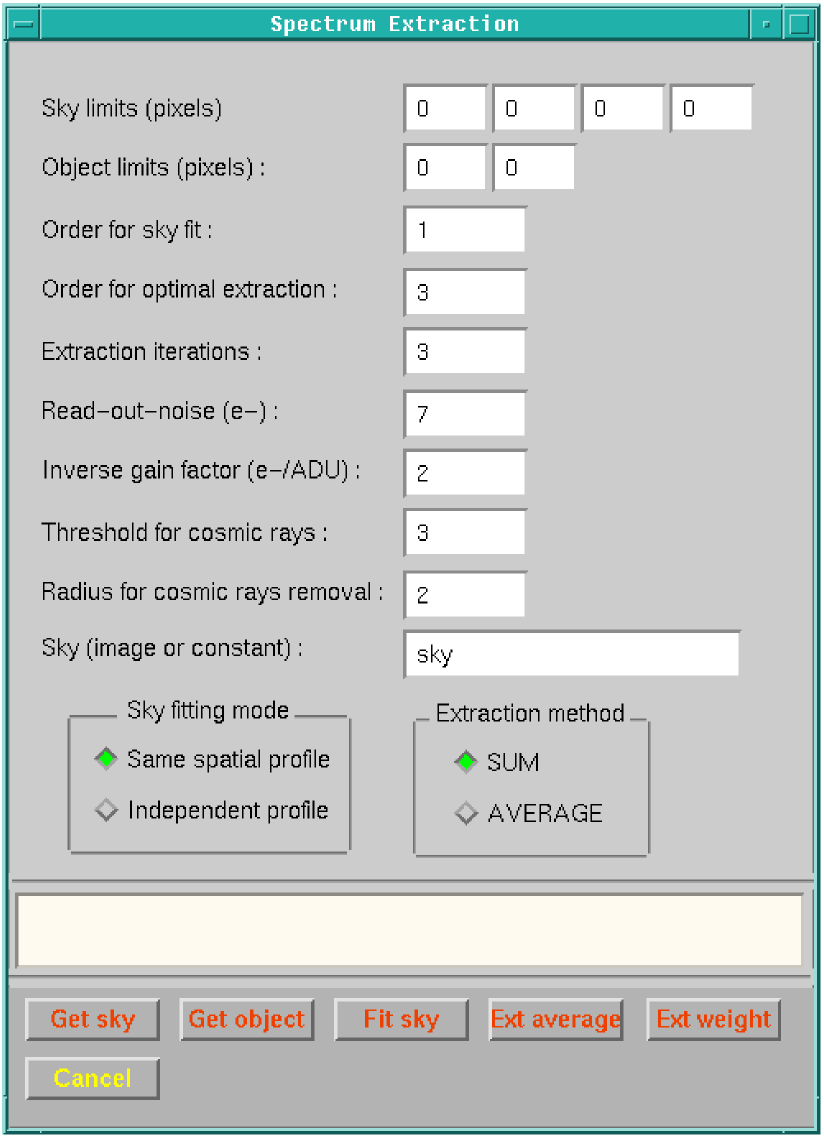

Figure G.7: Spectrum Extraction window

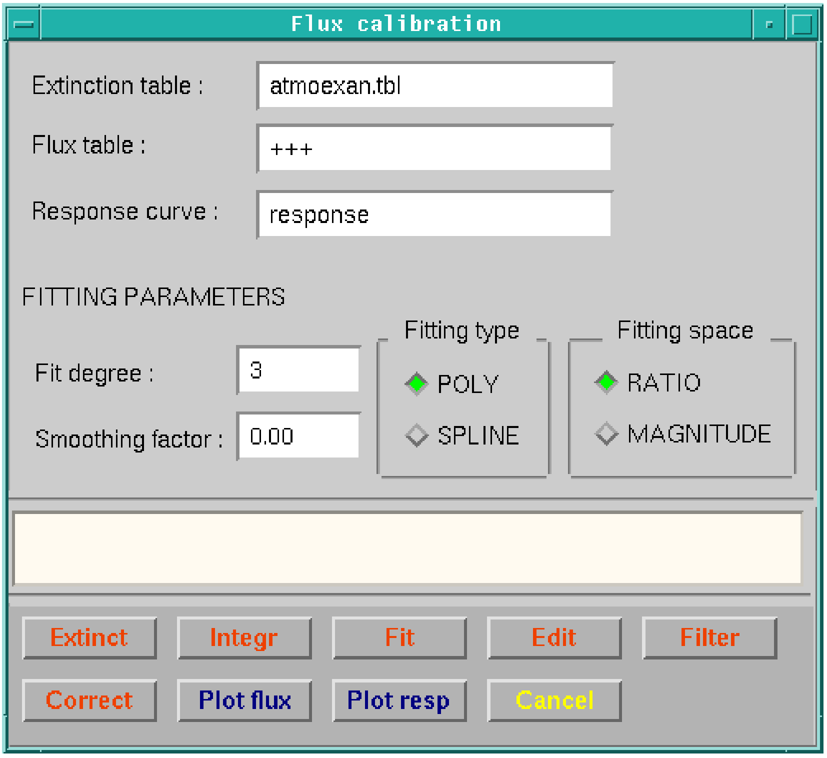

Figure G.8: Flux Calibration window

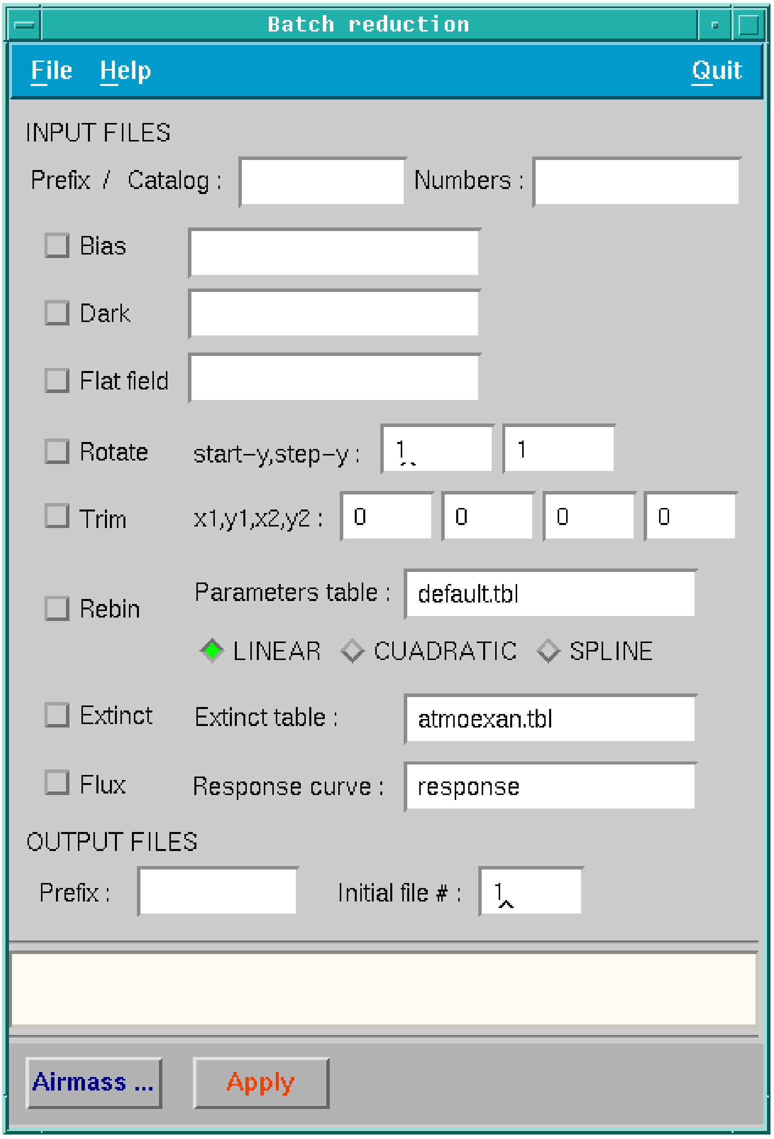

Figure G.9: Batch Reduction window

Appendix H

Optopus

H.1 Introduction

The purpose of this chapter is to describe the Optopus package, the set of programs, now implemented in MIDAS, which helps the Optopus observers to prepare their observations in a fast and easy way.

The Optopus facility is designed to enable conventional use of the Boller and Chivens spectrograph to be extended to perform spectroscopy of multiple or large extended objects simultaneously (see G. Lund, 1986,*OPTOPUS, ESO Operating Manual No. 6*, for further details). At present, Optopus is available for use at the 3.6m telescope only.

The fibre component of Optopus consists of 54 separately cabled optical fibres which enable light to be guided from freely distributed points in the focal plane at the entrance slit of the spectograph. The fibre ends are precisely located at the telescope focal plane by means of accurately drilled templates, known as "starplates".

The Optopus starplates (one for each observed field) are prepared in an automatic process in the ESO workshop on La Silla, after the observer has produced a drilling instruction file from his/her coordinate dat for each field. The Optopus package in MIDAS enables Optopus observers to create the file with the instuctions for the computer-controlled milling machine. It has been designed to be as "user friendly" as possible. All the commands request a minimum amount of input from the user; in addition they verify use of correct parameters.

Please do not forget that the instruction file has to be transferred to La Silla. Also, note that the milling machine on the mountain can only produce 2 to 3 starplates per day. If in doubt, always check with the Visiting Astronomers Section at ESO Headquarters in Garching at least three months prior to the observations.

H.2 Using the Optopus Package

H.2.1 Starting up

The commands in the Optopus package have to initialised by setting the Optopus context with the MIDAS command `SET/CONTEXT OPTOPUS`. All commands in this package have the qualifier OPTOPUS. Since the majority of them need quite a number of input parameters, and where their correct order and meaning is not always easy to remember, there are are two ways to enter command parameters. Besides the usual way of writing for each command:

```
command/qualifier p1 p2 p3 ...
```

it is also possible to define them in an explicit form:

```
SET/OPTOPUS param=value [param=value],
```

where `param` is the parameter name and `value` is the assigned value. Every parameter set in this ways can be defaulted on the command lines. Only the input and output filenames are required unless the general default names are used (see documentation of the individual commands). However, before executing a command it is recommended to check the session parameters by listing them with the command `SHOW/OPTOPUS`. This will produce an output table like in Table H.1. For a complete overview see Section H.3.2.

Input file:	mytab1.tbl
Output file:	mytab2.tbl
Plate label:	SA94
Plate center:	R.A.: 02h 43m 30.000s DEC.: -00d 15' 50.00"
Equinoxes:	Old : 1950.00000 New: 1991.76776
Date:	Year: 1991.00000 Month: 10. Day: 9. Epoch: 1991.76776
Exposure time:	120.0m
Wavelength range:	from: 4000. Ångstrom to: 8000. Ångstrom
Optimal sid. time slot:	from: 1h 0m to: 3h 0m
Optimal sid. time:	0.00h
ACFLAG:	N
PFLAG:	Y
ASTFLAG:	Y
EWFLAG:	N

Table H.1: Parameters listed by SHOW/OPTOPUS

The assigned values are maintained until the user gives the MIDAS command `CLEAR/CONTEXT` or decides to leave the MIDAS session. However, is is possible to save them with the command `SAVE/OPTOPUS table`, where `table` is the name of any table chosen by the user. `SAVE/OPTOPUS` saves the relevant session parameters by copying them to descriptors of `table`. It is advisable to use this command, not only when you want to interrupt a session and restart it later, but again during the session to protect yourself against system crashes or accidental logouts of MIDAS. When re-entering Optopus context, all parameters are re-initialised to the default values but they can be re-set to the values of a previously saved session with `RESTORE/OPTOPUS table`, where `table` is of course the name of the table that contains the saved setting.

Since almost all commands in the package work, both in input and in output, on MIDAS tables, another important task of the user at the start of an Optopus session will be to create a MIDAS table from the ASCII file where the data about the objects to be observed are being kept. The newly created table will have to contain, amongst others, an `:IDENT` and a `:TYPE` column, where `:TYPE` contains “B” or “S”, respectively for “big” and “small” guidestars, and “O” for scientific object. As the format of this table is fixed and crucial for all the following operations, there is a dedicated command for this purpose:

```
CREATE/OPTOPUS inp_file [out_tab] [fmt_file] [old_equinox]
```

A standard `fmt_file` can be seen in Table H.2.

```
define/field  1   16  c  a16   :ident
define/field  18  19  d  f2.0  :ahr    "hours"
define/field  21  22  d  f2.0  :amin   "min"
define/field  24  29  d  f6.3  :asec   "sec"
define/field  33  33  c  a1    :sign
define/field  34  35  d  f2.0  :ddeg   "degrees"
define/field  37  38  d  f2.0  :dmin   "arcmin"
define/field  40  44  d  f5.2  :dsec   "arcsec"
define/field  48  48  c  a1    :type
exit
```

Table H.2: Example format file

A copy of this format file will be put in the working directis available in the file:

- `$MIDASHOME/$MIDVERS/stdred/optopus/incl/opto.fmt` (UNIX)
- `MID_DISK:[MIDASHOME.MIDVERS.STDRED.OPTOPUS.INCL]OPTO.FMT` (VAX/VMS)

so that it can be copied to the user's working directory and subsequently modified according to the positions and field widths in his/her ASCII file. You can copy the format file into your working directory with the `CREATE/OPTOPUS` command itself. To do so give the third parameter `fmt_file` in the command the value `copy`. However, a copy will not be made if a file with the name `opto.fmt` is already present.

In case the targets are already stored in a MIDAS table the user should check if the table columns have the correct labels. If required, modifications in the table can be made by using one or more commands for table manipulation.

The equinox of the data has to be stored in the descriptor `TABEQUI` of the MIDAS table. It is important to verify whether the equatorial coordinates have been precessed or not. In fact, next step in this “building–up” of the Optopus session is the command:

```
PRECESS/OPTOPUS [inp_tab] [new_equinox],
```

which corrects the right ascension and declination for precession to the date of the observation (this is the default, which can be changed by defining the parameter NEWEQ), and updates the value of the double precision descriptor TABEQUI. To limit the number of files created in an Optopus session, this command will not to create a new table, but will add two new columns :RA and :DEC in the table created by CREATE/OPTOPUS. The old columns :RA and :DEC are renamed to :OLDRA and :OLDDEC, respectively. Note, that in this table the equatorial coordinates are in decimal format.

H.2.2 The Optopus session

After the initial setting up of the Optopus session, the user is now ready for the "real thing", that is to use of the main commands of the Optopus package:

- HOLES/OPTOPUS,
- MODIFY/OPTOPUS and ZOOM/OPTOPUS,
- REFRACTION/OPTOPUS.

HOLES/OPTOPUS converts the RA and DEC coordinates in the MIDAS table created by CREATE/OPTOPUS (and precessed by PRECESS/OPTOPUS) into :X and :Y positions of the holes to be drilled on the Optopus starplate. It outputs the following information:

1. objects or guidestars falling outside the plate area;
2. objects or guidestars falling in the so called "forbidden area", that is the thicker part of the plate used to fix it to the spectrograph;
3. objects which are too close to a guidestar (big or small);
4. objects of only which are in competition because of their proximity.

The plate is needed. The user is offered two alternatives: either to enter pre-determined center coordinates (using SET/OPTOPUS CRA=value1 CDEC=value2, where value1 has the format HH,MM,SS.sss, and value2 the format +/-DD,AM,AS.ss, and SET/OPTOPUS ACFLAG=N, or to use the command to compute them automatically (SET/OPTOPUS ACFLAG=Y. The automatic determination of the center simply uses the arithmetic mean of the :RA and :DEC columns. The result is not always optimal. To choose the "best" center (that is the one which permits to keep the maximum number of objects inside the plate limits) from this guess, the user uses the MODIFY/OPTOPUS command. This command displays graphically the position of the holes on the starplate and, if required, permits modifications of the RA and DEC of the center using SET/OPTOPUS CRA=value1 CDEC=value2. The user should re-run HOLES/OPTOPUS followed by MODIFY/OPTOPUS to verify the improvements.

An important point is that both the center of the plate and the :RA and :DEC coordinates in the input table must be corrected to the same equinox. If you decide to input your own pre-calculated center coordinates, either precessed or not, you also have to remember to set the value of the parameter PFLAG accordingly. In case of automatic determination of the center, the center is calculated by averaging the `:RA` and `:DEC` columns in an already precessed table, so `PFLAG` is by default set to `N`.

The output table created by the command `HOLES/OPTOPUS` contains also a column called `:CHECK`. A letter `N` in this column identifies objects or guidestars with location problems of any kind (they will be indicated by a square, in the graphic output produced by `MODIFY/OPTOPUS` and `ZOOM/OPTOPUS`).

The task of `MODIFY/OPTOPUS` is very simple and twofold:

- to visualise the RA and DEC positions of the holes to be drilled in an Optopus starplate.
  Care is also taken to distinguish between different kinds of objects by using different graphic symbols and to permit the correct identification of every single object by overlaying the content of the `:IDENT` column of the input table.

- to enable the rejection of objects or guidestars falling in a (for any reason) "in–convenient" position. For this purpose, a cursor is activated in the graphic display. The user can click on the objects or guidestars he/she wants to be ignored the subsequent commands of the Optopus session.

  In case of very crowded fields, the limited physical dimensions of the outputs of some graphic devices can make it difficult to read the identification labels of the objects, and hence making the task of deleting the "right" objects a really tricky one. To avoid undesirable results, some auxiliary information is displayed whenever the user clicks on an object: `:RA`, `:DEC`, `:IDENT` and content of the `:CHECK` column. The user is prompted for substitution the "N" already present in this later column with a "D" or "d" (for delete).

  In case the wrong object has been selected, that is the :CHECK column is empty, it is sufficient to hit return to keep everything as it was. It may also happen that a "wrong" object is selected twice, that is the `:CHECK` column already contains a "D" or "d". In this case one has to type the same letter ("D" or "d") again, otherwise the object wlll not be rejected. Note, that in case of close pairs of objects, both of them are surrounded by the square symbol which means "candidate for deletion"; however, both squares will disappear after having deleted only one of the two object.

  Finally, if for any reason you decide you would rather keep one of the objects marked by a square, click on it and type anything but a "D" or a "d" when prompted. In order to see what the starplate looks like after this editing, the user first has to deactivate the graphics cursor (by hitting the spacebar or pressing the right button of the mouse of your workstation) and then rerun `HOLES/OPTOPUS` and `MODIFY/HOLES`. Reverting `HOLES/OPTOPUS` is not compulsory, since it would be enough just to rerun `MODIFY/HOLES` to have a new plot of the Optopus starplate. However, it is also useful to check to review outliers and close pairs repetetively.

  The command `HOLES/OPTOPUS` is reasonably fast, so we advise the users to frequently switch between `HOLES/OPTOPUS` and `MODIFY/OPTOPUS` and vice versa.

Some users like to start their Optopus session with an populated field of candidate source. They then proceed to eliminate objects until a suitable number is reached. However, care should be taken to avoid eliminating more objects than necessary in cases where several targets are closely grouped together. In fact, even if the minimum separation between adjacent pairs is large enough to pass all the overlap checks performed by `HOLES/OPTOPUS`, once at the telescope it may become problematic to physically introduce the fibres into extremely close holes. It then may happen that one is forced to a late rejection of more scientific targets than one would have liked. However, this might turn out less harmful than expected if one had careful enough to have some "backup" holes drilled in the starplate.

In case of very close groups of objects, the command `ZOOM/OPTOPUS` may also be helpful. If the resolution provided by `MODIFY/OPTOPUS` is not enough, this command permits to actually blow up a section of the Optopus starplate plotted on the graphic screen by `MODIFY/OPTOPUS`. The user only has to choose the center of the section she/he wants to be enlarged with the cursor. In most cases the default zoom factor of 5 is sufficient to resolve close groups or pairs. However, should this resolution not to be enough, the possibility exists to enter the command `ZOOM/OPTOPUS` again, with a new zoom factor, the center remaining unchanged.

When all inacceptable objects have been removed, it is time to use the command `REFRACTION/OPTOPUS` to correct the `X` and `Y` position of the holes on the starplate for the effect of atmospheric refraction. For a detailed description of the correction algorithm and an estimation of such effects in the particular case of La Silla, we refer to G. Lund, 1986, *OPTOPUS, ESO Operating Manual No. 6*, pag. 17-18. Here, we summarise that from coordinated of the plate center coordinates, the specified temporal observating window and the wavelength range of interest, `REFRACTION/OPTOPUS` determines:

- an optimal differential correction vector, scaled according to the coordinates of each object;
- an optimal chromatic correction vector for the guidestars.

Note, that the coordinates of the plate center must be the same as the ones already used with `HOLES/OPTOPUS`. It is not necessary to reset these since `REFRACTION/OPTOPUS` will get the (precessed) values from the keyword `PLATECEN` that have been saved by `HOLES/OPTOPUS`.

In general, the observer will try to observe his/her fields at the smallest possible overall hour angle (airmass). This optimalisation has to be made in advance. The window in sidereal time for each of the plates which will be observed during a single night can be easily computed knowing that for the date entered by `SET/OPTOPUS DATE=value`, the command `REFRACTION/OPTOPUS` outputs the sidereal times at the beginning and end of the night on La Silla. Not more than 4 (in summer) or 5 (in winter) Optopus starplates can be used in one night. So, just to run `REFRACTION/OPTOPUS` using the default value for the sidereal time slot (ignore any error messages you may get, as in this first run you are only interested in the first line of the output, which will be correct anyway) and divide the night into 4 or 5 exposures (allowing for some start-up time at the beginning, approx. 20 minutes). An example of the output of `REFRACTION/OPTOPUS` can be found in Table H.3.

Darkness will begin at ST:	20.37
and end at ST:	5.16
Sidereal time for observation:	21.00
Hour angle:	–27.49 degrees
Zenith distance:	24.84
Maximum refraction correction:	0.23 arcsec
Position angle of correction vectors:	–106.56 degrees
Chosen length for exposure:	60 minutes
Approx. optimal obs. slot (ST):	20h 30m to 21h 30m
Approx. optimal obs. slot (UT):	24h 6m to 25h 6m
Corresp. range of corr. vectors:	from –99 to –116 deg.
Wavelength range for optimisation:	3800 to 5500 Ångstroms
Optimal correction at wavelength:	4329 Ångstroms
Chromatic correction needed in X:	–46. microns
Chromatic correction needed in Y:	–14. microns

Table H.3: Output of REFRACTION/OPTOPUS command

The sidereal time for which the corrections are finally calculated can either be enforced by the user, by setting the parameters `ASTFLAG=N` and `OST=value`, or automatically determined by the command. In the latter case `ASTFLAG` must be set to `Y`.

`REFRACTION/OPTOPUS` produces an output table quite alike the one created by `HOLES/OPTOPUS`. The most obvious differences are that now the `:X` and `:Y` columns contain coordinates corrected for the atmospheric refraction effects, and the column `:NUMBER`, has been added. This new column will later be needed to identify the holes on the starplate by a sequential number.

Another important characteristic of the table produced by `REFRACTION/OPTOPUS` is that, being the final table generated in the Optopus session and the one which the observer will presumably bring along to the telescope, it contains all relevant output information (*e.g.* see Table H.3) in its descriptors. Besides, as already remarked, the user has the possibility to save all parameters used in the session as well, by using the command `SAVE/OPTOPUS tablename`.

H.2.3 Closing down

Now that all the calculations to produce accurate positions of the holes which will host the fibres have been executed, you can proceed to a part of the process which is undoubtedly much more trivial but nonetheless vital both for having the holes actually drilled on the plate and for the subsequent work at the telescope.

First of all, you can obtain a map of the final coordinates of your objects and guide stars by using the command:

```
PLOT/OPTOPUS [table] [label] [EW_flag],
```

where `[table]` is the table output by `REFRACTION/OPTOPUS`. Recall that before using this command it is necessary to assign the plotter you want to use with the command:

```
ASSIGN/PLOT device
```

and then, after successful completion of `PLOT/OPTOPUS`, the plot is sent to the assigned device.

The option provided by the `EW_flag` allows the user to choose between two possible orientations of the map. Normally, only the (default) unflipped version, corresponding to the appearance of the starplate from the machined side, is needed. If direct comparisons will be made with photographic plates where east is to the right, a flipped map should be requested. The default version of the map has north at the top and east to the left, and will be extremely helpful at the telescope, to ensure identical numbering of both objects and fibers. If the two are not uniquely correlated, the observer will not know which spectrum comes from which object!

Once on La Silla, holes will have to be labelled using the provided self–adhesive labels, in the same way as they are numbered on the maps produced by `PLOT/OPTOPUS` (i.e. according to the consecutive numbers assigned by the `REFRACTION/OPTOPUS` command). In order to avoid object misidentifications at a later stage, and for a cross identification between object identifiers and hole numbers, it is recommended to bring a printout of the table output created by the command `REFRACTION/OPTOPUS` to the telescope. Also definitely needed at the telescope is the printout of the descriptors of this fundamental table. An example can be found in Table H.3.

We remind users less familiar with MIDAS, that both the table and its descriptors may be printed out by first assigning the output ASCII file with:

```
ASSIGN/PRINTER FILE filename
```

then using the MIDAS commands `PRINT/TAB` and `PRINT/DESCRIPTOR` for the table and the descriptors respectively.

The last command to be used in an Optopus session generally is

```
DRILL/OPTOPUS in_table [out_file]
```

to transform the object coordinates taken from the output file of `REFRACTION/OPTOPUS` into a long sequence of machine instructions, correctly formatted for the programmable milling machine on La Silla. These will be output in `[out_file]` in ASCII format.

At the end of the operations with the Optopus package, all the instruction files created by `DRILL/OPTOPUS`, together with a copy of the plots produced by `PLOT/OPTOPUS`, have to handed over to the Garching staff. They will take care of the transfer to La Silla.

H.3 OPTOPUS Commands and Parameters

Below a brief summary of the Optopus commands and parameters is included for reference. The commands in Table H.4 are initialised by setting the Optopus context with the MIDAS command `SET/CONTEXT OPTOPUS`. The parameters can then be set via `SET/OPTOPUS par=value`.

H.3.1 Optopus commands

CREATE/OPTOPUS	inp_file [out_table] [fmt_file] [old_equinox]
HOLES/OPTOPUS	[in] [out] [HH,MM,SS.sss] [+/-DD,AM,AS.ss] - [ac_flag] [p_flag] [old_eq,new_eq]
MODIFY/OPTOPUS	[table]
PLOT/OPTOPUS	[table] [label] [EW_flag]
PRECESS/OPTOPUS	[table] [new_equinox]
REFRACTION/OPTOPUS	[inp_tab] [out_tab] [year,month,day] [exp] - [lambda1,lambda2] [start_st_sl,end_st_sl] - [opt_st] [ast_flag]
RESTORE/OPTOPUS	table
SAVE/OPTOPUS	table
SET/OPTOPUS	par=value
SHOW/OPTOPUS	
ZOOM/OPTOPUS	[table] [zooming_factor]

Table H.4: Optopus commands

H.3.2 Session parameters

Below follows a description of all parameters that can be set by the SET/OPTOPUS command, the commands which use these parameters and the default values. These parameters are stored in keywords and will be use as default values for subsequent OPTOPUS commands. If an OPTOPUS command the default value is not used the command will, in addition to actual OPTOPUS operation, also overwrite the setting of this parameter and hence will change the default value.

```
SET/OPTOPUS OLDEQ=1950.0         !  set the equinox
CREATE/OPTOPUS mydata mytable ?  2000.0  !  use another equinox
SHOW/OPTOPUS                 ! equinox value is now 2000.0
```

CREATE/OPTOPUS	
Parameter	**Description**
`OLDEQ`	Equinox of RA and DEC coordinates of objects and guidestars in input table. Format must be: YEAR.yyyyy. Default value is "1950.0".

HOLES/OPTOPUS	
Parameter	**Description**
`CRA`	Right ascension of the center of the Optopus plate. Input format must be: HH,MM,SS.sss. Default value is "00,00,00.000".
`CDEC`	Declination of the center of the Optopus plate. Input format must be: +/– DD,AM,AS.ss. Default value is "00,00,00.00".
`ACFLAG`	Character flag. Y or y (for yes) and N or n (for no) are the only values. If ACFLAG is set to Y (default) the automatic determination of the plate center enabled.
`PFLAG`	Character flag. Y or y (for yes) and N or n (for no) are the only values. If PFLAG is set to Y the automatic precession of the plate center enabled. Default is Y.
`OLDEQ`	Old equinox of the plate center (if not yet precessed). Input format must be: YEAR.yyyyy. Default value is "1950.0".
`NEWEQ`	New equinox of the plate center. Must be the same used to precess RA and DEC of objects and guidestars with the command PRECESS/OPTOPUS. Input format must be: YEAR.yyyyy. Default value is "2000.0".

PLOT/OPTOPUS	
Parameter	**Description**
`LABEL`	Character string used to identify the plot. Default value is "Optopus plate".
`EWFLAG`	Character string. Y or y (for yes) and N or n (for no) are the only values. If EWFLAG is set to Y the EAST–WEST flipping of the plots is enabled. Default value is "N".

PRECESSION/OPTOPUS	
Parameter	**Description**
`NEWEQ`	New equinox used to precess RA and DEC coordinates of objects and guide stars in input table. Must be the same as used to precess center coordinates with the command HOLES/OPTOPUS. Input format must be: YEAR.yyyyy. Default value is 2000.0.

Table H.5: Command parameters

REFRACTION/OPTOPUS	
Parameter	**Description**
`DATE`	Year, month and day of the observation. The permitted input formats are: • YEAR,MONTH(number),DAY or • YEAR.yyyyy,0,0 Default value is 1999.,12.,31.
`EXTIM`	Exposure time in minutes of Optopus plate. Default value is 0.0.
`WRANGE`	Wavelength range to optimize the corrections for atmospheric refraction. Input format must be: LAMBDA1,LAMBDA2 both in Ångstrom. Default value is: 3800,8000
`SITSLT`	Sidereal time interval during which a given Optopus plate will probably be used. Input format must be: ST1.ss,ST2.ss. Default value is: 00.00,00.00
`OST`	Optimal sidereal time for correction determinations, that is sidereal time for which the corrections for atmospheric refraction are calculated. Input format must be: ST.ss. Default value is: 00.00.
`ASTFLAG`	Character flag. Y or y (for yes) and N or n (for no) are the only values. ASTFLAG set to Y enables the automatic calculation of the optimal sidereal time for correction determinations. Default is Y.

Table H.6: Command parameters (cont.)

Appendix I

File Formats Required for Photometry

I.1 Introduction

Photometrists obviously need lists of standard and extinction stars, both to plan observing sessions efficiently and to reduce the observational data. It is not so obvious that planning and reducing photometric observations also require information about the telescope used.

For example, the telescope aperture, which affects both photon and scintillation noise, must be known, both to select good extinction stars, and to weight the observations properly according to stellar magnitudes. Furthermore, the dome or telescope enclosure determines the sky area from which the Moon may shine on the telescope mirror, raising the apparent "sky" brightness. This information is needed both for planning observations and reducing data. The telescope's coordinates are required to determine when the Moon is likely to influence observations, as well as in determining the airmass of a star as a function of time.

Similarly, information is needed about the instrument. One must avoid stars that are too bright for a given instrument on a particular telescope. Many instrumental details influence the methods that must be used in reducing photometric observations, because the data-reduction process must accurately model the instrument's performance. Because instrumental configurations tend to change from one observing run to the next, it is appropriate to separate the instrumental data from the more permanent data that refer to the telescope alone.

Some of these data are already available at the NTT, and will become available at other telescopes, as part of the ESO archiving system. Unfortunately, the archiving system is designed around individual image frames, which do not correspond to the natural elements of photometric data. Therefore, it is necessary to extract the required information from the archive, and re-package it in a form more suited to photometry. So far as possible, individual data formats will be similar to those laid out in the ESO Archive Data Interface Requirements [5]. For example, column labels for the table files described here will (when possible) match the FITS-header keywords used for the same information in archiving.

The data needed in photometry can be grouped into MIDAS table files, each of which contains a natural set of data that belong together. Each table will be described in detail in the following sections. However, here is an overview of the types of tables needed for photometry:

I.1.1 Stars

Tables giving the identifications and positions of both standard and program stars are obviously required. The positions are required for air-mass calculations. In addition, the standard-star tables must give the standard magnitudes and colors or other indices.

I.1.2 Observatory data

A table file (called by default esotel.tbl) should contain more or less permanent information about the telescopes at an observatory. This includes sizes, positions, and other stable information. Each observatory should have its own file; the one for ESO telescopes will be called esotel.tbl.

I.1.3 Telescope obstruction data

Nearly every telescope seems to have some parts of the sky that are inaccessible. Trees, mountains, and even other telescopes obscure some areas above the horizon. Furthermore, many telescopes have peculiar mountings or other mechanical limitations on where they may look. In a surprising number of cases, the inaccessible regions extend into the parts of the sky one would normally use to measure extinction stars. Such restrictions should be placed in a separate "obstruction" table for each telescope.

I.1.4 Instrumental data

Instrumentation tends to change from one run to the next. Filters get broken or lost, or deteriorate and are replaced. Detectors change with age, may be destroyed by careless users, or just die for unknown reasons. Instruments suffer "improvements" that change their characteristics.

Only data taken with a fixed configuration can be reduced together. Usually, this will include all of the data taken during a run; occasionally, an equipment change is necessary during a run. Data describing a fixed instrumental configuration can naturally be grouped together in a table describing each particular run, or part of a run.

I.1.5 Observational data

Finally, we have the observations themselves. Here, the natural element is a single stellar (or sky, or dark) intensity measurement; and the natural set of these elements to put in a file is the whole group of measurements made on a single night.

I.2 Star tables

An important ingredient of both planning and data reduction is a set of standard stars. These can serve double duty as extinction stars, and are used for this by the planning program. They are obviously essential in data reduction.

If more than one table is used, the user must be careful *not* to intermix data that are not on the same system. For example, the Cousins E-region standards are clearly not on the same "UBV" system as the Landolt equatorial standards. Likewise, several distinct "RI" systems are in use.

Sometimes it is useful to include "catalog" stars in an observing program. These are stars that have been observed in (supposedly) the same system as the standard stars, but are of lower quality and are not suitable for use as standards. However, they are not obviously variable, and may be useful both as extinction stars and for checking transformations.

In addition, we need tables of program stars. These usually will contain at least a magnitude, which is used to confirm identifications but is not used in data reduction.

Different types of star (standard, catalog, and program) should be put in separate table files. You can have several tables of each type, but you should not try to mix different types in the same file.

These star tables all **require** the following data (*above* the line in Table I.1):

I.2.1 Required stellar data

Object name

The reference column of a star table contains the primary identification for the star. The column label, OBJECT, follows the archiving standards. As in the archive, this is a string of up to 32 characters. Shorter strings should be left-justified.

Standard IAU names should be used. Proper names are acceptable for bright stars, as are Bayer and Lacaille letters and Flamsteed numbers used with the constellation abbreviation. HR numbers can be used for bright stars. Telescopic stars should be designated by HD numbers. Stars lacking HD numbers should be named by BD or other DM number. Still fainter stars are identified by HST Guide Star Catalog names, if available. Nicknames like "Przybylski's star" should go in the COMMENT column (section I.2.2).

In clusters and around variable stars, one often finds field standards or reference stars denoted by letters of the alphabet. Some charts have used both capital and lower-case letters, so it is necessary to be case-sensitive in names. In such crowded areas, it is common practice to use one (or a few) common reference position(s) to measure sky, for the whole group of stars. In this case, the sky position(s) should also be recorded in the star-catalog table file; the only requirement is that the string in the OBJECT column begin with the word `SKY` (see discussion in section I.6, "Observational data".

Very often the observer will use a shortened form or abbreviation at the telescope. However, full names should be used in the star tables, to avoid ambiguity. The correspondence between full and abbreviated names will be resolved by programs only with the interactive consent of the user.

Right Ascension

Right Ascensions are tabulated in decimal degrees in the archive. This may be used with the column label RA, which conforms to the archive convention [5]. While this is a convenient machine-readable form, it is not very user-friendly, as most astronomical catalogs use sexagesimal time units.

Fortunately, the MIDAS table commands will accept sexagesimal input (see the on-line HELP for `CREATE/COLUMN` and `NAME/COLUMN`), in almost any human-readable form. By using the R11.6 format specification for the Right Ascension columns in the *.fmt file, existing ASCII tables can easily be converted to MIDAS table format. This *stores* the column as an angle in degrees, though it is *displayed* as hours, minutes, and seconds by the `READ/TABLE` command.

CAUTION: your ASCII file **must** have the `.dat` extension, or it will not be read correctly!

Declination

As for Right Ascension, Declinations are tabulated in decimal degrees in the archive. The column label DEC conforms to the archive convention. Here again, any sensible human-readable form may be used instead, if the ASCII data are read in with an s12.6 format in the *.fmt file. The result is stored as decimal degrees (as a Real*4 number), but displayed as degrees, minutes, and seconds.

Equinox

Because precession is a large effect, we must know the equinox to which these coordinates refer. Although this is usually a constant for a whole table, and so might be stored in a descriptor, observers may want to use lists of program stars compiled from heterogeneous sources. Therefore, the year of the equinox must be stored in a separate column of Real*4 values, with the label EQUINOX. This column is easily created for a star catalog with a single equinox date using the `COMPUTE/TABLE` *tablename* `:EQUINOX =` *year* command. A format of F7.1 is satisfactory.

I.2.2 Optional stellar data

In addition to the essential stellar data given in the previous subsection, it can be very useful to include supplemental information. For example, a program devoted to nearby dwarfs or low-luminosity stars would need proper motions to obtain accurate airmasses for some program stars. Spectral types are often adjoined to photometric catalog data. In many cases, rough photometric information that can aid in identification is available, such as a photographic magnitude from the HD. Finally, comments can be useful. These fields may be included in the table if the user wants them. They are described *below* the line in Table I.1.

Column Label	Contents	Units	Variable Type	Format	Req'd?
OBJECT	Object name		C*32 string	A32	Y
RA	Right Ascension	degrees	R*4 real	R11.6	Y
DEC	Declination	degrees	R*4 real	s12.6	Y
EQUINOX	Equinox date	years	R*4 real	F7.1	Y
MUALPHA	Annual p.m. in R.A.	sec/y	R*4 real	F7.4	N
MUDELTA	Annual p.m. in Dec.	arcsec/y	R*4 real	F7.3	N
EPOCH	Position date	years	R*4 real	F5.0	N
SPTYPE	Spectral type		C*15 string	A14	N
MAG	Approx. mag.		C*11 string	A11	N
COMMENT	Comment		C*32 string	A32	N

Table I.1: Columns for star-catalog table files

Proper motions

Proper-motion information requires three columns: the separate components, and the epoch of the catalog position. We follow the SAO Catalog in using annual proper-motion units of seconds of *time* per year in R.A., and seconds of *arc* per year in Dec. The epoch is given in decimal years. The respective column labels are MUALPHA, MUDELTA, and EPOCH. These are all Real*4 data.

Spectral types

Spectral types are stored in a column with the label SPTYPE. The contents of this column are character strings. You may make these strings as long as you want, but only the first 14 characters will be carried by the programs and formatted into output listings.

Rough magnitudes

Because approximate magnitudes are more useful when qualified by a terse description of their nature, the column labelled MAG should contain a character string rather than a number. For example, you might specify `'mpg=10.5 (HD)'`, or `'V=12.15'`. Up to 11 characters can be carried in this field.

Comment column

Up to 32 characters can be carried in an additional column, with the label COMMENT.

I.2.3 Standard values

The above information suffices for program stars. Standard-star files, however, must also have the standard photometric values (in place of the rough MAG column). The column label should be the usual name of the magnitude or index in the column, e.g., V or U-B or c1; see Table I.2. The data themselves are Real*4 numbers. Magnitudes are displayed in F5.2 or F6.3 format, which leaves room for two digits before the decimal point. Color and other indices are displayed in f6.3 format; this displays leading + signs.

System	Index	Col. label
UBVRI	V	`V`
UBVRI	B-V	`B-V`
UBVRI	U-B	`U-B`
UBVRI	V-R	`V-R`
UBVRI	R-I	`R-I`
UBVRI	V-I	`V-I`
uvby	b-y	`b-y`
uvby	u-v	`u-v`
uvby	m_1	`m1`
uvby	c_1	`c1`
H-beta	β	`beta`

Table I.2: Standard column labels for photometric indices

In addition, standard-star files should specify the system they employ, and the source of the standard values. The system should be placed in a character descriptor called **SYSTEM**. This may be up to 32 characters long, because of the need to distinguish between such alternatives as `'JOHNSON-MORGAN UBV'`, `'LANDOLT UBV'`, and `'KRON-COUSINS UBVRI'`,

Catalog stars may have not only columns specifying values in the system of observation, but also other systems. However, data from other systems, except for V magnitudes, will be ignored.

There is a potential problem in the use of indices like U-B as column labels that users should be aware of. MIDAS table files do not distinguish between upper and lower case in column labels. Thus, while it is possible to use `'u-b'` as a column label for the uvby system, it cannot be distinguished from `'U-B'` by programs or MIDAS procedures that read these files. However, the string may be entered in the proper case when the label is created, and will appear correctly on plots, table listings, etc.

Furthermore, because the provision for column arithmetic was built into the table system before the need for color index-names as column labels was apparent, it will be necessary to use the double-quote mark (") around such indices when referring to them as column labels. For example, in a MIDAS command line, the B-V column must be referred to as `:"B-V"`. Although this is inconvenient, it does allow such names to appear on plots, etc. The alternative (which will be automatically applied by MIDAS in the absence of the double-quote marks) is to convert the minus sign to an underscore, so that we would have B_V instead of B-V. This appears to be even more inconvenient for photometrists than to put up with the quotes.

I.2.4 Moving objects

For moving objects such as asteroids, comets, or planets, it is useful to include ephemeris information, both to provide predicted coordinates for planning observations, and to allow accurate airmass calculations in the reductions. Each such object requires several positions, each associated with a time. These times can be given either as Modified Julian Dates ($MJD = JD - 2400000.5$), or as ordinary date strings like `1995 Jan 23.0`. To allow good interpolation, include two ephemeris entries before and two after the interval of observations. The tabular interval need not be one day, but can have any constant value from one table entry to the next.

UT date — column label: DATE

Dates can be read by the programs in a variety of formats. To avoid ambiguity, use the first three letters of the month name instead of numerical month designations. The UT date should be stored in a C*16 (or shorter) string. The usual format in ephemerides is that shown above: year, month, day; 11 characters are enough in this case. However, the month, day, and year can be placed in any desired order.

MJD — column label: MJD_OBS

Occasionally, one finds an ephemeris given with Julian Dates as argument. Only the Modified Julian Date is used here. This must be a double-precision (R*8) variable, in F12.5 format. Modified Julian Dates are discussed further in section I.6 on data files.

The object name must be repeated on successive rows of an ephemeris file that refer to the same object; the repeated name tells programs to look for ephemeris data, and to interpolate positions as required. One table file can contain several objects, which may be convenient for some observing programs.

If the tabular positions are referred to the equator and equinox of date, as in some of the tables in the *Astronomical Almanac*, the EQUINOX column can be omitted. If astrometric positions are given, as in *Ephemerides of Minor Planets*, the EQUINOX column is required. In general, ephemeris data for moving objects should be kept in table files separate from star-position files.

I.3 Permanent telescope parameters

Data for all the telescopes at an observatory can be stored in a single MIDAS table file, called (by default) esotel.tbl. A standard path needs to be established for this file. Observers who use different observatories may need to keep files for two or more observatories.

Each observatory file should contain a C*72 descriptor **OBSERVATORY** that contains the name of the observatory, preferably as it is given in [6]. If effect, this descriptor takes the place of the standard descriptor **IDENT** in *.bdf files.

Each individual focus of each telescope has a single row in this table. The duplication of most information for the telescope itself is not a problem, as this will be a short file in any case. Each column described below should be present in the table, although some columns do not apply to some individual instruments, and will contain null entries.

Each column label can be a standard FITS keyword, if an appropriate one is available. The following sections describe the columns of the table in detail (cf. Table I.3).

I.3.1 Column label: TELESCOP

This is the name of the telescope focus, using the standard ESO archive notation [5]. The name begins with an abbreviation for the telescope's governing organization. The next part of the name is the aperture in decimeters (preceded by letters A, B, etc. if there are multiple telescopes with similar apertures). The name ends with the suffix P for prime focus, A for Cassegrain focus, C for coudé, and Nx for Nasmyth foci.

Thus, for example, the Cassegrain focus of the ESO 1.5-meter telescope is designated ESO15A. But this telescope has an asymmetrical mounting that is difficult to switch from one side of the pier to the other. In this case, it is necessary to operate on one side only during the night; this is indicated by appending the letter E or W to show which side of the pier the telescope is actually used on. So, for the 1.5-m telescope used east of the pier, the designation of the Cassegrain focus is ESO15AE.

I.3.2 Column label: DIAM

This is the actual diameter in meters of the telescope entrance pupil. For small telescopes, it provides significantly more precision than the rounded value embedded in the telescope name. For large telescopes with segmented apertures, it will suffice to specify the equivalent diameter corresponding to the total collecting area. (This will give the right value for the photon noise, but not for the scintillation noise. Most segmented apertures are on large telescopes whose scintillation noise is negligible anyway, so this is not a significant problem.)

I.3.3 Column label: LON

This is the longitude of the telescope, measured in decimal degrees *east* from Greenwich. Note that the sign convention for longitudes changed in 1984! The older versions of the Astronomical Almanac gave *west* longitudes. For photometric data reduction, an accuracy of about 0.003 degree (about 20 arcsec) is required. Current volumes of the Astronomical Almanac give positions to 0.1 arc minute, which is adequate for our purposes.

Astrometric observations, including occultation timings that may be measured with photometric equipment, require much better accuracy. You can find precise positions of telescopes in the pre-1984 volumes of the Astronomical Almanac. You should try to provide as accurate a value as possible.

Column Label	Contents	Units	Variable Type	Format
TELESCOP	focus name		C*6 string	A8
DIAM	diameter	meters	R*4 real	F7.3
LON	longitude	degrees	R*4 real	s12.5
LAT	latitude	degrees	R*4 real	s11.5
HEIGHT	height	meters	R*4 real	F6.0
TUBETYPE	tube type		C*6 string	A8
TUBEDIAM	tube diameter	meters	R*4 real	F8.3
TUBELEN	tube length	meters	R*4 real	F7.3
DOMETYPE	enclosure type		C*4 string	A8
DOMEDIAM	dome diameter	meters	R*4 real	F8.2
SLITWID	slit width	meters	R*4 real	F7.2

Table I.3: Column specifications for esotel.tbl file

I.3.4 Column label: LAT

This is the longitude of the telescope, measured in decimal degrees. The usual convention of + for north and − for south of the Equator should be observed. The same accuracy is required as for longitudes, so you should supply an accurate value.

I.3.5 Column label: HEIGHT

Height above sea level (meters). This need only be known to the nearest 10 meters. After a lapse of a few years, it is again listed in the "Observatories" section of the Astronomical Almanac, where you can find it for most established observatories. Topographic maps will provide good enough values for the newer sites.

I.3.6 Column label: TUBETYPE

This is a string, either 'OPEN' or 'CLOSED'. (Only the first letter is actually used.) Use 'OPEN' if the tube is an open framework, and 'CLOSED' if it is an opaque cylinder. If (as in some telescopes) the part near the declination axis is closed, but the front end of the tube is open, use 'OPEN'. This information is needed to determine when the Moon can shine directly on the objective.

If the telescope is a refractor used with*out* a dewcap, use 'OPEN'. A refractor used *with* a dewcap should be marked 'CLOSED'; the dewcap information will be used in the TUBEDIAM and TUBELEN columns (see below).

I.3.7 Column label: TUBEDIAM

If TUBETYPE is 'OPEN', ignore this field, and go on to DOMETYPE.

If TUBETYPE is 'CLOSED', TUBEDIAM should contain the *inside* diameter (in meters) of the telescope tube (or dewcap, if a refractor).

I.3.8 Column label: TUBELEN

If TUBETYPE is 'OPEN', ignore this field.

If TUBETYPE is 'CLOSED', TUBELEN should contain the length (in meters) of the telescope tube (or dewcap, if a refractor), measured from the front edge of the objective to the front end of the tube (or dewcap). Thus, for a reflector, TUBELEN is just the length of the telescope tube measured from the front surface of the primary mirror. For a refractor, it is the effective length of the dewcap.

I.3.9 Column label: DOMETYPE

If TUBETYPE is 'OPEN', DOMETYPE should contain the type of enclosure that surrounds the telescope. Use 'DOME' for a conventional dome with a slit, or for a turret with a slit of constant width. Use 'ROOF' for a building with a roll-off roof. Use 'NONE' for a roll-off building or any other disappearing enclosure that leaves the telescope unshaded from moonlight. As for TUBETYPE, only the first letter is significant.

If TUBETYPE is 'CLOSED', ignore this field.

I.3.10 Column label: DOMEDIAM

If DOMETYPE is 'DOME', and the telescope is a reflector, this is the dome diameter, in meters. If the telescope is a refractor, enter the distance from the objective to the surface of the dome, in meters.

If DOMETYPE is 'ROOF' or 'NONE', you are done; the remaining fields can be left empty.

I.3.11 Column label: SLITWID

If DOMETYPE is 'DOME', this is the width of the shutter or slit opening, in meters.

Some domes are equipped with upper and lower windscreens that could, in principle, be used to shade the mirror from moonlight. In practice, these screens always move too slowly to be used routinely in photometric programs. Therefore, wind screens are ignored, even though they might occasionally be used in particular programs.

I.4 Horizon obstructions

Information on horizon obstructions and limitations to telescope motion should be in a table file, which we can call the "horizon" file. Usually, the same restrictions apply to all foci of a telescope, and the same table file can be used for all of them.

The default name of the file is the "telescope" variable name described above, but with the suffix "hor.tbl" in place of the focus designation. Thus, for the ESO 1.5-meter telescope, the file name would be ESO15hor.tbl. If mechanical considerations make separate tables necessary for different foci of the same telescope, the "hor.tbl" should be appended to the full name of the focus.

For most telescopes, it is most natural to provide the hour-angle limits (east and west) as functions of declination. However, for telescopes with alt-azimuth mountings, the data may most easily be gathered in the form of altitude limits as functions of azimuth. No provision is made at this time for alt-alt mounts, although some are in use.

Notice that telescopes with German equatorial (cross-axis) and other asymmetrical mountings require two groups of columns: one for telescope east of the pier, and one for telescope west. Also, notice that "telescope east" is the position usually used in observing the *western* part of the sky, with the telescope above the polar axis.

Finally, two kinds of columns are required. The first subset specifies the observing limits (i.e., the accessible region of sky in which the telescope pupil is *completely* illuminated by a star). In this region, no part of the telescope pupil may be shaded by an obstruction. Photometric observations can be made only in this part of the sky.

The second subset specifies the region from which *any part* of the pupil may be illuminated by the Moon. As the Moon can only appear between limits of about $\pm 30°$ declination, the "moonlight" part of the table need only include this range.

Users should remember that trees have a tendency to grow larger, so that the "horizon" file should be re-checked from time to time if nearby trees are significant obstructions. Compilers of these tables should also bear this in mind, and leave a little margin for safety near trees.

I.4.1 Getting the data

If the photometer has an eyepiece, one can easily check whether the pupil is clear or obstructed by examining the exit pupil formed by the eyepiece. If you have a choice between eyepieces, choose the lowest power available to get the largest pupil. You may need to examine the pupil image with a magnifier; another eyepiece will do.

Telescopes without eyepieces can be checked by removing the instrument, and examining the focal plane by eye. The image of any distant obstruction will appear in the focal plane. Nearby obstructions more than a focal length away will be imaged behind the focal plane. The edge of the dome will be visible as an out-of-focus blur seen on the far side of the primary.

In either case, simply fix the telescope at a given declination (or azimuth, if it has an alt-az mounting), and move it in the other coordinate toward the horizon until an obstruction appears. The "observing" limit is a position at which the obstruction is near, but *completely outside of*, the usable field. The "moonlight" limit is the position at which the last speck of sky disappears behind terrestrial obstructions. If the "observing" limit is set by mechanical obstructions, you may have to estimate the "moonlight" limit, or just adopt the true horizon to be safe.

The measurements can easily be made in daytime, or during the brighter part of twilight. It will be most convenient to determine the "observing" and "moonlight" limits on the same side of the sky together, and then to move to the other side of the sky. The necessary data can be gathered in a few hours, and will prevent many unpleasant surprises while observing or in reducing observations.

Near $\delta = 90° - \phi$ and $\delta = \phi - 90°$, the limits change rapidly with declination, and should be gathered at 1° intervals. If there are no irregular obstructions, an interval of 10° is probably sufficient near the equator. You should assume that programs using the information in this table will interpolate linearly between the adjacent points, and adjust your spacing accordingly. (A program should be available to produce a blank form to fill in.)

I.4.2 Descriptor for the "horizon" table

A character descriptor is needed to specify which of three possible formats is actually used:

MOUNTING: possible values are `'FORK'`, `'GERMAN'`, or `'ALTAZ'`. `'ALTAZ'` is self-explanatory; `'GERMAN'` applies to "cross-axis" and other asymmetrical mountings that have different constraints, depending on which side of the pier the telescope tube is on. Everything else should be designated `'FORK'`, whether it really is a fork or yoke mount, or any other symmetrical form that is equatorially mounted and has no east-west differences like the German form (so called because it was first designed by Fraunhofer).

In the case of an alt-azimuth mounting that displays only equatorial coordinates to the user, it would be more convenient to use the `'FORK'` form of table, despite the actual mounting. (This may be the case for the NTT, for example.) If only Right Ascension is provided and not hour angle, the user will have to record the local sidereal time and compute the hour angle. Even so, this will be less work, and is less likely to introduce errors, than to convert between equatorial and horizon coordinates.

Three separate table formats correspond to each of these three descriptor values. The next three subsections describe these three forms.

I.4.3 MOUNTING=`'FORK'`

The independent variable (i.e., the reference column of the table) is labelled DEC, and contains the declination in degrees. The table entries are the hour-angles of the various limits, measured in decimal degrees.

Table I.4 shows the layout for fork mounts. Each column is described in detail below (cf. Table I.5).

DEC	OBSE	MOONE	OBSW	MOONW
+30.	285.4	279.3	104.5	109.1
+20.	283.6	276.5	94.7	104.3
+10.	280.3	272.7	91.1	99.5
0.	275.4	269.3	84.5	89.2
−10.	270.1	262.4	80.2	84.6

Table I.4: Example of partial table contents for fork-type equatorial mountings

Column Label	Contents	Units	Variable Type	Format
DEC	declination	degrees	R*4 real	F7.2
OBSE	eastern observation limit	degrees	R*4 real	F7.2
MOONE	eastern moonlight limit	degrees	R*4 real	F7.2
OBSW	western observation limit	degrees	R*4 real	F7.2
MOONW	western moonlight limit	degrees	R*4 real	F7.2

Table I.5: Column labels and contents in detail, for fork mountings

Column label: DEC

The values in the reference column are declinations, in decimal degrees. These are the fixed values set by the operator in compiling the obstruction-limit data. They should be accurate to about 0.1 degree of declination.

Column label: OBSE

The values in this column are the corresponding hour angles of the eastern "observation" limits, in decimal degrees. An accuracy of about 0.1 degree is adequate. Most telescopes read out hours and minutes instead; these should be acceptable inputs to the table-making program, which should do the conversion. If hours and minutes of time are read, try to read hour angles to the nearest minute or better.

In some cases, hour angles east of the meridian are read as negative values; in other cases, values will lie between 180° and 360° (12^{h} and 24^{h}). Both styles should be acceptable. The minus sign may be omitted if the values are numerically smaller than 180° (12^{h}).

Column label: MOONE

The values in this column are the corresponding hour angles of the eastern "moonlight" limits, in decimal degrees.

Column label: OBSW

The values in this column are the corresponding hour angles of the western "moonlight" limits, in decimal degrees.

Column label: MOONW

The values in this column are the corresponding hour angles of the western "moonlight" limits, in decimal degrees.

A simple sanity check on the tabular data is that the "moonlight" limits should always be closer to the horizon (i.e., farther from the meridian) than the "observing" limits, at every declination. The difference is approximately the angular size of the telescope entrance pupil, as seen from the object that obscures the horizon. If the obscuration is nearby, the difference may be many degrees; if distant, it may be a fraction of a degree.

I.4.4 MOUNTING='GERMAN'

The independent variable (i.e., the reference column of the table) is labelled DEC, and contains the declination in degrees. The table entries are the hour-angles of the various limits, measured in decimal degrees. All angles should be accurate to about 0.1 degree.

These asymmetrical mountings suffer different obscurations and mechanical limits when the telescope is east of the pier and west of the pier, so it is necessary to have a double-sized table. Each half of the table corresponds to the whole of a fork-mount table, but for a particular side of the pier. Therefore, you should read the detailed description in the previous section, for MOUNTING='FORK', before gathering the data for a German equatorial. It will probably be most convenient to gather all the data for the "telescope east of pier" position, and then all those for "telescope west".

The column labels are the same as for the fork mount, but prefixed by `TE` for telescope east of the pier, and `TW` for telescope west; see Tables I.6 and I.7.

DEC	TEOBSE	TEMOONE	TEOBSW	TEMOONW	TWOBSE	TWMOONE	TWOBSW	TWMOONW
+30.	5.6	−84.5	109.3	119.1	285.4	283.5	6.3	99.1
+20.	5.3	−70.1	104.7	114.3	283.6	281.7	6.5	84.3
+10.	5.4	−76.8	104.7	106.1	280.3	279.1	6.7	79.5
0.	5.4	−78.6	88.0	89.5	275.4	272.5	6.3	79.2
−10.	5.7	−85.7	79.3	82.1	270.1	267.2	6.4	84.6

Table I.6: Example of partial table contents for German equatorial mountings

Table I.6 shows the table layout for German equatorials. Each column is described in detail below; see Table I.7. See section I.4.3 on fork-mounts for more detailed discussions of the quantities in the table, as the entries are basically the same for both types of mounting.

Column label: DEC

Declination in decimal degrees. An accuracy of 0.1 degree is appropriate.

Column label: TEOBSE

Eastern "observing" hour-angle limit (decimal degrees) for Telescope East of the pier, at the given declination. Eastern hour angles may be given as negative quantities, smaller in magnitude than 180°.

Column Label	Tel. pos.	Contents	Units	Variable Type	Format
DEC	E of pier	declination	degrees	R*4 real	F7.2
TEOBSE	E of pier	"Obs." limit E	degrees	R*4 real	F7.2
TEMOONE	E of pier	"Moon" limit E	degrees	R*4 real	F7.2
TEOBSW	E of pier	"Obs." limit W	degrees	R*4 real	F7.2
TEMOONW	E of pier	"Moon" limit W	degrees	R*4 real	F7.2
TWOBSE	W of pier	"Obs." limit E	degrees	R*4 real	F7.2
TWMOONE	W of pier	"Moon" limit E	degrees	R*4 real	F7.2
TWOBSW	W of pier	"Obs." limit W	degrees	R*4 real	F7.2
TWMOONW	W of pier	"Moon" limit W	degrees	R*4 real	F7.2

Table I.7: Column labels and contents for German equatorials

Column label: TEMOONE

Eastern "moonlight" hour-angle limit (decimal degrees) for Telescope East of the pier.

Column label: TEOBSW

Western "observing" hour-angle limit (decimal degrees) for Telescope East of the pier.

Column label: TEMOONW

Western "moonlight" hour-angle limit (decimal degrees) for Telescope East of the pier.

Column label: TWOBSE

Eastern "observing" hour-angle limit (decimal degrees) for Telescope West of the pier.

Column label: TWMOONE

Eastern "moonlight" hour-angle limit (decimal degrees) for Telescope West of the pier.

Column label: TWOBSW

Western "observing" hour-angle limit (decimal degrees) for Telescope West of the pier.

Column label: TWMOONW

Western "moonlight" hour-angle limit (decimal degrees) for Telescope West of the pier.

Special remarks on German mountings

One should expect some "TEOBSE" and "TWOBSW" hour angles to lie near the meridian, because the telescope tends to run into the pier at declinations close to the latitude (i.e., near the zenith). There is the possibility that some telescopes cannot quite reach the zenith and may have small limits of either sign. For this reason, the minus sign cannot be optional for hour angles east, for German equatorials.

This mechanical limitation on setting the telescope may also affect the "moonlight" limits. Suppose the telescope is a reflector with an open tube, and can just reach the meridian when it is near the zenith. Then the Moon can only shine on the mirror (in this orientation) when it is high enough to shine in over the edge of the dome. As the lower edge of the dome is usually about the same height as the intersection of the polar and declination axes, the telescope mirror is some distance below the bottom of the dome shutter when the telescope points at the zenith. Then the Moon may have to be (say) 15° high to illuminate the mirror, with the telescope on this side of the pier. Such considerations should be taken into account in compiling tables for German equatorials. (See Table I.6 for examples.)

In some cases (e.g., the ESO 1.5-meter), it is very inconvenient to move the telescope from one side of the pier to the other during the night. With some instruments, it may be necessary to change wiring harnesses in reversing the telescope. Because photometry requires efficient use of telescope time, these situations make reversing the telescope impractical. The best way to handle such cases is to designate the "telescope East" and "telescope West" conditions as separate foci in the observatory table file. The obvious suffixes to use are E and W. Then separate horizon-limit tables would be used for the two conditions.

I.4.5 MOUNTING=`'ALTAZ'`

The independent variable (i.e., the reference column of the table) is labelled AZI, and contains the azimuth in degrees. Note that astronomical azimuth is normally measured positive *eastward* from the north point on the horizon. The table entries are the altitudes of the two limits, measured in decimal degrees.

It should be obvious in this table that the "moonlight" limit is always less than the "observing" limit. As azimuth runs from 0° to 360° completely around the horizon, there is no need to separate halves of the sky.

Table I.8 shows the layout for this form. Each column is described in detail below (see Table I.9).

Column label: AZI

Azimuth, in decimal degrees, at which a limit is determined. An accuracy of 0.1 degree is appropriate.

Column label: OBSALT

Limiting altitude for observations, in decimal degrees. An accuracy of 0.1 degree is appropriate.

Column label: MOONALT

Limiting altitude for moonlight, in decimal degrees. An accuracy of 0.1 degree is appropriate.

I.5 Instrument configuration and run-specific information

Information specific to a particular instrument is not so easy to categorize. There is some information connected with individual passbands that can naturally be stored in tabular form: the codes used for filters, and the names of the bands, for example. But each particular type of detector has a different set of characteristics, which in turn require different sets of supplementary data. Furthermore, there are distinct modes of operation peculiar to each type: we need dead-time information about photomultipliers only when they are used as pulse-generators, not in DC photometry or charge integration.

I.5.1 Storage format

This diversity of possibilities does not lend itself readily to MIDAS table structures. If, as is usual, only a single detector is used, it would be wasteful and inconvenient to burden a table with several columns containing identical detector data in every row. Furthermore, information that should be the same for every filter, such as detector information for a single-channel instrument, could become inconsistent if presumably duplicate entries were stored in table format. To include code to test for the consistency of such constant-valued columns would impose overhead and maintenance problems for the programs that read the table.

A possible solution would be to use a table file with one row per filter, and to store information that remains the same for *all* filters in descriptors. This seems to be the policy that has been adopted for FITS tables. Then ordinary single-channel instruments could keep all the detector information in descriptors. But multi-channel instruments should

AZI	OBSALT	MOONALT
0.	5.4	4.5
+10.	0.3	0.1
+20.	3.6	2.7
+30.	5.4	4.5

Table I.8: Example of table contents for alt-azimuth mountings

Column Label	Contents	Units	Variable Type	Format
AZI	azimuth	degrees	R*4 real	F7.2
OBSALT	max. altitude for observations	degrees	R*4 real	F7.2
MOONALT	max. altitude for moonlight	degrees	R*4 real	F7.2

Table I.9: Column labels and contents for alt-azimuth mountings

store the detector information for each passband in the table itself, restoring the problem of duplicated data for all the bands that use the same detector.

On the other hand, some kind of array structure is needed to hold the information about detectors in multi-channel instruments. But the channels do not necessarily correspond to passbands in a one-to-one way; for example, an instrument might use a blue-sensitive photomultipler tube to measure U, B, and V, and a "red" tube to measure R and I (see the example in section I.5.7). We could store the detector information as MIDAS "descriptors"; then the problem is that instruments with multiple detectors would require multi-element descriptors, and a cross-reference table column to identify which detector goes with which filter combination.

A practical if not very elegant solution is to store everything in one *physical* table file, which contains two logical sub-tables – one for passbands, and one for detectors. Each sub-table contains an explicit index column, to allow explicit cross-references, despite any rearrangement of the actual table rows. This index column serves as the natural sequence number within each sub-table. This reduces the number of files the user has to keep track of. Invariant information for the whole instrument then goes in the table file's descriptors.

Most of the information in this table is stored as strings, rather than numerical values, thus keeping the entries easy for humans to read. Many of the items make sense (and will be looked for) only if others have particular values; see Tables I.10, I.11, and I.12.

I.5.2 General instrumental information

The two principal classes of information needed about photometric instruments concern filters and detectors. We need two kinds of information for filters and detectors: physical information about the instrument itself, and logical information about the way it is represented in data files – for example, the codes used to identify filter positions.

In addition, it is useful to know the condition of the telescope optics: clean optics give not only better signal/noise ratio than dirty ones, but also more stable zero-points from night to night. Sometimes it is possible to have the optics cleaned before a critical observing run; observers should consider this possibility.

Some, but by no means all, of the required information is available in the ESO Archive – mostly in the archive log files [5]. This can be stripped out and used when it is available; when it is not, the observer will have to supply the information.

In addition to information about the instrument itself, the reduction program needs to know the structure of the data in their table file (see section I.6, "Observational data").

Thus, you must make sure that information such as the number of filter-code columns and the filter codes that correspond to particular passbands, which is in the instrument-description table, agrees with the actual data tables. Usually, this is determined by the program that converts raw data files to table format.

I.5.3 Passbands

To identify the passbands, we need a small sub-table to hold the relation between codes recorded in the raw data and the standard passband names (see Table I.10), as well as other information. The number of passbands should always be the number of rows in the table, so there is no need to save it explicitly.

Columns BAND and NBAND

The column named BAND gives the standard name of the band. This column contains 8-byte character variables, and is normally displayed with A8 format.

Standard values for the BAND column are:

for the UBV(RI) system: `'U', 'B', 'V', 'R', 'I'`

for the 4-color system: `'u', 'v', 'b', 'y'`

for the H-beta system: `'betaW', 'betaN'` (for Wide and Narrow, respectively)

for red-leak filters: the name of the main band followed by `'RL'`; e.g., `'URL', 'BRL'`, etc.

for "neutral" filter combinations: the name of the band, followed by `'ND'` if only one value of attenuation is used, or by `'ND1', 'ND2'`, etc., if more values are used

for any opaque position: `'DARK'` for a single detector; or `'DARK1', 'DARK2'`, etc. for multiple detectors

Notice that each distinct type of measurement counts as a separate band entry, including measurements made with different neutral-density filters, red-leak, and dark measurements. For example, a one-channel photometer with a filter wheel that measures U, B, V, and the red leak in U, plus a slide containing open, opaque, and two neutral-density filters would have [4 filters $\times$ (2 ND + 1 clear positions = 3 different intensity measurements)] = 12 different rows, plus a 13th row for the dark measurement. The combinations of red-leak filters that involve ND filters should be named with the RL before the ND — e.g., URLND1. For more details, see the examples in subsection I.5.7, "Sample instrument files".

The column named NBAND contains an integer used for cross-referencing. An I4 format is convenient. Normally, this is the reference column for the whole table file.

Column FILTCODE_n

If **NFILTCAR** (see section I.5.4, "Filter descriptors") contains 1, a column headed FILTCODE_1 gives the code in raw data files that represents each position of the filter carrier. If multiple filter carriers are coded separately, the column FILTCODE_*n* contains the coding for the *n*-th carrier. On the other hand, it may be convenient to combine two or more filter mechanisms into a single filter-code column in the instrument table.

The values used in the FILTCODE column(s) depend on the particular data-logging system. They can even be the standard names of the bands. Thus, FILTCODE may just be a duplicate of the BAND column. These columns are 8-byte character variables, and are normally displayed with A10 format.

Some multichannel instruments have separate filter mechanisms for each detector. For example, a two-channel instrument might use "red" and "blue" beams. Then the filters used in the "red" beam would not affect measurements made in the "blue" beam. In this case, the word "any" can be inserted in the filter code for the beam that is *not* used with a given output channel (see examples below).

For spectrometric multichannel instruments, there may be no choice of filters for any channel: the data from different channels are distinguished by position rather than by code in the raw data. Then **NFILTCAR** can be set to zero, and the FILTCODE column can be omitted from the instrument description file. In this case, the separate channels must still be identified on separate rows in the final data table file (see section I.6, "Observational data") by the standard names of the bands.

On the other hand, the Danish 6-channel spectrometer at La Silla has two separate mechanisms for inserting neutral-density filters. In this case, one must append an ND code to the passband name in the data file (see section I.6, "Observational data"), to indicate the combination of neutral filters in use.

Note that the OPTI-n keyword in the ESO Archive [5] includes other optical elements than filters, and so is not uniquely related to the numbering of filter wheels.

Column NDVALUE

If "neutral" filters are used, their quantitative effect should be stored in a column named NDVALUE. The real number stored here should normally be the factor by which the intensity measured through the ND filter should be multiplied to be on the same scale as data taken without the ND filter. Therefore, the numbers are normally all bigger than unity, except for the un-attenuated passbands, for which one normally puts unity in this column. Because "neutral" filters are not really neutral, the value will differ somewhat from one passband to the next.

Column NDETUSED

The column named NDETUSED gives the *number* of the detector used to measure each band. This number corresponds to the number stored in the NDET column of the detector sub-table (see section I.5.5). It need not correspond to a data-logging code.

Columns REDLEAK, RLTYPE, and MAKER

Special problems arise with filters intended to measure red leaks in blue and ultraviolet filters. If the "red-leak" filter is simply an uncemented sandwich of the short-wavelength filter and a sharp-cutoff (long-pass) filter to block the main passband, the leak will be measured about 8% too low, because of Fresnel reflections at the two added air-glass surfaces. An accurate red-leak measurement is possible only if the leak-measuring filter is a cemented combination, *and* if the leak-isolating component does not absorb (as Pyrex glasses do) at the wavelength of the leak. These problems are most easily handled by including some additional columns in the table (see table I.10).

The column named REDLEAK contains an 8-character string that specifies the treatment of red leaks in short-wavelength filters. Possible values are `'MEASURED'` if the red leak is measured by observations through a filter that isolates the leak; `'BLOCKED'` if a copper-sulfate or other blocking filter is used; or `'IGNORED'` if the leak is neither measured nor blocked. `'ABSENT'` can be used for long-pass filters, like the V of standard UBV. It is the user's responsibility to determine that blocking is adequate, particularly if interference filters and/or red-sensitive detectors are used. An unblocked red leak can produce very large transformation errors.

If the leak is `'MEASURED'`, additional information is required, because most instruments do not provide a true measurement of the red leak. The additional information is stored in character columns named RLTYPE and MAKER.

Column RLTYPE may contain the values `'CEMENTED'` if the ultraviolet and long-pass components are cemented together; `'LOOSE'` if the two components are *not* optically contacted; `'UNKNOWN'` if information is not available. If the filters are loose, the two extra air-glass reflections cause excess loss that must be accounted for.

The column named MAKER may take the values `'CORNING'` if a Corning or Kopp (successor to Corning) glass is used for the *long-pass component* of the red-leak filter; `'SCHOTT'` if a Schott glass (or other non-Pyrex base glass) is used; or `'UNKNOWN'` if information is not available. This information is required because the Pyrex glass used as the base for Corning filters absorbs appreciably at typical red-leak wavelengths; the measured leak must therefore be increased to compensate for the absorption.

Column Label	Contents	Variable Type	Format	When used
`BAND`	band name	C*8 string	A8	always
`NBAND`	band number	I integer	I4	always
`FILTCODE_1`	filter code for 1st wheel	C*8 string	A10	always
`FILTCODE_`*n*	filter code for *n*th wheel	C*8 string	A10	**NFILTCAR** > 1
`NDVALUE`	attenuation factor	R real	F7.3	ND filters used
`NDETUSED`	detector number	I integer	I4	**NDETS** > 1
`REDLEAK`	red leak treatment	C*8 string	A8	always
`RLTYPE`	RL filter construction	C*8 string	A8	REDLEAK=`MEASURED`
`MAKER`	RL isolating glass maker	C*8 string	A8	REDLEAK=`MEASURED`

Table I.10: Passband columns of the instrument table file

Sometimes only the shortest-wavelength band of a system has red-leak problems (e.g., U of UBV, or u of uvby.) However, if silicon detectors or red-sensitive photocathodes are used, blue and even green ("visual") bands may have red-leak problems, especially if heavily reddened or late-type stars are observed.

Note that the importance of red leaks depends on the photometric system, the filter set, the detector used, and the stars observed. Failure to treat red leaks correctly will produce serious systematic errors, which cannot be "transformed away" by any reduction program. Worse yet, incorrect treatment of red leaks can propagate these errors (through incorrect transformations) into data for early-type stars that otherwise have negligible red-leak problems. It is the user's responsibility to be aware of these problems.

For additional information on red leaks, see Shao and Young [8], besides the very brief treatment on pp. 109 and 184 of Young [10]. Stetson [7] has illustrated how severe the problem can be when cool objects (light bulbs!) are observed with CCDs. Note his warning: "Don't be satisfied with statements like 'The red leak is negligible'." Unfortunately, the important cautionary remarks of Ažusienis and Straižys [2] regarding reddened stars are available only in Russian. They show that the simple correction formula used by Shao and Young [8] is not correct for heavily reddened stars.

To sum up, one may say that red leaks should be measured whenever they exceed the accuracy desired in the final results – which is more often than you might think.

I.5.4 Instrument descriptors

Properties of the instrument that are associated with the instrument as a whole, rather than with a particular filter or detector channel, belong in descriptors in the instrument table file. Most of these relate to the filter mechanisms. In addition, it is convenient to have a character*72 descriptor called **INSTNAM** that gives the name or designation of the particular instrument; and one to describe the condition of the telescope optics, which will be treated at the end of this section. The descriptors are summarized in Table I.11.

Filter descriptors

The two most critical types of physical information needed about the filters themselves are (1) what are their transmission curves? and (2) does the instrument either regulate or measure their temperature? Filter temperature is important, because filters are somewhat more temperature-sensitive than detectors (see [9], and pp. 105-108 of [10]). In addition, we need to know how many different filter-code fields appear in the data. Bearing all this in mind, here are the descriptors for filters:

Descriptor NFILTCAR

This integer descriptor tells the number of *filter-code fields* in the data files, and hence the number of FILTCODE_*n* columns in the instrument table. It is usually the same as the number of filter carriers (usually wheels or slides). NFILTCAR is the logical rather than the physical number of carriers; if two or more filter wheels are encoded as a single character in the data, there is effectively only one filter wheel, as far as data handling is concerned.

If one filter mechanism carries chromatic filters and another carries "neutral" filters, the total number of bands (i.e., rows) in the table should be the product of the number of positions in the two carriers. Thus, a 4-position main filter wheel and a 3-position attenuator in the same measuring beam give 12 total passband combinations. These may be described as a single logical filter mechanism if the positions of the two wheels are indicated in adjacent data columns that can be combined into a single code field. They may equally well be treated as two logical filter carriers with separate code columns.

It is also possible to have two filter wheels in series, arranged so that one wheel carries chromatic filters for one photometric system, but "neutral" filters for another, whose chromatic filters are in the second wheel. Then only the combinations that make sense need be included in the instrument table; those that would put two neutral filters in series, or two chromatic filters of different systems, can be omitted. (Note that there might be useful combinations of two chromatic filters: if one wheel contains UBV filters and the other contains uvby filters, one might combine a V filter with u to give a uRL combination.)

Character descriptor FILTCAT

If curves are available for the filters used, this holds the name of a MIDAS catalog file that points to individual table files, which in turn hold the transmission data.

The individual filter table files should give transmittance (column name TRANS) as a function of wavelength in nanometers (column name LAMBDA). Each filter's table file should contain a character descriptor named **BAND** that names the passband for which that filter is used.

If no filter curves are available, **FILTCAT** should contain only spaces.

Character descriptor FILTSTAT

This descriptor specifies the state of filter temperature control. It contains the value `'REGULATED'` if filter temperature is regulated; `'MEASURED'` if filter temperature is measured; and `'DOME'` if filters are unregulated, and approximately at dome temperature.

If **FILTSTAT**= `'REGULATED'`, there may be a Real*4 descriptor named **FILTTEMP** in the instrumental ".tbl" file, giving the temperature of the set-point in kelvins. If **FILTSTAT**= `'MEASURED'`, there should be a data column named FILTTEMP in the *data* files (see next section) that contains the measured filter temperature.

Name	Contents	Variable Type	When used
INSTNAM	name of instrument	C*72 string	always
NFILTCAR	no. of filter carriers	I integer	always
FILTCAT	filter catalog name	C*80 string	always
FILTSTAT	filter thermostating	C*9 string	always
FILTTEMP	filter temperature	R*4 real	**FILTSTAT**=`REGULATED`
NDETS	no. of detectors	I integer	always
CONDITION	condition of optics	C*7 string	always

Table I.11: All possible Descriptors for the instrumental ".tbl" file

I.5.5 Detectors

The number of detectors is always kept in an integer descriptor, **NDETS**. This is ordinarily the same as the number of output data channels from the instrument.

Columns NDET and DETNAME

The integer column named NDET contains the sequence number for the detector sub-table. Each detector channel has its own row in this sub-table. For multichannel instruments that devote one detector to each passband, these rows can be considered a natural continuation of the rows of the passband sub-table. In this case, the NBAND and NDET columns should have the same values within each row.

The A*16 column named DETNAME is optional, and is provided only for the user's convenience. As its name indicates, it can hold a character string naming the detector.

Column DETCODE

If a multichannel instrument indicates which channel is being read out by a code, it should be kept in a column named DETCODE.

Column DET

The detector type should be specified, so that programs using the data can have approximate information about the spectral response and other characteristics of the detector. The primary indicator is a character column, DET, which contains `'PMT'` if the detector is a photomultiplier; `'SILICON'` for Si-CCDs and Si photodiodes; or `'OTHER'` for any other type. (As detector technology evolves, one would expect to extend this list.)

Photomultipliers: DET=`PMT`

The spectral responsivity of a photomultiplier depends on the photocathode composition, the window composition, and the mode of illumination (i.e., from the vacuum side or the substrate side). For many common situations, these factors have been combined into standard spectral responses known as "S-numbers" (S-1, S-4, S-20, etc.) that are used by most, but not all, manufacturers. The spectral response should therefore be indicated by a character column named SNUMBER, containing the S-number, if it is known (e.g., `'S-13'`). Other valid string values are `'BIALKALI'` for "bialkali" photocathodes with glass windows; `'QBIALKALI'` for "bialkali" photocathodes with fused-quartz windows; `'GAAS'` for gallium arsenide "negative-electron-affinity" photocathodes with glass windows; and `'QGAAS'` for gallium arsenide "NEA" photocathodes with fused-quartz windows. Any spectral response markedly different from these should be flagged as SNUMBER=`'OTHER'`, and the spectral response supplied in a separate table file, as described under DET=`'OTHER'` (see section I.5.5).

Photomultipliers are used in different modes, which have different properties. This is specified by a character column MODE, whose values may be `'PC'` for pulse counting; `'DC'` for DC photometry; or `'CI'` for charge integration.

If MODE=`'PC'`, additional information is needed to describe the pulse-overlap ("dead-time") correction. This is given in another character column, DEADTYPE, which can have the values `'EXTENDING'`, for a paralysable counter, or `'NONEXTENDING'`, for a non-paralysable counter. If the counter's behavior is unknown, and cannot be determined, set DEADTYPE=`'UNKNOWN'`. The estimated value of the dead-time parameter itself, in seconds, goes in a Real*4 column called DEADTIME, and the uncertainty of the dead-time parameter, again in seconds, goes in a Real*4 column called DEADTIMEERROR.

Users should be aware that, because pulse pile-up partly offsets coincidence losses, the effective dead-time parameter depends on a combination of the resolution of the discriminator, the discriminator setting, and the characteristics of the individual photomultiplier under actual conditions of use, such as temperature and voltage, that affect the pulse shape and pulse-height distribution. Therefore, it is essential to keep these parameters fixed during a run. Also, the effective dead-time parameter should be determined from actual photometric data gathered for the purpose of determining its value accurately; nominal values of pulse-resolution times from manufacturers, or pulse-resolution times determined with pulse generators, are *not* suitable for correcting photometric measurements.

Column COOLING

As the stability of zero-points and transformation coefficients depends partly on the detector temperature, information must be given on the detector cooling arrangements. This is stored in a character column called COOLING, which may contain the strings `'REGULATED'` if an active closed-loop cooling system is used (either thermoelectric or some other servo-controlled cooling); `'UNREGULATED'` if the tube is cooled in some way, but not regulated; `'ICE'` if ordinary (water) ice is used as coolant; `'DRYICE'` if dry ice (solid carbon dioxide) is used with*out* a heat-transfer fluid; `'MEASURED'` if the temperature is measured but not regulated; or `'NONE'` if the PMT runs at ambient temperature.

One should be careful not to push temperature-regulated systems beyond their capabilities. A servo-controlled system that tries to cool all the time and never oscillates about its set point is, in fact, unregulated rather than regulated. As long as dark current is well below sky measurements, dark noise probably adds little to the photon noise from the sky, and additional cooling is unnecessary. If the dark current is kept a little above the minimum possible value, so that it remains in the strongly temperature-dependent regime, it serves as a useful indicator of tube temperature and health.

If dry ice is used with a low-viscosity heat-transfer liquid such as ethyl acetate or Freon-11, COOLING should be set to `'REGULATED'`; but if dry ice is used with a viscous fluid such as alcohol, `'DRYICE'` should be used. Note: Freon-11 is one of the CFCs most destructive to the ozone layer, and should be avoided.

If COOLING=`'REGULATED'`, a Real*4 column named DETTEMP should contain the estimated detector temperature in kelvins. With dry ice and ethyl acetate, the value in DETTEMP should be 195, corresponding to -78°C.

If COOLING=`'MEASURED'`, detector temperatures should be part of the regular data stream. In this case, a Real*4 data column DETTEMP should be in the data files, rather than in the instrumental ".tbl" file. (Cf. the similar usage of FILTTEMP, and further details in the next section.)

Column Label	Contents	Variable Type	Format	When used
DET	detector type	C*8 string	A8	always
DETNAME	detector name	C*12 string	A12	optional
NDET	detector number	I integer	I4	always
DETCODE	detector code	C*8 string	A8	if dets. are coded
SNUMBER	PMT S-number	C*9 string	A9	DET=PMT
MODE	PMT mode (PC/DC/CI)	C*9 string	A9	DET=PMT
DEADTYPE	PMT deadtime type	C*12 string	A12	DET=PMT
DEADTIME	PMT deadtime (sec)	R*4 real	E8.3	DET=PMT
DEADTIMEERROR	PMT deadtime error	R*4 real	E13.2	DET=PMT
COOLING	detector cooling	C*12 string	A12	always
DETTEMP	detector temperature	R*4 real	F8.1	COOLING=REGULATED
SPECTRESPTBL	actual spectral response	C*(*) string	A	(see section I.5.5)
BLUERESP	CCD blue response class	C*8 string	A8	DET=SILICON

Table I.12: Detector columns of the instrument table file

Silicon photodiodes and CCDs: DET=SILICON

Unfortunately, the spectral responses of these devices are quite varied, depending on details of manufacturing (e.g., polysilicon vs. aluminum electrodes; thinned vs. non-thinned), preparation (use of blue-enhancing treatments such as phosphors, UV-irradiation, and flashgates), and use (front-side vs. back-side illumination). The response also depends on temperature, as it does for all semiconducting materials.

The preferred method of handling this problem is to use a table of (averaged) spectral responsivity determined for the individual device under actual conditions of use. This means using the spectral-response table file described under DET=OTHER (section I.5.5). The name of the table file should be contained in a character column named SPECTRESPTBL, of adequate width to hold a file name (see section I.5.5 for a description of the file). If no table is available, this column can be omitted. If tables are available for only some detectors, the column should exist, and contain blanks in the rows of detectors lacking detailed information.

Even if no detailed spectral response is available, it is sometimes necessary to make a rough guess at the spectral response, particularly for planning purposes. Then, if SPECTRESPTBL is missing, there should be a character column named BLUERESP that gives some indication of the expected blue response. Valid string values are `'FRONT'` for front-side illumination of a normal CCD, with no phosphor coating; `'BACK'` for thinned, un-enhanced CCDs illuminated from the substrate side; `'ENHANCED'` for CCDs treated by UV irradiation, special gases, or flashgate; or `'PHOSPHOR'` for CCDs coated with a blue-sensitive phosphor. These general categories provide some information on spectral response, but it may easily be in error by a factor of 3 or more.

As for PMTs, the COOLING column is required. Usually, CCDs are cooled with liquid nitrogen, but are actually kept thermostatted by a servo system. In this case, COOLING=`'REGULATED'`. The same applies to chips cooled thermoelectrically, in most cases. Occasionally one finds unregulated cooling; then COOLING=`'UNREGULATED'`, and the treatment is the same as for cooled but unregulated PMTs (see above).

Other detectors: DET=`OTHER`

Other detectors *require* a spectral-response table file, whose name should be the contents of a character column named SPECTRESPTBL in the detector sub-table of the instrumental ".tbl" file.

This spectral-response table file uses a column named `'LAMBDA'` (containing wavelengths in nanometers) as the reference column. Responsivities, in amperes per watt, go in a column named `'RESP'`. (Observers who have difficulty converting manufacturers' data from microamperes per microwatt to amperes per watt should be encouraged to take up a less demanding field than photometry.) Alternatively, responsive quantum efficiencies can be put in a column named `'RQE'`; this quantity is dimensionless. All these data are Real*4. The table should contain one character descriptor, named **RESPTYPE**, whose value is either `'RESP'` or `'RQE'`, to indicate which type of data are tabulated.

The remarks made above about the COOLING column apply here also; see the discussions given in connection with PMTs and CCDs.

I.5.6 Telescope optics

Finally, the condition of the telescope optics should be stored in a character descriptor named **CONDITION** in the instrumental ".tbl" file. Possible values are `'CLEAN'`, if the mirrors have been re-coated or cleaned within two weeks of the observing run, and/or a flashlight beam is barely visible on optical surfaces; `'AVERAGE'`, if a flashlight beam is plainly visible on the optical surface, but the dirt is not very bad; and `'DIRTY'`, if shining a flashlight beam on the primary produces a sensation of revulsion in a trained observer. Obviously, some experience is required to judge these categories accurately.

I.5.7 Sample instrument files

Because the instrumental table file is rather complicated, and has a variable structure that depends on the nature of the instrument, here are some examples.

A simple photometer

First, consider a very simple 1-channel UBV photometer. Let us assume a rather minimal instrument: no cooling or temperature regulation, and simple DC photometry. This resembles the instrument with which the UBV system was first set up. Here is the table:

Descriptor values:

NFILTCAR = 1
NDETS = 1
FILTSTAT = `DOME`
FILTCAT = ' ' (blank string)
CONDITION = `AVERAGE`

Passband sub-table

NBAND	BAND	FILTCODE_1	REDLEAK	RLTYPE	MAKER
-----	----	----------	--------	------	-------
1	U	3	MEASURED	LOOSE	CORNING
2	B	1	IGNORED		
3	V	2	ABSENT		
4	URL	4	ABSENT		

Detector sub-table

NDET	DET	SNUMBER	MODE	COOLING
----	---	-------	----	-------
1	PMT	S-4	DC	NONE

Table I.13: Instrument table file for a simple photometer

As there is only one detector, we need not include the NDETUSED column.

A modern single-channel photometer

Next, consider a more modern photometer that uses pulse-counting. It has a "neutral" filter in addition to standard uvby filters. The "neutral" filter is in series with the passband filters, and is used to determine the dead-time correction. The positions of these two filter mechanisms are recorded as two adjacent digits in the data, so we can treat them as a single filter code; the second digit is 1 when the "neutral" filter is in the beam.

There is also a shutter whose position is encoded separately in the data stream: 0 means open, 1 means shut. This can be treated as a second logical filter mechanism.

The photomultiplier in this modern instrument is in a thermoelectrically cooled chamber, regulated to run at 0° C (273 K). Table I.14 shows what we get.

Again, there is only one detector, so we need not include the NDETUSED column.

A modern two-channel photometer

Next, consider a more elaborate UBVRI photometer with two channels (a "red" and a "blue" tube). We suppose it has separate filter wheels in the two beams; the first filter wheel contains the UBV filters, and the second has the R and I filters. Filter position 0 in each wheel is opaque; note that we need separate DARK codes for the two channels. Red

Descriptor values:

NFILTCAR = 2
NDETS = 1
FILTSTAT = `DOME`
FILTCAT = `' '` (blank string)
CONDITION = `AVERAGE`

Passband sub-table

NBAND	BAND	FILTCODE_1	FILTCODE_2	NDVALUE	REDLEAK	RLTYPE	MAKER
-----	----	----------	----------	-------	--------	--------	------
1	u	10	0	1.000	MEASURED	CEMENTED	SCHOTT
2	v	20	0	1.000	BLOCKED		
3	b	30	0	1.000	BLOCKED		
4	y	40	0	1.000	BLOCKED		
5	uND	11	0	9.897	MEASURED	CEMENTED	SCHOTT
6	vND	21	0	9.763	BLOCKED		
7	bND	31	0	9.674	BLOCKED		
8	yND	41	0	9.695	BLOCKED		
9	DARK	any	1				

Detector sub-table

NDET	DET	SNUMBER	MODE	DEADTYPE	DEADTIME	DEADTIMEERROR	COOLING	DETTEMP
----	---	-------	----	---------	--------	-------------	---------	-------
1	PMT	QBIALKAI	PC	EXTENDING	6.74E-8	4.2E-9	REGULATED	273.

Table I.14: Instrument table file for a typical modern photometer

leaks are blocked, and filter temperatures are measured.

If the filter wheels had been in series instead of in separate beams, the unused wheel would have to be set to a clear position, instead of the "any" entered in the first five lines of Table I.15.

Because we have two detectors, the NDETUSED column is mandatory; see Table I.15. In our example, the "blue" tube is uncooled, but its temperature is measured; while the "red" tube is cooled with dry ice. Both run in charge-integration mode.

CCD photometry

Our final example is an instrument-description file used for B and V magnitudes extracted from CCD frames. The CCD is a front-illuminated chip, run at 180 K. The filters are at ambient temperature, uncontrolled. Bands are identified by name in the file of extracted values.

The instrumental description is in Table I.16. Notice how this resembles the primitive

Descriptor values:

NFILTCAR = 2
NDETS = 2
FILTSTAT = MEASURED
FILTCAT = ' ' (blank string)
CONDITION = AVERAGE

Passband sub-table

NBAND	BAND	FILTCODE_1	FILTCODE_2	REDLEAK	NDETUSED
1	U	1	any	BLOCKED	1
2	B	2	any	BLOCKED	1
3	V	3	any	ABSENT	1
4	R	any	1	BLOCKED	2
5	I	any	2	ABSENT	2
6	DARK1	0	any		1
7	DARK2	any	0		2

Detector sub-table

NDET	DET	SNUMBER	MODE	COOLING
1	PMT	S-13	CI	MEASURED
2	PMT	GAAS	CI	DRYICE

Table I.15: Instrument table file for a two-channel photometer

UBV photometer in the first example.

I.6 Observational data

Every instrument seems to produce its data in a different format. However, it is relatively simple to re-format the data to a standard form; MIDAS table files are the obvious standard form to use. Usually the conversion is done in two steps: first reformat the existing data as an ASCII file with all records in the same format, and then convert this ASCII file to MIDAS table format.

The ASCII re-formatting can be done by a user-written program, or by using UNIX tools such as the stream editor (**sed**) and the table-oriented programming language **awk**. While the UNIX manual pages provide little useful information on these tools, there are some excellent books available, such as [4] and [1].

Often, much of the work has already been done. If a program exists that reads the current instrumental data, it can readily be modified to read the data and then reproduce them as an ASCII file, suitable for conversion to MIDAS table format. The data-reading

Descriptor values:

NFILTCAR = 1
NDETS = 1
FILTSTAT = `DOME`
FILTCAT = `' '` (blank string)
CONDITION = `AVERAGE`

Passband sub-table

NBAND	BAND	FILTCODE_1	REDLEAK
1	B	B	BLOCKED
2	V	V	BLOCKED

Detector sub-table

NDET	DET	BLUERESP	COOLING	DETTEMP
1	SILICON	FRONT	REGULATED	180.

Table I.16: Instrument table file for a simple CCD setup

part of the existing program is adaptable as the front end of the reformatter. To simplify the conversion, FORTRAN routines will be made available to handle the back end. Thus, very little work really has to be done. In any case, the reformatting program only has to be written once for a given instrument.

ESO telescopes provide relatively clean data files, but these files contain more than one kind of record. Thus, even these data files must be reformatted before they can be converted to MIDAS tables by the CREATE/TABLE command. Programs will be provided to convert data from ESO telescopes to the standard table format.

I.6.1 Required observational data

A certain minimum set of data is required to make observations reducible. These are the identity of the object measured; the identity of the bandpass in which the measurement was made; the time of the observation; and, for integration-type measurements, the duration of the integration. Table I.17 describes the basic table-file columns in detail.

What was the measurement?

The measurements themselves should be stored in a column with the label SIGNAL. These are Real*4 data; the format specified in the table may depend on the instrument. Often the readings are recorded as integers; then assume the decimal point at the end of the field.

Programs will expect that SIGNAL represents the integration of the photon flux for the exposure time given in the EXPTIME column (see section I.6.1). That is, it is the *ratio* of SIGNAL to EXPTIME that represents an actual photon count *rate*, or intensity, if the exposure times vary.

This assumes the data are in arbitrary intensity units (times time). Sometimes one must deal with data in the form of magnitudes, as in re-reducing old data, or data measured from strip-charts with a magnitude ruler. In that case, the column label should be RAWMAG; see Table I.18. Programs will expect RAWMAG to be the usual negative logarithms of intensities; that is, RAWMAG values only use EXPTIME in determining weights.

Except for pulse counting with photomultipliers, where reasonably accurate models exist for the nonlinearity, the SIGNAL values should have been corrected for nonlinearity. In any case, the EXPTIME values should have been corrected for differences between nominal and actual exposure times. Such corrections are especially important for CCD data, where they can vary across the frame (see Stetson [7]).

What was measured?

An OBJECT column gives the name of the object observed (usually a star name). Many instruments record only a code for the object, instead of a name. These codes must be turned into standard names for the observational-data tables. Presumably the owners of such instruments already have software to do this that can be cannibalized for the format conversion.

For measurements of dark current, the OBJECT column should contain the word `DARK`. As dark measurements must be referred to the electrical zero of the system, they are usually accompanied by such measurements. The electrical-zero values should be identified as `ZERO` in the OBJECT column. If no such data exist, they will be assumed numerically zero.

Naming the object is simple when the measurement is just star + sky; one normally uses the name of the star. However, sky observations must be matched up with the proper stars. This is not a trivial task, for multiple sky observations may be used for one single star, and multiple star observations may have to share a single sky measurement.

Thus, "sky" alone is not a sufficient identification; it must be something like "sky for such-and-such a star". A related problem is to know the coordinates where the sky is observed, which are needed in modelling the sky brightness. In critical cases, it is common practice to measure sky on two or more positions around a target object, and these positions must be identified. Finally, we must be able to use sky-subtracted data, for which the sky brightness may no longer be available (some CCD photometry falls in this category).

A general treatment of this problem requires a column called OBJECT, bearing the usual name of the object (as in the star tables); a column labelled STARSKY, to distinguish between star+sky, star alone, or sky alone; and additional columns to identify the sky position.

Valid strings for the STARSKY column are `STAR` for the sum (star + sky); `SKY` for sky alone; and `DIFF` for the difference. The latter is often produced in CCD photometry. Note that `SKY` data may still be useful in this case.

The position where sky was measured can be specified by its offsets from the star in both coordinates. Usually, these will be in two columns named SKYOFFSETRA and SKYOFFSETDEC, to identify the sky position used. These are convenient in most cases, as observers usually offset in just one coordinate. Often only one sky position is used, nearly always offset in the same direction. For users of alt-azimuth mountings, these labels can be replaced by SKYOFFSETALT and SKYOFFSETAZ; see Table I.18.

It can be foreseen that lazy observers will fill up these columns with zeroes. They should be warned that assuming the sky position to coincide with the star, combined with measuring the sky always on the same side of a star, can introduce systematic errors, because of the gradient of sky brightness with zenith distance. For faint stars this may not be negligible, particularly if the offset is always in declination, which tends to correlate strongly with zenith distance.

Some telescopes record the apparent position of each measurement. In such cases, it will be more convenient to use columns named SKYRA and SKYDEC instead of offsets (see Table I.18).

In clusters and variable-star fields, the sky may be measured in a common position for a group of stars. In this case, the observations of a group are delimited by putting `BEGINGROUP` in the OBJECT column at the start of the group, and `ENDGROUP` at the end. Then the sky position(s) should be given as absolute coordinates in a star-catalog MIDAS table file, identified there as a string beginning with the word `SKY`. The R.A. and Dec. for these sky positions can generally be determined accurately enough by reference to some star on finding charts, or by interpolation among the known positions of variable and comparison stars. Normally, these reference sky positions will simply be included in the program-star table files, with OBJECT names like `'SKY for NGC 7789'` or `'SKY position 1 for RU Lupi'`. This allows several distinct sky positions to be measured in the neighborhood of such a group.

Because MIDAS tables may in principle be sorted on any column, and because groups delimited by `BEGINGROUP` and `ENDGROUP` are inherently time-dependent, programs using such data must make sure the data are sorted in time sequence. This is not normally a problem, as time is the natural independent variable in such a list of observations. However, observers must be sure that the MJD_OBS column is correct for the `BEGINGROUP` and `ENDGROUP` pseudo-objects.

Bandpass and detector identification

The bandpass in which the measurement is made must be identified. The bandpass name is recorded in a column labelled BAND. (This name, combined with the information in the instrument file, is used to identify the detector in multichannel instruments.) Standard passband names should be used: `'V', 'B', 'U', 'URL', 'R', 'I', 'u', 'v', 'b', 'y', 'betaW',` etc. These should agree with the notation used for standard indices for the standard stars (see section I.2, above, which describes standard-star table files). Standard band names are also listed in subsection I.5.3, "Passbands".

For `DARK` measurements, a digit must be appended to indicate the detector number, if more than one detector is used: `DARK1, DARK2,` etc. If red leaks are measured for two

Column Label	Contents	Units	Variable Type	Format
SIGNAL	object measurement	exposure	R*4 real	Fw.d
OBJECT	object identification		C*32 string	A32
STARSKY	star/sky identification		C*4 string	A8
SKYOFFSETRA	sky measurement position	arcsec	R*4 real	F4.0
SKYOFFSETDEC	sky measurement position	arcsec	R*4 real	F4.0
BAND	passband identification		C*8 string	A8
MJD_OBS	start of integration	days	R*8 real	F12.5
EXPTIME	duration of integration	seconds	R*4 real	F8.3
COMMENT	comment field		C*32 string	A32

Table I.17: Basic column specifications for observational-data tables

or more passbands, they must be plainly marked; e.g., `'URL'`, `'BRL'`, etc. If "neutral" filters are used to measure nonlinearity, as is often done with pulse-counting systems, the appropriate suffix `'ND'` (for a single attenuator), or `'ND1'`, `'ND2'`, etc., should be appended to the BAND value.

Often a filter position is carried in the original data as a code. Such information must be decoded to a standard band name in the observation-table file. The decoding information is normally found in the instrumental table file.

Timing information

It is not always easy to identify "the" time of an observation. Some instruments record the time at which the observation began; some record the end of the measurement, or even the time at which the readout was recorded; very few record the middle of the exposure. The ESO Archive records the starting time of the observation, recorded in a FITS-header keyword named `MJD-OBS`. Because of problems in MIDAS with strings containing embedded minus signs, the column label used here is MJD_OBS. This quantity is the geocentric Julian Date at the start of the integration, minus 2400000.5 days. To retain adequate precision (1 second is required), this *must* be stored in double precision.

Note that this is geocentric, not heliocentric MJD. It is not suitable for use in computing phases of eclipsing binaries and the like without light-time corrections.

Having accepted a starting time as the basic timing datum, we must in every case have an integration time, even for data (like DC photometry) where this is not directly involved in calculating signal strength. In any case we would need this information for weighting purposes. Again the ESO Archive name is used as column label: EXPTIME.

Comment field

Comments are so common, and so useful, that a 32-byte COMMENT column should be considered part of the basic data, even though it may be blank for the majority of the data. This field may be used to append comments to a particular observation, such as `DOME IN THE WAY?`, `MOON ON MIRROR`, or `CONTRAIL`.

General comments, such as `Cirrus low in NW`, or `LUNCH BREAK`, should be stored as data with the OBJECT field set to `COMMENT`. Comments longer than 32 characters can be split into 32-byte pieces with the same time value. The time field MJD_OBS should always be filled in, but the rest can remain undefined.

I.6.2 Additional information

Several other data are useful, or even essential. Some of these, like temperature and humidity, are independent variables that are likely to affect sensitivity and spectral response. Atmospheric pressure, apart from very small effects, is directly proportional to the Rayleigh optical depth of the Earth's atmosphere, which is the most wavelength-dependent part of the extinction. Additional instrumental parameters, like the size of the measuring aperture or the high voltage used on a photomultiplier, are essential and must be recorded for each observation if they vary. Some systems have gain steps that must be recorded for every observation. Quasi-neutral attenuators may be used to calibrate nonlinearity; such information must be provided to the reduction program. The Geneva quality-control parameters [3] may be available; again, they should have separate columns in the table. Measurements from a seeing monitor may also be available.

Table I.18 describes these additional columns in detail.

Temperature and humidity data

The most temperature-sensitive parts of most photometric instruments are the filters, the detector, and the electronics. The likely temperature coefficients are of similar orders of magnitude: generally several tenths of a percent per degree. As temperatures usually vary by several degrees during a run, temperature effects are likely to exceed 0.01 magnitude.

Although the ESO Archive standard is to record temperatures in kelvins, this is often inconvenient. Temperature data may be recorded in Fahrenheit or Celsius, or in some scale with perfectly arbitrary units, such as the output of some uncalibrated thermistor sensor. Although Celsius or Kelvin degrees are a useful basis for judging whether the actual size of an apparent temperature coefficient is reasonable, the reduction program simply needs an independent variable to work with. Therefore, if the temperatures are not in kelvins, the appropriate units should be provided in the table file (if temperatures are in a column), or as a comment.

The temperatures of filters and detectors were discussed in previous sections dealing with instrumental parameters. If they are regulated, such data should be stored in Real*4 descriptors in the instrumental ".tbl" files (see above). If they are measured, they should be in data columns with the labels FILTTEMP and/or DETTEMP.

If temperatures are measured only occasionally, and not with every observation, the values should still be recorded in columns of the data table file. In this case, the temperature measurements are essentially asynchronous with the photometric measurements; then the OBJECT column should contain the word FILTTEMP or DETTEMP, as appropriate, and the temperature columns will contain the "undefined" value for actual observations.

For example, in the ESO Archive logfiles, `DOME TEMP` is recorded every 15 minutes. This should be used for FILTTEMP if the filters are at ambient temperature, and for DETTEMP for an uncooled detector.

If filter and detector temperatures are otherwise regulated or recorded, it may still be useful to record DOMETEMP in the data file as a Real*4 column. The times and temperatures can be stripped out of the logfiles and stored in the observational-data files.

Relative humidity is treated exactly the same way. The column label is RELHUM, and this should be put in the OBJECT column for sporadic readings.

Note that the times must be converted to Real*8 because of the MJD format required! As data in the ESO Archive logfiles are stored in *hh:mm:ss* form, they will have to be converted.

Pressure

Some observatories have accurate information available on atmospheric pressure. If it is routinely available for every measurement, it should go in a column labelled PRESSURE, with the SI unit kPa. Note that pressures read from aneroid (dial-type) barometers are not very accurate, and probably should not be used. Only absolute pressures should be recorded, *not* values "corrected to sea level".

Measuring aperture

The wings of a telescopic image are due to surface scattering caused by microroughness of polished optical surfaces, and to (usually) a much smaller extent to scattering by dirt on the optics. Contrary to popular mythology, atmospheric effects are quite negligible. These wings contain some tens of per cent of the total starlight, for typical surface quality. If the measuring aperture varies, a varying amount of starlight will be excluded. Furthermore, the excluded fraction is wavelength-dependent; so the transformation from instrumental to standard system changes. Therefore, the field stop, physical or synthesized, within which the measurements were made, should always be constant.

Unfortunately, sometimes it is necessary to combine measurements made with different field stops. Observers should realize that this really means different instruments; the differences are usually several per cent. Calibration data should be taken (i.e., several stars of different colors observed with all the apertures used) to determine the transformations between them.

Usually, the actual aperture sizes are only approximately known. In any case, the actual variation of the excluded energy fraction with radius is not accurately predictable. Therefore, it suffices to retain codes for apertures, rather than try to deal with them quantitatively. The code should be put in a column labelled DIAPHRAGM (the common term for the field stop).

Column Label	Contents	Units	Variable Type	Format
RAWMAG	object measurement	magnitudes	R*4 real	Fw.d
SKYRA	sky measurement position	degrees	R*4 real	F4.1
SKYDEC	sky measurement position	degrees	R*4 real	F4.1
SKYOFFSETALT	sky measurement position	arcsec	R*4 real	F4.0
SKYOFFSETAZ	sky measurement position	arcsec	R*4 real	F4.0
FILTTEMP	filter temperature	see text	R*4 real	F4.1
DETTEMP	detector temperature	see text	R*4 real	F4.1
DOMETEMP	dome temperature	see text	R*4 real	F4.1
RELHUM	relative humidity	per cent	R*4 real	F4.1
PRESSURE	atmos. pressure	kPa	R*4 real	F4.1
DIAPHRAGM	field stop code		C*4 string	A4
PMTVOLTS	PMT high voltage		C*4 string	A4
GENEVA_Q	Geneva Q parameter		R*4 real	E9.2
GENEVA_R	Geneva R parameter		R*4 real	E9.2
GENEVA_G	Geneva G parameter		R*4 real	E9.2
SEEING	seeing value	see text	C*4 string	A4
ESTERR	estimated error	see text	R*4 real	Fw.d

Table I.18: Other column specifications for observational-data tables

PMT Voltage

When photomultipliers are used, the spectral response depends in a complicated way (involving several different phenomena) on the electrode potentials in the first few stages. If the high voltage is changed to vary the gain, as is sometimes done in DC or CI work, the spectral response varies. This must be treated exactly like a variable measuring aperture: only data taken under constant conditions should be reduced together, but sometimes this can be done if adequate calibration data have been measured.

The column label is PMTVOLTS. Once again, because several complex effects are involved, the effects are not predictable, so there is no point in trying to use quantitative values. Besides, the voltage settings on many power supplies are not very accurate, nor can one read an analog voltmeter precisely enough to obtain useful numbers. The values in this column will be treated as strings rather than converted to numbers.

Gain steps

Here we consider only purely electrical gain changes that are guaranteed to be spectrally neutral. Examples are voltage-divider steps used to vary the gains of amplifiers; switch-selected capacitors used in charge-integration systems; and pre-scalers sometimes used to extend the dynamic range of counting systems.

The problem is complicated because there can be more than one set of variable gain steps (often both “coarse” and “fine” steps are provided), and because the actual gain values may not be well known. In the latter case, we would like to determine them, if adequate calibration data are available.

This is handled by having up to three columns containing the gain codes; these codes may actually be strings representing the approximate values in magnitudes, or other convenient labels, such as switch positions recorded in the raw data stream. The columns are named GAIN1, GAIN2, and GAIN3.

A character descriptor, **GAINTBL**, in the observational-data table gives the name of a MIDAS table file, whose columns are again labelled GAIN1, GAIN2, and (if needed) GAIN3 (see Table I.19). The reference column of this table contains the gain codes, and is labelled CODES.

Note that the gain values are multipliers or scale factors; they are *not* expressed in magnitudes. The true signal is the value in SIGNAL multiplied by the value(s) in the GAINn column(s). It is immaterial whether the largest or the smallest gain is assigned the value unity; the scale is perfectly arbitrary.

All the gains are of course pure numbers, and so have no units. It is the user's responsibility to make sure the gain columns in the gain-table and the data-table files match up correctly. To assist this matching process, *both* files may contain a character descriptor named **GAINNAMES**, containing up to three words (separated by commas) that name the three gain adjustments.

The gains should be determinable to extremely high accuracy by purely electrical measurements. In some cases, the measurements have not been done, and only nominal values are available, perhaps based on resistor tolerances. The uncertainties should be placed in the GAINERROR columns.

Column Label	Contents	Variable Type	Format
CODES	gain codes	C*4 string	A4
GAIN1	gain values for first gain adjustment	R*4 real	E9.4
GAIN2	gain values for second gain adjustment	R*4 real	E9.4
GAIN3	gain values for third gain adjustment	R*4 real	E9.4
GAINERROR1	uncertainty of first gain adjustment	R*4 real	E9.4
GAINERROR2	uncertainty of second gain adjustment	R*4 real	E9.4
GAINERROR3	uncertainty of third gain adjustment	R*4 real	E9.4

Table I.19: Columns of the gain-table file

If the gain steps are unknown, the **GAINTBL** descriptor should contain one space. If adequate calibration data are available, the reduction program will try to construct a gain table, with the default name gain.tbl.

This table is a little peculiar, in that it is unlikely that the same names will be used for the steps of the different adjustments. For example, the high gain steps might be coded by letters, but the fine steps by numbers. In such cases, only one column in each row will have values defined. This causes no problems, as only the combinations that have meaning should occur in the data.

Geneva parameters

Some instruments produce the Geneva Observatory quality parameters Q, R, and G [3]. These should occupy separate columns in the data table, with labels GENEVA_Q, GENEVA_R, and GENEVA_G.

Seeing

Sometimes seeing estimates are available, either from the observer at the eyepiece, or from a nearby seeing monitor. It might also be estimated from the core widths of images on CCD frames. Such values should go in a column headed SEEING.

While quantitative measures, such as FWHM in seconds of arc, are most useful, it may still be possible to obtain usable information from seeing expressed on some arbitrary scale. One should be careful not to mix the two types, or to intermix seeing estimated on different scales.

Error estimates

Data extracted from CCD frames may be accompanied by error estimates, which can be used in determining weights in the general solution. These should be in the same intensity units as the data in the SIGNAL column. The same format should be used as for the SIGNAL. The column label for estimated errors is ESTERR.

This column should not be used for ordinary photometric data (i.e., do not put estimates of "photon noise" here). It should only be used when an independent estimate of noise is available *in addition to* the information in the other columns.

If the data were expressed as magnitudes (column RAWMAG), any error estimates should also be in magnitudes.

Bibliography

[1] Aho, A.V., Kernighan, B.W., Weinberger, P.J. : 1988, *The AWK Programming Language,* Addison-Wesley, New York.

[2] Ažusienis and Straižys : 1966, Bull. Vilnius Astron. Obs., No. 17, 3.

[3] Bartholdi, P., Burnet, M., Rufener, F. : 1984, *A&AP* **134**, 290.

[4] Dougherty, D. : 1991, *sed & awk,* O'Reilly & Assoc., Sebastopol, CA.

[5] *ESO Archive Data Interface Requirements* Rev. 1.3, March, 1992.

[6] Heck, A. : 1991, *Astronomy, Space Sciences and Related Organizations of the World* (Publ. Spec. du C.D.S. no. 16) Observatoire Astronomique de Strasbourg, Strasbourg, France.

[7] Stetson, P.B. : 1989, in *Highlights of Astronomy* **8**, 635.

[8] Shao, C.-Y., Young, A.T. : 1965, *AJ* **70**, 726.

[9] Young, A.T. : 1967, *MN* **135**, 175.

[10] Young, A.T. : 1974, in *Methods of Experimental Physics,* vol. **12**, Part A, *Astrophysics: Optical and Infrared,* ed. Carleton, N., Academic, New York.

Appendix J

IRAC2 Online and Off-line Reductions

J.1 Introduction

In many ways, IR array data is similar to that taken with optical CCDs; however, there are a number of important differences that are mainly due to the large, variable sky and instrumental backgrounds that arise at these longer wavelengths. In extreme cases, the objects of interest can be several thousand times fainter than the sky background. This, together with the variable sky background and the unusual bias patterns of some readout methods, implies that objects of interest are generally not visible in a single image.

For the above mentioned reasons, fully automated reduction is difficult to achieve, and is unlikely to produce optimally reduced data even if achievable. It is thus inevitable that a more "hands on" approach be adopted for the data reduction, with the astronomer monitoring the quality of such things as flat fielding and sky subtraction much more carefully than is usually the case for optical CCD data.

In addition, if observers are to get the most out of their observations, some online processing of the data at the telescope is required. The IRAC2 context in MIDAS includes a number of commands that are most useful at the telescope.

The first part of this appendix concentrates on the data reduction that can be done at the telescope. The second part describes what is required for off-line reduction of IRAC2 data (and IR data in general), with a specific view towards use of standard MIDAS routines.

The IRAC2 context can be activated by the command `SET/CONTEXT IRAC2`. Together with the IRAC2 context, the CCDRED context is also activated. The CCDRED context contains many useful commands for image combining, mosaicing, etc.

J.2 Online Reduction

The first version of an online reduction system for IRAC2 has recently been developed. The principle aim of the online system is to enable observers to visualize their programme objects. The system has to be simple, fast and versatile. This section describes the online system and the commands used to create images.

J.2.1 The OST table

The heart of the IRAC2b online reduction system is the Observation Summary Table (OST). This table contains the essential details, such as the filter used, the lens used, the exposure etc., of every image taken with the camera. It is continuously updated as new exposures are made. For IRAC2B the table is called *irac2b_ost.tbl* and it is a regular MIDAS table. The contents of the table can be displayed or printed with the usual MIDAS table commands or with the `OBSLIST/IRAC2` and `OBSREP/IRAC2` commands described below. For a general description of OSTs and the `DO` context, see Chapter 15 in Volume B of the MIDAS users manual.

J.2.2 Online Commands

The online commands consists of commands for monitoring the data acquisition (via the OST table), commands for general image inspection, and commands for combining the incoming images. Additionally, a focus command helps the user to determine the best focus position of the telescope. Below follows a brief overview. For a detailed command description the reader is referred to the help files.

ACUTS/IRAC2	display an image with cuts mean-3*sig and mean+*upper**sig.
DCOMBB/IRAC2	sky subtract and combine dithered images
RCOMB/IRAC2	combine frames created with the task `DCOMB/IRAC2`
FOCUS/IRAC2	determine the best focus from a focus sequence
LAST/IRAC2	give very brief information on the most recent exposures
OBSLIST/IRAC2	print out the most relevant parts of the OST table
OBSREP/IRAC2	create a hardcopy of the most relevant parts of the OST table
QL/IRAC2	subtract one IRAC2 image from another and divide by DIT
SEEING/IRAC2	determine the seeing

Table J.1: IRAC2 On-line Commands

J.3 Off-line Reduction

The overall approach to IR data reduction is summarized in Figure J.3 (inspired by a similar diagram in T.M. Herbst's manual for the Calar Alto MAGIC cameras). We shall describe each of these steps below, and indicate the necessary MIDAS routines to achieve them.

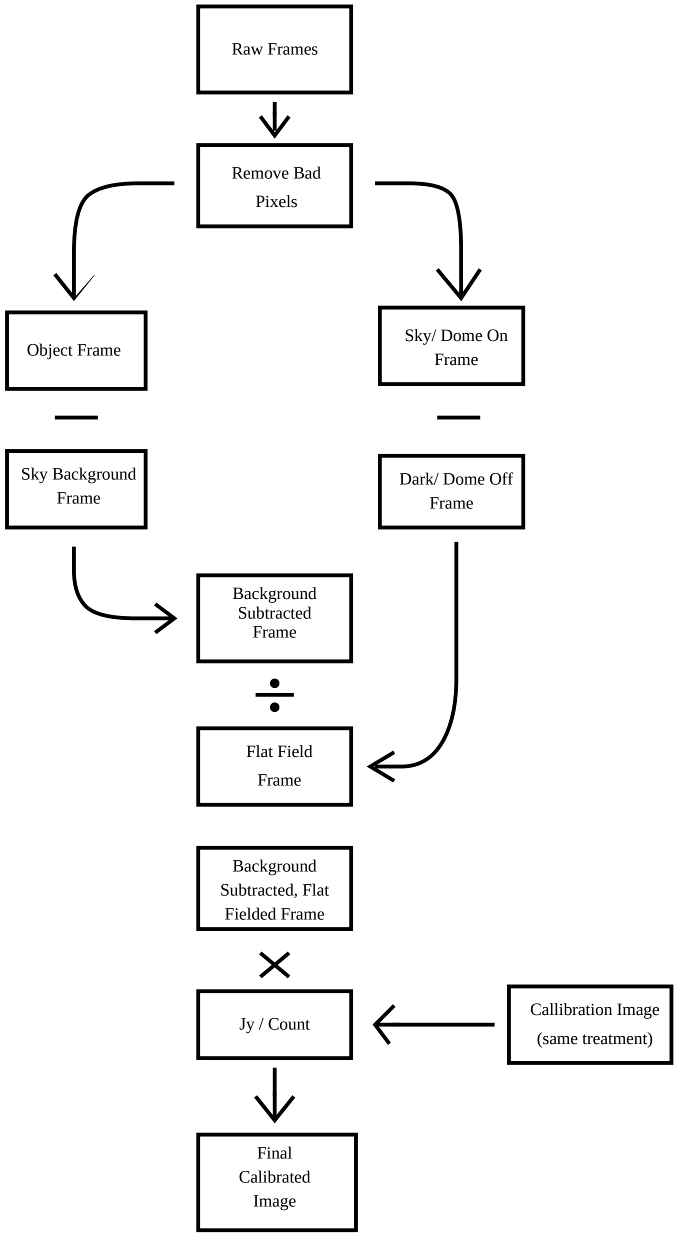

Figure J.1: IR Data Reduction

J.3.1 Bad Pixel Detection and Removal

Before most of the reduction process can be conducted, bad pixel values must be removed. This is usually achieved by flagging pixels above or below some threshold value as bad, and replacing their values by those of nearby, non-bad pixels. The LAMP ON or LAMP OFF flat field images are a good place to start to define your bad-pixel map. To determine the threshold values to use, you should examine the statistics of the image and set thresholds 5 to 10 standard deviations away from the mean in both positive and negative directions. The exact thresholds for determination of bad pixels will depend on the details of your flat field observations - filter, objective lens, integration time and lamp voltage - so you should experiment with different values until you obtain a satisfactory result.

Because IR arrays are sensitive to thermal cycling and to atmospheric contamination, the bad pixel lists change over time. Recent lists are available via the ESO WWW pages for comparison to bad pixel maps you generate yourself, but are unlikely to perfectly match the list derived from your own data.

In MIDAS, bad pixel detection is possible with the command `MASK/IRAC2`; the command `CMASK/IRAC2` can be used for bad pixel removal. Refer to the MIDAS manual for further documentation of these commands.

J.3.2 Construction of Flat Fields

Standard construction

There are two ways to produce flat fields for use with IR data. The standard method is to take 'LAMP ON' and 'LAMP OFF' exposures of the calibration spot on the telescope dome. The 'LAMP ON' exposure contains light from both the calibration lamp, which should provide the even illumination needed for the flat field, together with potentially uneven illumination from the telescope and instrument environments. To retain only the even calibration illumination, the "LAMP OFF" image must be subtracted from the "LAMP ON" image. This subtraction also removes the bias pattern from the flat field image.

The subtracted image should then be normalised for later use as a flat field.

In MIDAS, you may directly use `COMPUTE/IMAGE` to perform the subtraction. The image mean may then be calculated using `STATISTICS/IMAGE` or similar. The "LAMP on - LAMP OFF" image can then be divided by this value using `COMPUTE/IMAGE` for normalisation. Alternatively, the procedure MKFLAT/IRAC2 combines these actions. To improve the signal-to-noise ratio of the flat-field, a number of flat field images may be used together. These should be combined using `COMBINE/CCD` (see later).

Alternative construction

For a number of projects where very deep integrations and accurate flat fields are required, and the targets are point sources or much smaller than the frame size, the sky background itself can be used to calculate the flat field. When an object is dithered on the chip, so that it exposes pixels at different positions in different integrations, the resulting 'stack' of images may be combined to produce a very accurate flat field. Since a reference image, such as the "LAMP OFF" frame, is not subtracted, a bias frame of the same integration time must first be subtracted from all these images. They should then all be median combined, with suitable scaling introduced to match the medians of each image to account for the variability of the sky background level. The resulting image is then normalised to obtain the flat field.

To subtract the bias from the science images, use COMPUTE/IMAGE. Then use COMBINE/CCD to median all the frames together with multiplicative or additive scaling to match individual frame medians (you should examine the results of both options to determine which is working best for your data). Use STATISTICS/IMAGE to calculate the image median, and COMPUTE/IMAGE to divide by this to get the normalised flat field.

J.3.3 Sky Subtraction

The removal of the bright sky background from your IR images is one of the key steps in the data reduction process. Once bad pixels have been removed from all the astronomical frames, the object frames must have the matching sky background images subtracted from them. Depending on how rapidly the sky is varying, and on how long your integrations are, this process can be fairly simple or rather complicated.

The simplest sky subtraction is where you have separate sky and object frames taken a short time apart. These can be directly subtracted. However, the reference sky fields are seldom blank, with faint stars appearing at a number of points. It is thus advisable to take several sky frames at different positions. These may be median combined, making sure there is appropriate scaling to the same sky level, to eliminate the contribution of these objects.

More complicated sky subtraction schemes are possible. For a deep, dithered integration examining targets that are much smaller than the array size, a reference sky image may be created by median combining several neighbouring integrations, at different dither positions. The resulting reference frame can then be subtracted from the appropriate object frame. For a long dithered integration, the calculation of reference sky values can be a running process - the sky frame for a given object frame may be produced by mediannning together, for example, the 8 integrations nearest to it in time.

The resulting sky subtracted images should be examined to make sure that sky subtraction has been properly achieved. If the sky has significantly varied between the object and sky reference images, you may find large scale gradients or patterns in the resulting sky-subtracted image. This is an indication that you need to look carefully at the reference frames used and at matching the sky values in the relevant frames, which might be achieved by using a multiplicative or additive term. Considerably more complicated sky subtraction schemes are possible and may be required for certain observational projects (see Bunker et al 1995, MNRAS, 273, 513 for an example based on IRAC2b data).

In MIDAS, simple sky subtraction is achieved by using either the COMPUTE/IMAGE command or SSUB/IRAC2. COMBINE/CCD can be used for the median combination of frames. Combinations of these commands, and others, will be necessary for some of the more complicated sky subtraction schemes.

J.3.4 Flat Fielding

Once the flat field has been prepared in the manner described above, the astronomical images may be flat fielded simply by dividing by the flat field. This part of the process is similar to that for optical CCDs.

To do the flat fielding in MIDAS, use either `COMPUTE/CCD` to divide the sky subtracted images by the flat field frame, or use the command `FFIELD/IRAC2` to perform the same operation.

J.3.5 Combining Images

A number of the above steps require the combination of a number of IR images. This can be a sensitive matter because of the continuously varying sky background, and care must be taken to ensure that a suitable multiplicative scaling, or additive shift, is applied to match the sky background in all those frames being combined. Whether shifting or scaling is most appropriate unfortunately depends on the prevailing conditions, and you should experiment to find out which is best.

The command `COMBINE/CCD` is used to combine several images. The command itself has a number of options to use DO tables or catalogues, but the simplest version takes the form:

```
COMBINE/CCD type images output
```

where `type` is one of `BS` (for bias), `FF` (for flat field), `DK` (for dark), `SK` (for sky) and `OT` (for other) and is used to indicate what type of image you are combining; `images` is a comma separated list of images you wish to combine; `output` is the name of the resulting output file. Scaling and/or shifting options can be specified for each of these image types. These parameters can be checked using `SHOW/CCD type`, and be set using `SET/CCD type`, where `type`, using the codes listed above, specifies the type of image whose combination parameters you are interested in. An example is displayed below:

```
COMBINE/CCD FF kflat1,kflat2,kflat3 kflat
```

will combine the images `kflat1`, `kflat2` and `kflat3` into an output image `kflat` using the combination options specified for the FF image type.

J.3.6 Mosaicing

Mosaicing is the name applied to the process by which images of astronomical targets are combined in such a way that the positions of the objects are matched up. This can also lead to a final image larger than the input images if you are, for example, mapping out the IR emission in an extended target. Different reduction packages have several different methods for doing this, but all basically rely on the user specifying the positions of the objects to be matched up from one image to another, and/or specifying the relative shifts between each image that has to be combined.

A series of commands in MIDAS are used to perform mosaicing, all included in the `CCDRED` context. `CREATE/MOSAIC` is used first to create a master frame including all the subframes. Alignment of the frames is done using `ALIGN/MOSAIC`, and background levels are matched using `MATCH/MOSAIC` and `FIT/MOSAIC`. Objects to be used in matching the subimages together are selected using the `SHIFT/MOSAIC` command. Overall parameters for the mosaicing routines can be examined using `SHOW/CCD MO` and set using `SET/CCD MO`. More extensive documentation on the mosaicing routines is supplied in the chapter on CCD reductions, Chapter 3.

J.3.7 Further Off-line Analysis

Once the astronomical observations have gone through the above process, they are fully reduced and standard analysis packages, such as ROMAPHOT or DAOPHOT, may be used to calibrate and extract photometry etc. Such analysis techniques are detailed elsewhere.

J.4 Commands in the IRAC2 package

Table J.2 contains a brief summary of the IRAC2 commands and parameters. All commands are initialized by enabling the IRAC2 context with the MIDAS command `SET/CONTEXT IRAC2`. The CCDRED context is also made available when the IRAC2 context is initialized. Consult Chapter 3for details of the CCDRED context.

CMASK/IRAC2	ffield clnffield lthrshold,hthrshold [dispflag]
	create bad pixel mask from flat field
MASK/IRAC2	inframe outframe
	replace bad pixels by neighbouring good ones
MKFLAT/IRAC2	lamp_on lamp_off flat_field
	create a flat field
SSUBTR/IRAC2	obj_frame sky_frame out_frame
	subtract a sky image from a science image
FFIELD/IRAC2	obj_frame ff_frame out_frame
	flat field an image
ACUTS/IRAC2	[image] [load] [plot] [upper]
	display an image with cuts setting
DCOMB/IRAC2	[select] [seqname] [accsky] [align] output [trim] [tag]
	sky subtract and combine dithered images
FOCUS/IRAC2	seqnum [focout] [create]
	determine best focus from a focus sequence
OBSLST/IRAC2	[start] [end]
	lists a subsection of the IRAC2B OST
OBSREP/IRAC2	start end
	print out a subsection of the IRAC2B OST
LAST/IRAC2	[num]
	gives brief information on recent exposures
QL/IRAC2	image1 image2 [outimage]
	subtracts one IRAC2 image from another
RCOMB/IRAC2	select [align] output
	combine frames created with DCOMB/IRAC2
SEEING/IRAC2	determine the seeing

Table J.2: IRAC2 On-line and Off-line commands

Appendix K

Testing CCD Performance

K.1 Introduction

This chapter describes CCD test package that can be used to check the performance of the CCD detectors used.

In order to ensure the quality of the data delivered by the CCDs on La Silla, ESO run a programme to monitor all CCDs available at the Observatory. In this CCD monitoring programme, for each CCD test for each CCD data is collected on regular intervals. A standard data to check the performance looks like the following one:

- 9 bias frames;
- 16 pairs of flat fields (both of each pair have the same integration time) using a stable light source and with exposure levels ranging from just above bias to digital saturation;
- 9 low-count-level (of order a few hundred electrons per pixel) flat-fields with stable light source;
- one flat-field exposure obtained with 64 rapid shutter cycles;
- 3 30-minute dark images;
- the time taken to read out and display an image.

The quality of the data collected is checked using a number of commands and procedures available in the MIDAS CCDTEST context. Although the composition of the calibration data of the user is most not identical to ESO's test data set, the same CCD commands can still be executed to check the quality of the user's calibration data.

K.1.1 Test Commands

The quality control can be done by six test commands in the MIDAS CCDTEST context. The commands are called `TESTX/CCD` where `X` can be: `B` for the bias, `D` for dark, `F` for flat, `T` for transfer, `S` for shutter, and `C` for charge transfer efficiency. All output (*i.e.* ASCII and MIDAS tables, postscript files of graphics and display output) will be put in the users working directory. In addition, the MIDAS logfile will contain a complete log of the results. A description of the commands and the output produced follows below.

`TESTBA/CCD`

The command does a series of tests on a catalogue of bias frames. Since this commands produce the bias offset that is needed in most of the other test commands (*e.g.* `TESTF/CCD` and `TESTD/CCD`), it should be the first command to be executed. The whole test is split in five smaller tests, commands `TESTB1/CCD` to `TESTB5/CCD` that do the following:

1. Test B1: Creation of the combined bias frame. The result is loaded onto the display.
2. Test B2: Find the hot pixels. The combined bias frame is median filtered (using the parameter 'fil_siz') and subtracted from the original. A plot is generated showing the position of the hot pixels and the affected columns. Hot pixels will only be searched for within the requested area and above the intensity level of (mean + 0.25*sigma + 5.), where mean is the mean int level, simga is the standard deviation.
3. Test B3: Inspection of single frames. From the combined bias frame rows and columns are averaged and plotted.
4. Test B4: The last frame in the catalogue is first corrected for hot pixels and then rebinned. A histogram of this rebinned frame is made.
5. Test B5: For each input frame in the catalogue determine the mean bias and standard deviation after hot pixel correction (using the hot pixel table determined in test B2), box averaging and median filtering. The keyword BIASMEAN and BIASSIGM are filled with the average values for the mean and sigma.

To avoid unnecessary computations the command checks for the presence of the combined bias frame and the median filtered hot pixel frame and does not recompute these frames if they are already present. In the case of subtests the command will (re)created these output frames.

The complete `TESTBA/CCD` command produces the following:

- A combined bias frame.
- A map of hot pixels in bias frames (obtained from a median stack of the raw bias frames);
- An ASCII and a MIDAS table containing the hot pixels;
- Plots of row and column averages of the mean bias;
- The mean bias level and standard deviations after hot pixel correction median filtering.

In order to make this test useful a minimum of 5 bias frames is recommended. The mean bias level and the standard deviation of the mean are stored in the keywords `BIASMEAN` and `BIASSIGM`.

`TESTFA/CCD`

The command does a series of tests on a catalogue of low count flats. The whole is split in two smaller tests, commands `TESTF1/CCD` to `TESTF2/CCD` that do the following:

1. Test F1: Creation of the combined flat frame, using only those flat frames in the input catalogue that have exposure times falling within the allowed range. The combined flat is corrected for the bias offset. The bias offset is taken from the keyword BIASMEAN filled by the command TESTB/CCD. The combined flat is loaded on the display.
2. Test F2: Thereafter all pixels in the stacked master flat frame that show values less than thresh times the median counts in the frame are listed. Only pixels within the area contained in 'area' are considered and repetitions of cold pixels in the increasing y coordinate are not listed.

The complete `TESTFA/CCD` command produces the following:

- A combined low count flat frame corrected for the bias offset;
- An ASCII and MIDAS table containing traps and other defects in the stacked master flat frame that show values less than N times the median counts in the frame. Only pixels within the input area are considered and repetitions of cold pixels in the increasing y coordinate are not listed.

The combined low count flat field is corrected for the mean bias offset, stored in the keyword `BIASMEAN`, filled by the command `TESTB/CCD`. The user can also supply this keyword with the name of the combined bias frame, also produced by `TESTB/CCD`. In that case this frame will be used for the bias correction.

`TESTTA/CCD`

The command does a series of tests on a catalogue of flat frames. The flat fields in the catalogue should be grouped in pairs with the same exposure time. Most ideally, one should be of two groups of the order of 8 frames each - the first with increasing exposure times and the second with decreasing exposure times, interleaved with those of the first group. In this way, trends observed in the CCD response that are probably caused by the effect of temperature variations on the light source can be rejected.

The command requires a value for the mean bias level and the standard deviation in the keywords `BIASMEAN` and `BIASSIGM` to be filled and hence should be executed after the command `TESTB/CCD`. If no value or the value zero is found no bias offset will be subtracted.

The whole test is split in three smaller tests, commands `TESTT1/CCD` to `TESTT3/CCD` that do the following:

1. Test T1: Creation of the transfer/linearity table. The table will contain 5 columns: column 1 for the exposure time of the first of each sets (frames 1) (label :Exp_tim1); column 2 the exposure time of the second frames (frames 2) (:Exp_tim2); column 3 the median pixel intensities over the selected frame sections in frames 1 (:Med_cnt1); column 4 the median pixel intensities over the selected area in frames 2 (:Med_cnt2) ; column 5 the variance of the difference of the frames 1 and 2 (:Variance).

2. Test T2: Determination of linearity curve and the shutter error and the shutter offset. Entries in the linearity table not fullfilling the selection criteria will now be selected out. From the remaining entries in the table a linear fit is done to determine the linearity curve for frames 1 and 2 and the shutter error. Using the linearity data the fractional count rates are plotted against the median counts, applying a shutter offset in the measured exposure times. The real shutter offset is determined by the value for which the fit give the minimum mean residual.

3. Test T3: Determination of the transfer curve. From the selected entries of the table a linear regression analysis is done to determine the analog to digital conversion factor and the electronic readout noise. The readout noise is determined by the inverse of the slope between the median and the variance multiplied by the sigma of the bias (determined by TESTBA/CCD OR TESTB5/CCD and stored in keyword BIASSIGM).

The complete command produces:

- A table containing the exposure time of the first of each sets (frames 1); the exposure time of the second frames (frames 2); the median pixel intensities over the selected frame sections in frames 1; the median pixel intensities over the selected area in frames 2; the variance of the difference of the frames 1 and 2
- Two linearity curves, expressed as count rate versus true exposure time. The mechanical shutter delay is determined either by linear extrapolation of the normal linearity curve (observed counts versus exposure time), thus assuming the response of the CCD is linear, or by adjusting the exposure times such that the count rate curve is closest to a straight line, thus allowing for a first-order nonlinearity in the response of the CCD.
- A transfer curve (Janesick et al., 1987) generated for any window onto the images obtained.

The linearity and the transfer curves may be generated for any section of the images.

`TESTD/CCD`

The command does a series of tests on a catalogue of dark frames and produces:

- An estimate of the electron analogue-to-ADU conversion factor;
- A map of dark current across the CCD.

The command uses the bias offset that is expected in the keyword `BIASMEAN` which is produced by the command `TESTB/CCD`. Alternatively, the user can store the name of the combined bias frame created by the command `TESTB/CCD` in the keyword `BIASMEAN`.

`TESTS/CCD`

The command determines the shutter error distribution. The error distribution is computed as follows. If in_frm1 has a total reported exposure time of $t1$ seconds, and the shutter is opened and closed n_exp times (including the beginning and end of the exposure) and $img2$ has a total exposure time of $t2$ seconds, and the shutter is only opened and closed once, then the final shutter error frame out_frm is determined by:

$$out\_frm = (in\_frm2 * t1 - in\_frm2 * t2)/(in\_frm1 - N * in\_frm2). \qquad \text{(K.1)}$$

An image and a contour plot of the error frame are produced.

`TESTC/CCD`

This command produces an estimate of the bulk charge transfer efficiency in the horizontal (HCTE) and vertical direction (VCTE) (by the EPER method (Janesick et al., 1987)). For the HCTE the command first averages the rows given as the second parameter. The command uses the number of image pixels, the last image pixel and the first bias overscan pixel is obtained and computes the HCTE according to the formula:

$$HCTE = 1 - bc/ic * ni, \qquad \text{(K.2)}$$

where: bc are the counts above the bias level in the first overscan pixel in a row; ic are the count above the bias level in the last image pixel in a row; ni are the number of image pixels in a row.

The values for the bias offset is extracted from the keyword `BIASMEAN`, and is computed by the command `TESTB/CCD`. To determine the image section of the CCD and the overscan regions one can use the commands `READ/IMAGE`, `PLOT/COLUMN` and `PLOT/ROW`.

Note that the last column of a row is often slightly brighter than the rest of the row (because the pixel is slightly larger). The vertical charge transfer efficiency is computed in a similar way.

K.2 Commands in the CCD test package

Below follows a brief summary of the CCD test commands and parameters are included for reference. The context is enabled by the command `SET/CONTEXT CCDTEST`. Enabling the CCDTEST will also enable the `CCDRED` context that is needed to do some of the combining of the images in the various input catalogues.

TESTBA/CCD	in_cat [out_id] [meth] [rows] [colums] [area] [fil_siz] [dec_fac]
	do a series of tests on a catalogue of bias frames
TESTC/CCD	in_frm [rows] x_pix [colums] y_pix
	compute horizontal and vertical charge transfer efficiency
TESTD/CCD	in_cat [out_id] [dec_fac]
	do a test on a catalogue of dark current frames
TESTFA/CCD	in_cat [out_id] [meth] [area] [exp_ran] [theshold]
	do a series of tests on a catalogue of low count flat frames
TESTS/CCD	in_frm1 in_frm2 [out_frm] n_exp [dec_fac]
	find the shutter error distribution
TESTTA/CCD	in_cat [out_id] [area] [select]
	do linearity and transfer tests on a catalogue of flat frames

Table K.1: CCDTEST command

Appendix L

Multi-Object Spectroscopy

Note

The default values for any parameters, input data, etc. are given in (**bold font**) *and keywords are marked by* `KEYWORD`.

L.1 Introduction

The MOS context is written for the reduction of multi-object spectra obtained with numerous slitlets. Long-slit spectra are considered a special case of multi-object spectra and therefore not treated in a separate chapter. The MOS package is meant especially for the reduction of FORS data but may also be used for other MOS data.

It is assumed that the basic correction for bias, dark, and overscan are performed with standard MIDAS commands (maybe also using COMBINE/LONG to average several frames of the same type). There are no special MOS averaging commands.

During the whole description we assume that the detector is a CCD with the rows along the dispersion direction and the columns along the slit.

A demonstration of the package can be obtained with the command TUTORIAL/MOS, which has an automatic and an interactive mode and also allows to look at selected parts of the package. In order to create all necessary test data you should run it once automatically.

L.2 Location of slitlets and flat-field correction

The very first step after correcting bias, dark, and overscan is to find the edges of the slitlets. This is done by the command LOCATE/MOS. This command locates the slitlets in an MOS flat-field frame by searching for the maximum (normalized) gradient in a trace perpendicular to the direction of dispersion. Position and width of the trace are given by `SCAN_POS` (**0**). `FLATLIM(1)` (**0**) gives the minimum normalized gradient that must be exceeded, after median filtering the scan with a median of width `FLATLIM(2)` (**0**) and discarding scan values below `FLATLIM(3)` (**0**). The result is written to the output table `MOS`.tbl (which is used by most MOS commands) and the number of detected slitlets is written to `NSLIT` (**0**). The programs allow at most 100 slitlets. If the algorithm does not find any slitlets the chosen threshold (`FLATLIM(1)` (**0**)) may either be too high (above the intensity of the flats in the center of the frame) or too low (below bias value). Also the width (`FLATLIM(2)` (**0**)) may be chosen to high or too small. Typical values are between 0.1 and 0.2 and 3 and 5, respectively.

It is also possible to define the slitlets interactively with `DEFINE/SLIT`. Here you first initialize the table `MOS`.tbl (**mos**) and then enter the limits with the cursor on the displayed flat field frame. This comand also allows an easy definition of the `MOS` table for long-slit data.

With `LOCATE/MOS` the offsets in dispersion direction between the slitlets will be read from the header of the flat-field frame for FORS data and stored in the table `MOS`.tbl (**mos**) in column `:xoffset`. For other data or `DEFINE/SLIT` you will have to determine the offsets yourself using the command `OFFSET/MOS` on a wavelength calibration frame (see below).

As spectroscopic flat-fields normally exhibit the spectral characteristic of the lamp that was used to produce them you have to take out this characteristic in order to correct the CCD sensitivity variation and keep the original flux distribution. This is done with the command `NORM/MOS`. It takes an averaged flat frame and the slit limits stored in the table `MOS`.tbl. There are two methods provided for the normalization (`NORMMET` (**poly**)): In case `NORMMET=poly` it averages separately for each slitlet the rows, fits a polynomial of chosen degree (`FFORD` (**3**)) to the flux distribution obtained this way and divides each row in the slitlet by this polynomial. In case `NORMMET=median` it averages separately for each slitlet the rows, smooths with a median filter of `FFORD` pixels width and divides each row in the slitlet by the filtered average. You may also perform the flat correction at the same step using the command `FLAT/MOS`.

L.3 Wavelength Calibration

L.3.1 Detection of Arc lines

At first you have to find the calibration lines in the wavelength calibration frame belonging to your object. This is achieved by `SEARCH/MOS` which detects all lines separately for each slitlet and centers them either with center of gravity or with a gaussian fit. The search threshold and window are stored in `SEAPAR` (**200,5**). The positions of the lines, the CCD rows and the slitlets where the lines have been detected are stored in the table `LINPOS`.tbl (**linpos**). First the median of (`YBIN(2)` (**3**)) rows in the middle of each slitlet is calculated to search in this average spectrum for lines. Then in all rows the detected lines are centered using the line list obtained by the search algorithm. In this way we want to avoid unstable dispersion solutions which might arise from different line lists for different rows of the slitlet. In order to speed up the search you may decide not to take every row, but only some (`YBIN(1)` (**3**)) and for the detection of weak lines you may bin several row together (`YBIN(2)` (**3**)). In the case of the Gaussian centering method, blends could prevent the proper convergence of the non-linear fitting algorithm. The accuracy of the centering is significantly improved, if the peak flux of the Gaussian is hold fixed to be the meassured maximum value in the central pixel. Free parameters are σ and the center of the line.

L.3.2 Offsets between slitlets

If you have used `DEFINE/SLIT` there are no offsets copied to $\boxed{\texttt{MOS}}$.tbl automatically and you have to determine the offsets yourself using the command `OFFSET/MOS` on a wavelength calibration frame. This command will take the calibration spectra of the different slitlets and search for arc lines with the parameters given by $\boxed{\texttt{SEAPAR}}$ (**200,5**). Correlating these line lists with that of the first slitlet gives the offsets relative to the first slitlet. The resulting offsets may be wrong if there are not enough or too many lines to get unambiguous correlation results. The safest way is to use this command after you are satisfied with the results of `SEARCH/MOS` **and** you have stored the parameters for threshold and width to $\boxed{\texttt{SEAPAR}}$. In this case you will **NOT** have **the offsets from the center of the CCD but relative to the position of the first slitlet!** The changing image scale over the detector can make the method unappropiate. Therefore it is strongly recommended to copy the offsets from descriptor data to the $\boxed{\texttt{MOS}}$.tbl table, but not from `OFFSET/MOS` – if possible.

Note

Before starting the calibration, a carefully selected line catalog must be prepared. Reject any line that is below the detection limit and any line which is blended!

L.3.3 Fitting the dispersion curve

Now you have several possibilities to perform your wavelength calibration. At first there are three different modes to identify the calibration lines ($\boxed{\texttt{WLCMET(1)}}$ (**F**)):

Identify: You identify at least 2 arc lines in one slitlet with the command `IDENTIFY/MOS`. The command `CALIBRATE/MOS` then performs a first fit for the CCD row with the identified lines.

Linear: You know the central wavelength and the mean linear dispersion of your grism. These values are used as first fit of the first selected CCD row. You have to correct the value of this central wavelength if you used the command `OFFSET/MOS` to determine the offsets of your slitlets:

Example: Your reference slitlet has an offset of -100 relative to the center of the CCD in x-direction and you have a mean dispersion of 2 Å/pixel and a central wavelength of 5500 Å. This central wavelength will always lie at the x-position of the respective slitlet, which is in this case -100 pixels (i.e. -200 Å) from the center of the CCD. This means that you have a wavelength of 5700 Å at the center of the CCD within the reference slitlet. This wavelength should be used as `wcenter`, since the program assumes that `:xoffset = 0` means that the slitlet is at the center of the CCD in x-direction.

Recall: Method Linear is performed in the first slitlet. The dispersion coefficients of the this slitlet is recalled to calibrate the remaining slitlets. This identification is more stable in this case than for method Linear and the convergence of the fit is reached faster.

The first fit is used to identify as many lines as possible in the corresponding CCD row by comparing the fitted line positions to wavelength catalogue `LINECAT`.tbl (**hear**). For the identified lines a polynomial fit of chosen order is performed (using Legendre or Chebyshev polynomials - the type being selected with the keyword `POLTYP` (**CHEBYSHEV**)). The line identification criterion will identify a computed wavelength (λ_c) with a catalog wavelength (λ_{cat}), if the residual

$$\delta\lambda = |\lambda_c - \lambda_{cat}|$$

is small compared to the distances to the next neighbours (in the arc spectrum as well as in the catalog):

$$\delta\lambda < min(\delta\lambda_c, \delta\lambda_{cat}) * \alpha$$

where $\delta\lambda_{cat}$ ($\delta\lambda_c$) is the distance to the next neighbour in the catalog (arc spectrum) and α is the tolerance parameter ($0 \ldots 0.5$) given by `ALPHA` (**0.2**). The automatic line identification is repeated with this polynomial fit in order to identify additional lines to further improve the dispersion curve.

Note

For very low dispersion spektroscopy one would expect that a linear guess will cause line mismatchs at the edge of the detector. One can avoid this, if more then two lines are identified with method Identify.

After the polynomial fit the residual of each line are checked and the line is thrown out, if it the residual exceeds the tolerance parameter `TOL` (**2**) (> 0 – in pixels; < 0 – in units of the wavelength). One of three fitting methods can be selected by keyword `WLCMET(2)` (**C**):

Constant fit in spatial direction: the dispersion coeficiants are constant for the whole slitlet. This method is typically appropriate for small slits.

Variable fit in spatial direction: bad lines are thrown out. Dispersion coeficiants are calculated for any row. The dispersion relation of the first fitted row is used as estimate for all following rows. A large number of arc-lines is required for this method. If there are only a few lines identified at the edge of the detector, small oscillations at the edge of the detector may occur in spatial direction.

Two dimensional fit over the slitlet: A two-dimensional fit is performed in spatial and dispersion direction over the slitlet. In spatial direction a "normal polynom" is fitted but a polynom as specified in keyword `POLTYP` in dispersion direction. The dispersion coeficiants may smoothly evolve over the slitlet. This method is the most accurate for most applications, although the resulting residuals are typically larger than for a variable fit.

The iteration is repeated until a stable solution is obtained (and the minimum number of iterations `WLCNITER(1)` (**3**) is exceeded) or the maximum number of cycles (`WLCNITER(2)`

(**20**)) is reached. The resulting dispersion coefficients are stored in table LINFIT.tbl (**coerbr**), together with the r.m.s. error of the fit, the slitlet and the y-coordinates (world and pixel coordinates). Also a plot option for the resulting residuals (PLOTC, (**N**)) and various degrees of display (DISP, (**0**)) are available.

After fitting all rows of the respective slitlet with polynomials of the chosen order the program performs at last a linear fit to get the central wavelength and the mean linear dispersion necessary to derive a starting wavelength for the next slitlet from its known offset (modes Linear/Recall). In the mode Ident it tries to match the manually identified lines in the next slitlet, using the known offsets and a maximum allowed shifting tolerance stored in SHIFTTOL (**10**). Rows where no fit could be achieved are stored in the table LINFIT.tbl with the slit number -1.

Any selection of slitlets made in the table LINPOS.tbl will be taken into account, but all selections of the table MOS.tbl will be ignored. If you want those to be respected, too, redo the search for the wavelength calibration lines with the chosen selection in MOS.tbl.

After the wavelength calibration you may rebin your frame two-dimensionally to constant wavelength steps with `REBIN/MOS`. Point sources are normally wavelength calibrated after extraction (see below).

L.4 Definition of objects and sky subtraction

`DEFINE/MOS` helps you to localize your objects and sky regions and by default works automatically. It averages XBIN (**20**) columns around the position SCAN_POS(1) (**0 = center of frame**) (in world-coordinates!). In the target frame the program will detect objects above the threshold (THRESH, **-0.04**, see below) relative to the local background within the search window WIND (**5**) and fits a gaussian to the spatial profile of any detected object. The threshold may be given in absolute (> 0.0) or relative (< 0.0) numbers. It may be advisable to do at least a rough sky subtraction ahead of this command to facilitate the detection of the objects. In this case you have to use an absolute threshold for the detection of the object spectra afterwards. One may also think about rebinning the object frame to constant wavelength steps because then the search could be done in the same wavelength region for all slitlets. The limits of the objects are defined at the position where the gaussian fit has reached the detection limit INT_LIM (**0.001**). A safety margin of 3 pixels is taken on both sides of each object where no sky is automatically defined (can be overridden manually later) and the remaining part of the slitlets is taken as sky region. The results are stored in WINDOWS.tbl (**window**) and can be displayed in the overlay channel display and/or the graphics window. If you are not satisfied with the results you can change the windows interactively. You may also choose the interactive mode from the very beginning with `DEFINE/WIND`. Then no automatic search is performed; instead you enter the the objects and sky regions for each slitlet by keyboard input. By default the sky region is defined as the complete slitlet.

The sky fit methods (SKYMET) available for `SKYFIT/MOS` are a simple median along CCD columns within each slitlet (`skymet=median`) and a more appropriate polynomial fit along the columns (`skymet=polynomial`), respectively. These two methods use only rows marked as sky regions in the table WINDOWS.tbl (**window**) to fit the sky background. With `skymet=nowindows`, however, the table WINDOWS.tbl is ignored and the sky is determined as a simple median over the full slitlet. The limits of the slitlets are taken in this case from MOS.tbl. This mode may be useful for a preliminary sky determination, before the object positions are known. If no sky regions are marked in some slitlet, the input frame is just copied to the sky frame for this slitlet. In this way, after sky subtraction, the slitlet contains only zeros thereby marking that the sky background is unknown for this slitlet. The keyword SKYMET contains the order of the polynom fit or the width of the median filtering, respectively. If a polynom fit is performed the cosmics must be rejected. SKYFIT/MOS rejects (but not replaces) pixels that exceed a given limit before the fit is performed. Read out noise, gain and the detection limit (in units of σ) must be given by keywords SKYMET(1),SKYMET(2),REJTHRES.

L.5 Extraction of objects

The object extraction EXTRACT/MOS is done with an optimum extraction scheme using different weights for the individual rows (following Horne, 1986, PASP 98, 609). With this command also the sky is subtracted. In addition the sky frame is needed to compute the optimum weights. The object postions are taken from the table WINDOWS.tbl. Cosmics are removed by assigning weight zero to the affected pixels. The procedure is iterative. If the number of iterations EXTPAR(2) is set to a negative number, a spectrum computed with equal weights is returned. The extracted spectra are stored in a 2-D frame, each line corresponding to one extracted object. In addition, the errors for the extracted spectra are returned, one line per spectrum, in the upper part of the same frame.

The command REB1D/MOS splits the 2-dimensional frame (produced by EXTRACT/MOS) up into 1-dimensional frames, one for each object. A root name is given and the rebinned frames are named by appending their row number to the root name. The extracted frames are rebinned to constant wavelength steps using the dispersion relation which has been obtained near the center row of the extracted objects.

If you prefer to avoid the resampling noise you can apply the dispersion relation to each pixel of an extracted frame (table option) without rebinning to constant wavelength steps with the command APPLY/MOS. This command produces for each extracted object a table with the columns :FLUX, :FLUX_ERROR, :WAVELENGTH and :BIN.

Context MOS	
APPLY/MOS	[EXTOBJEC] [LINFIT] [CALOBJEC]
CALIBRATE/MOS	[TOL] [WLCORD(1)] [WLCMET] [WCENTER,AVDISP] [PLOTC] [DISP]
DEFINE/MOS	[OBJ] [MOS] [WINDOWS] [THRESH] [WIND] [XBIN] [SCAN_POS(1)] [plotopt]
DEFINE/SLIT	mode [slit] [low,upp] [offset] [MOS]
DEFINE/WIND	[mode] [sequence] [obj] [sky] [MOS] [WINDOWS]
EXTRACT/MOS	[OBJ] [SKYFRAME] [WINDOWS] [EXTOBJEC] [EXTPAR] [CCDPAR] [REJTHRESH]
FLAT/MOS	[OBJ] [objf] [FLAT] [MOS] [NORMFLAT] [FFORD] [NORMMET]
HELP/MOS	[keyword]
IDENTIFY/MOS	[WLC] [YSTART] [LINPOS] [TOLWIND]
INIT/MOS	[session]
LINPLOT/MOS	[mode] [slit] [LINPOS]
LOCATE/MOS	[FLAT] [FLATLIM] [MOS] [SCAN_POS] [XBIN] [nml]
NORM/MOS	[FLAT] [MOS] [NORMFLAT] [FFORD] [NORMMET]
OFFSET/MOS	[MOS] [WLC] [SEAPAR]
PLOT/LOCATE	[MOS]
REBIN/MOS	in out [REBSTRT,REBEND,REBSTP] [REBMET] [LINFIT] [MOS] [WLCORD(2)]
REB1D/MOS	in root [REBSTRT,REBEND,REBSTP] [REBMET] [LINFIT]
RESPLO/MOS	[LINPOS] [ISLIT]
SAVE/MOS	[session]
SEARCH/MOS	[WLC] [SEAPAR] [YBIN] [CENTMET] [MOS] [LINPOS]
SHOW/MOS	[param]
SKYEX/MOS	[OBJ] [MOS] [WINDOWS] [LINFIT] [SKYPAR] [SKYMET] [EXTPAR]
SKYFIT/MOS	[OBJ] [MOS] [WINDOWS] [SKYFRAME] [SKYPAR] [SKYMET] [CCDPAR(1),CCDPAR(2),REJTHRES(1)]
TUTORIAL/MOS	[mode] [action]
WLDEF/MOS	[WLC] [SEAPAR] [YBIN] [CENTMET] [TOL] [WLCORD(1)] [WLCMET] [PLOTC]

Table L.1: Commands of the context MOS

L.6 Data Structures

MIDAS tables and keywords are used to store most the information created and/or used by the MOS context. The tables `MOS`, `LINPOS`, `LINFIT`, and `WINDOWS` are created by the MOS commands; the table `LINECAT` is available in the **MIDAS** data base. A short description of the tables is given below; a detailed description of their contents can be found in Table L.6.

`MOS`**.tbl** is created by `LOCATE/MOS` and used during the whole reduction process. It contains the limits of the slitlets.

`LINPOS`**.tbl** is created by `SEARCH/MOS` and updated by `CALIBRATE/MOS`. It contains the pixel-positions of the calibration lines, the row and the slitlet they belong to, their identifications and their intensities.

`LINFIT`**.tbl** is created by `CALIBRATE/MOS` and used by `REBIN/MOS`. It contains the fit coefficients for the dispersion relations of each row scanned by `SEARCH/MOS`.

`WINDOWS`**.tbl** is created by `DEFINE/MOS` and used by `FITSKY/MOS` and by `EXTRACT/MOS`. It contains the positions of the objects and the sky regions.

`LINECAT`**.tbl** contains the wavelength calibration lines. There will be several line-catalogs available in the instrument related system area, adapted for the use with the FORS grisms.

Table MOS		
Label	**Unit**	**Description**
SLIT	–	sequential number of slitlet
YSTART	PIXEL	first row of slitlet
YEND	PIXEL	last row of slitlet
XOFFSET	PIXEL	offset of slitlet

Table L.2: Table MOS

Table LINPOS		
Label	**Unit**	**Description**
X	PIXEL	x-position of calibration line
Y	–	world coordinate of row in which the line was found
PEAK	–	Peak intensity of calibration line
SLIT	–	number of slitlet in which the line was found
WAVE	Angstroem	identification of the line (Å)
WAVEC	Angstroem	fitted wavelength (Å)
RESIDUAL	Angstroem	WAVEC-IDENT (Å)
REJECT	–	Rejection code (-5: line has been rejected due to too large residual)

Table L.3: Line positions table

Table LINECAT		
Label	**Unit**	**Description**
WAVE	ANGSTROEM	wavelength of calibration lines

Table L.4: Line catalog table

L.7 MOS Cookbook - A typical session

L.7.1 Starting the whole thing

Before you start the MOS context you should have done the following preparations:

- average your bias frames
- average your dark frames
- average all flat-fields of the same setup

Table LINFIT		
Label	**Unit**	**Description**
SLIT	–	number of slitlet in which the dispersion relation was determined
ROW	PIXEL	row in which the dispersion relation was determined
Y	–	world coordinate of ROW
RMS	Angstroem	r.m.s. error of dispersion relation (Å)
COEF_i	–	dispersion coefficients

Table L.5: Line fit table

Table WINDOWS		
Label	**Unit**	**Description**
OBJ_SLIT	–	number of slitlet in which the object was found
OBJ_STRT	–	first row of the object's spectrum (world coordinates)
OBJ_END	–	last row of the object's spectrum (world coordinates)
NET_INTENS	–	net peak intensity the object's spectrum
SKY_STRT	–	first row of sky regions (world coordinates)
SKY_END	–	last row of sky regions (world coordinates)
SKY_SLIT	–	number of slitlet in which the sky was found

Table L.6: Windows table

Table STANDARD		
Label	**Unit**	**Description**
MAGNITUDE	–	magnitude
WAVE	ANGSTROEM	wavelength
BIN	ANGSTROEM	bin width
FLUX	ERG/S/CM/CM/ANG	flux

Table L.7: Standard star table

- correct the object frames for bias, overscan, and dark
- make sure that the dispersion direction in all your frames is along the x-axis

For the averaging of the frames you may want to use the command `COMBINE/LONG` from the context LONG. Now you can start the session with

Midas ...> `SET/CONTEXT mos`

The initialization of the keywords to their default values is done by

Midas ...> `INIT/MOS`

If you saved the results of an earlier reduction session with

Midas ...> `SAVE/MOS session`

you can now initialize these parameters again (and restore all auxiliary tables) with

Midas ...> `INIT/MOS session`

A demonstration of the package can now be obtained with

Midas ...> `TUTORIAL/MOS`

which has an automatic (default) and an interactive mode (first parameter) and also allows

Name	Default value	Contents
Keywords for flat-field related commands		
FLATLIM/R/1/3	0,0,0	detection parameters for slitlet location (detection threshold, median width, lower limit)
FFORD/I/1/1	3	order of fit for the FF normalization
NORMMET/C/1/20	POLY	method for FF normalization
ISLIT/I/1/1	0	counter for slitlets
NSLIT/I/1/1	0	total number of slitlets
SCAN_POS/D/1/1	0.	scan position and width for slitlet location
Keywords for wavelength calibration related commands		
SEAPAR/C/1/20	200,5	detection threshold and width
YBIN/C/1/20	3,3	Step and Binning in Y
CENTMET/C/1/16	GRAVITY	centering method for calibration lines
XPOS/D/1/50	0.	Positions of identified lines (pixels)
LID/D/1/50	0.	Positions of identified lines (Angstroems)
WLCMET/C/1/2	FC	Wavelength calibration method (Ident/Linear/ Fors) and (Constant/Variable/Fit)
POLTYP/C/1/10	CHEBYSHEV	type of polynomials used for fitting (Polynom, Legendre or Chebyshev)
TOL/R/1/1	2.0	tolerance for autom. wavel. ident. (< 0 Å, > pix)
TOLWIND/I/1/1	4	tolerance window for interact. wavel. ident.
WLCORD/I/1/2	2,1	order of fit used to compute the dispersion
ALPHA/R/1/1	0.2	Rejection parameter for lines matching [0,.5]
SHIFTTOL/I/1/1	10	tolerance for line identification from one slitlet to next
YSTART/I/1/1	0	Starting row for calibration (pixel value)
WLCNITER/I/1/2	3,20	Minimum, Maximum number of iterations
MAXDEV/R/1/1	10.	Maximum deviation (pixels)
WCENTER/D/1/1	0.	Central wavelength
AVDISP/R/1/1	0.	Average dispersion per pixel
PLOTC/C/1/1	N	Plot residuals of wavelength calibration
CAL/I/1/100	0	Results of moscalib (+1 dispersion relation fitted, -1 no dispersion relation fitted)
DISP/I/1/1	0	amount of intermediate display for `CALIB/MOS`
GRISM/I/1/1	1	No. of grism used
REJTHRES/R/1/1	3.	rejection threshold

Table L.8: Keywords used in context MOS

Name	Default value	Contents
Keywords for rebinning commands		
REBMET/C/1/12	LINEAR	Rebinning method (LINEAR, QUADRATIC, SPLINE)
REBSTRT/D/1/1	0.	Starting wavelength for rebinning
REBEND/D/1/1	0.	Final wavelength for rebinning
REBSTP/D/1/1	0.	Wavelength step for rebinning
Keywords for object extraction and sky subtraction related commands		
THRESH/R/1/1	-0.04	detection threshold for object search
WIND/I/1/1	5	detection window for object search
XBIN/I/1/1	20	Binning in X for object search
SCAN_POS/D/1/1	0.	center for scan (define/mos) in world coordinates
INT_LIM/D/1/1	0.001	fraction of central intensity where object limits shall be defined
NOBJ/I/1/1	0	number of objects found by define/mos
NSKY/I/1/1	0	number of sky-regions found by define/mos
SKYMET/C/1/16		method used to fit sky
SKYPAR/I/1/6	0	order of fit for polynomial or width of window for median
EXTPAR/C/1/60		extraction parameters (order, iter)
Keywords for frame names		
OBJ/C/1/60		object frame
EXTOBJEC/C/1/60		extracted object frame
CALOBJEC/C/1/40		extracted object frame
SKYFRAME/C/1/60	sky	sky frame
WLC/C/1/60	wlc	calibration frame
FLAT/C/1/60	flat	flat-field
NORMFLAT/C/1/60	normflat	normalized flat-field
Keywords for table names		
MOS/C/1/60	mos	table with slitlets' positions
LINFIT/C/1/60	coerbr	table with dispersion coefficients
LINPOS/C/1/60	linpos	table with wavelength positions
LINECAT/C/1/60	hear	table with wavelength positions
WINDOWS/C/1/60	windows	table with sky and objects positions

Table L.9: Keywords used in context MOS (cont'd)

Name	Default value	Contents
Keywords for CCD parameters		
STEP/D/1/2	0.,0.	step-size of raw frame
START/D/1/2	0.,0.	start values of raw frame
NPIX/I/1/2	0,0	size of raw frame
CCDPAR/C/1/60		read-out-noise and conversion factor of CCD

Table L.10: Keywords used in context MOS (cont'd)

to look at selected parts of the package (second parameter). In order to create all necessary test data you should run it once automatically.

Most of the numerical parameters are normally set to "sensible" values (which only means that they looked reasonable to the people who wrote this context), but the keyword for object ($\boxed{\texttt{OBJ}}$) is empty since there are no sensible defaults for these data. This holds also true for other keywords, which are normally filled either from the headers of your object (etc.) frames or with the results of MOS commands. Thus for the very first try you should just fill in the keyword mentioned above and continue (example see below).

Midas ...> `SET/MOS obj=fors0001`

If you want to make sure that the setups of your files are all the same use the command `CHECK/MOS`.

!!! NOT YET IMPLEMENTED !!!

By default it will check the FORS keywords for

- slitlets' positions
- grism
- CCD parameters (e.g. binning)
- NAXIS, START, STEP

If your data were not produced by FORS you may give the FITS keywords that should be compared in an ASCII file. The use is then

Midas ...> `CHECK/MOS file`

Now you may start the real business!

L.7.2 Locating slitlets and flat-field correction

The very first thing you should do now is to locate the limits of your slitlets because this information is needed for all further commands. Therefore you type

Midas ...> `LOCATE/MOS`

This should produce the table $\boxed{\texttt{MOS}}$.tbl (**mos**) with the columns

`:SLIT`	sequential number of slitlet
`:YSTART`	first row of slitlet (world coordinates)
`:YEND`	last row of slitlet (world coordinates)
`:XOFFSET`	offset of slitlet from center of CCD

and write the total number of slitlets to `NSLIT` (**0**). It may be that the threshold defined by `FLATLIM)(1` (**0**) is either too low (e.g. below bias value) or too high for your data. Also, the width (`FLATLIM(2)`) (**0**) may be chosen to high or too small. If you detect too many slitlets, where only noise is visible, you should increase (`FLATLIM(3)` (**0**)). You also can change the scan position and width (`SCAN_POS`).

Then you should have a look at your flat-field and try again with

Midas ...> `SET/MOS flatlim=threshold,width,limit`

Midas ...> `SET/MOS scan_pos=xpos xbin=width`

Midas ...> `LOCATE/MOS`

You may also try to identify the slitlets interactively with

Midas ...> `LOAD {flat}`

Midas ...> `DEFINE/SLIT init`

Midas ...> `DEFINE/SLIT add ##`

where ## stands for the number of slitlets you want to identify. You will have to determine the offsets between the slitlets with `OFFSET/MOS` (see below).

If you do not have FORS data the column `:xoffset` will be set to zero. This is due to the fact that for FORS data the slitlet positions given in the header of the frame are transformed to offsets from the center of the CCD. This transformations is obviously not valid for other instruments. As you will need the offsets for the wavelength calibration frame you can determine the offsets **relative to the first one (which is not necessarily identical with the center of the CCD)** with

Midas ...> `OFFSET/MOS`

As this command does a line search in the wavelength calibration frame `WLC`.bdf (**wlc**) and correlates only the detected arc lines the resulting offsets may be wrong if there are not enough lines to get unambiguous correlation results. Setting the parameter `SEAPAR` (**200,5**) to the values successfully used for SEARCH/MOS will help to yield reasonable results.

Normally spectroscopic flat fields show the spectral signature of the lamp with which they were taken. You can take out this spectral intensity distribution with

Midas ...> `NORMALIZE/MOS`

By default this command will fit a polynomial of `FFORD`th (**3**rd) order to the averaged (along the slitlet) spectral intensity of the flat field (`FLAT` (**flat**)) (separately for each slitlet) and divide it by these fits. The results are stored in the frame `NORMFLAT` (**normflat**).

Alternatively, you can normalize the flat field by dividing through an average smoothed with a median filter. To perform the actual flatfield correction together with the normalization type

Midas ...> `FLAT/MOS`

This command will do the normalization and divide the frame `OBJ`.bdf by `NORMFLAT`.bdf. If you have not given any name for the result frame it will derive the name of the flat field corrected object frame by adding an 'F' to the name of the input frame (e.g. `Ffors0001`).

L.7.3 Wavelength calibration

Line search and interactive identification

The detection of arc lines in the wavelength calibration frame `WLC`.bdf (**wlc**) for all slitlets selected in `MOS`.tbl (**mos**) is done by

Midas ...> `SEARCH/MOS`

This command will take the slitlets' limits from `MOS`.tbl and the eventual stepping and binning factor from `YBIN` (**3,3**). In all CCD rows of `WLC`.bdf that are selected by these parameters it will look for intensities above the chosen threshold when comparing to the median intensity over a chosen window (threshold and window are defined by `SEAPAR` (**200,5**)). Any lines detected this way will be centered by `CENTMET` (**GRAVITY**) and stored in `LINPOS`.tbl (**linpos**).

:X	x-position of line (world coordinates)
:Y	y-position of line (world coordinates)
:PEAK	maximum intensity of line
:SLIT	number of slitlet

With

Midas ...> `LINPLOT/MOS`

you will get a plot of the x-positions of all found arc lines versus the CCD rows. With

Midas ...> `LINPLOT/MOS 1 3`

you will get a plot of the first CCD row of slitlet 3 for which line positions are stored in `LINPOS`.tbl. The detected lines are marked in the plot.

If you do not know the central wavelength and mean dispersion of the grism you used you have to identify at least two arc lines in any row of any slitlet.

Midas ...> `IDENTIFY/MOS`

The row, in which the lines shall be identified, is read from `YSTART` (**0**). If you do not provide any number either there or on the command line, the program will take the first CCD row that has been scanned. As frame it will use `WLC`.bdf.

Offsets between slitlets

To determine the offsets between the slitlets for Non-FORS data use

Midas ...> `OFFSET/MOS`

This command will take the calibration spectra of the different slitlets from `WLC`.bdf and correlate them with the arc spectrum of the first slitlet. Thereby it will derive their offsets relative to the first slitlet (**which will normally not be in the center of the CCD**).

Wavelength calibration

After identifying the arc lines you may now start the actual wavelength calibration with e.g.

Midas ...> `SET/MOS wlcmet=IC`

Midas ...> `CALIBRATE/MOS`

The first character of `WLCMET` gives the way the first lines are identified (**I**dentify, **L**inear, **F**ors), the second character gives the mode of fitting (**C**onstant or **V**ariable linelist, **Fi**t all). For a detailed description see subsection L.3.3. This program fits a dispersion relation at first to the row with the identified lines, then to the slitlet, and afterwards to all slitlets (row by row, separately for each slitlet). The fitted line positions are compared to those listed in `LINECAT`.tbl (**hear**). The type of polynomials (Legendre or Chebyshev) is read from `POLTYP` (**CHEBYSHEV**), the fit order from `WLCORD(1)` (**1**), and the tolerance for automatic line identification from `TOL` ((**2**)). `TOL` can be given in pixel (> 0) or wavelength units (< 0). In case you know the central wavelength and the mean dispersion of the grism you used and you determined the offsets of the slitlets with `OFFSET/MOS` you have to correct the central wavelength for the offset between the center of the CCD and the reference slitlet used by `OFFSET/MOS`.

Example: Your reference slitlet has an offset of -100 relative to the center of the CCD in x-direction and you have a mean dispersion of 2 Å/pixel and a central wavelength of 5500 Å. This central wavelength will always lie at the x-position of the respective slitlet, which is in this case -100 pixels (i.e. -200 Å) from the center of the CCD. This means that you have a wavelength of 5700 Å at the center of the CCD within the reference slitlet. This wavelength should be used as `wcenter`, since the program assumes that `:xoffset = 0` means that the slitlet is at the center of the CCD in x-direction.

Then you can use the following command

Midas ...> `SET/MOS wcenter=central_wavelength`

Midas ...> `SET/MOS avdisp=mean_linear_dispersion`

Midas ...> `SET/MOS wlcmet=LC`

Midas ...> `CALIBRATE/MOS`

which will read the two parameters from `WCENTER` and `AVDISP`. If you have FORS data use

Midas ...> `SET/MOS wlcmet=RT`

Midas ...> `CALIBRATE/MOS`

This will read the grism number from the header of `WLC` and then take the parameters stored in the respective keyword. The plot option `PLOTC` (**N**) decides whether or not you get a plot of the residuals and the display option `DISP` (**0**) influences the amount of intermediate results displayed on screen. `CALIBRATE/MOS` stores the results in `LINFIT`.tbl (**coerbr**) in the columns

:SLIT	number of slitlet
:ROW	row for which dispersion relation was fitted (pixel coordinates)
:Y	row for which dispersion relation was fitted (world coordinates)
:RMS	r.m.s. error of fit
:COEF_i	fit coefficients

and adds to `LINPOS`.tbl (**linpos**) the columns

:WAVE	wavelength identification for :X
:WAVEC	fitted wavelength for :X
:RESIDUAL	:WAVE-:WAVEC
:REJECT	mark for rejected lines

Slitlets where no dispersion relation could be fitted are marked with '-1' in column `:SLIT`. With

Midas ...> `RESPLOT/MOS ? slit`

you can also get a plot of the residuals for a certain slit (`slit` = number) that are stored in `LINPOS`.tbl (**linpos**) (or in the first parameter given on the command line).

Rebinning

You may now want to rebin your object frame to constant wavelength steps 2-dimensionally with `REBIN/MOS`. For this you have to provide start, end, and step of the wavelength range you want to rebin with

Midas ...> `SET/MOS rebstrt=start rebend=end rebstp=step`

and then use

Midas ...> `REBIN/MOS`

The interpolation between pixels along the x-axis is taken from `REBMET` (**LINEAR**) and the order of interpolation along the y-axis between scanned rows is read from `WLCORD(2)` (**1**).

L.7.4 Object definition and sky subtraction

To define your object's and sky regions use

Midas ...> `DEFINE/MOS`

By default it automatically averages `XBIN` (**20**) columns around the position `SCAN_POS` (**0=center of frame**) of the frame `OBJ` and searches for objects in this averaged frame that have an intensity above `THRESH` (**-0.04**) when compared to a median over `WIND` (**5**) pixels. `THRESH` can be absolute (> 0) or relative to the median intensity (< 0). It searches the slitlets that are defined by `MOS`.tbl (**mos**) and stores the results in `WINDOWS`.tbl (**window**). The plot option (**0**) defines whether you get a two-dimensional display of the result, a graphical plot, both, or nothing (default).

If an object is detected a gaussian is fitted to its spatial profile and the limits of the object's region are defined at those pixels where the gaussian fit has reached `INT_LIM` (**0.001**) of the central intensity. On both sides of each object a safety margin of **3** pixels is established (can be overridden manually later); the remaining slitlet is defined as sky region. It may be advisable to perform a very crude sky subtraction first to get rid of the sky continuum intensity. This can be done with `SKYFIT/MOS` with **`skymet=nowindows`**. This command determines the median value along the columns over the slitlets. If you choose this way you should use an absolute threshold for object detection. If you want to search for all objects at the same wavelength region you have to rebin your frame first to constant wavelength steps with `REBIN/MOS`.

After the automatic definition of objects' and sky regions you are asked if you are satisfied with the results. If you are not (or are not yet sure), you answer 'no' and can now inspect the results more closely in the graphical plot and also change the results manually. You may also start with the interactive definition using `DEFWIN/MOS`.

The results are stored in table `WINDOWS` (**windows**):

`:Obj_Slit`	number of slitlet for object
`:Obj_Strt`	first row of object
`:Obj_End`	last row of object
`:Sky_Slit`	number of slitlet for sky
`:Sky_Strt`	first row of sky
`:Sky_End`	last row of sky

The command `SKYFIT/MOS` is used to fit the sky. Normally, the sky regions are taken from the table `WINDOWS`.tbl. Several sky windows may be defined in each slitlets. To fit the sky background you may use the median along the sky regions (**`skymet=median`**) or a polynomial fit (**`skymet=polynomial`**). Method polynomial requires rejection of cosmic rays and bad pixels. These pixels are rejected by `SKYFIT/MOS` before fitting the data. Read out noise (in electrons), gain (electrons/adu) and the rejection criterion in units of σ must be specified in keywords `CCDPAR` and `REJTHRES` (**3**).

Midas ...> `SKYFIT/MOS ? ?  ?  ?  3 poly 8,2.3,3` does a polynomial fit of 3^{rd} order in the frame `OBJ`.bdf, in the regions defined by `WINDOWS`.tbl (within the slitlet limits listed in `MOS.tbl`) and stores the result in `SKYFRAME` (**sky**).bdf.

L.7.5 Object extraction

Use `EXTRACT/MOS` to sky-subtract and extract the detected objects in an output frame with an optimum extraction scheme using different weights for the individual rows (following Horne, 1986, PASP 98, 609). The procedure is iterative. If the number of iterations `EXTPAR(2)` is set to a negative number, a spectrum computed with equal weights is returned. The sky frame (`SKYFRAME` (**sky**)) and the CCD read-out-noise and gain (`CCDPAR`) are used to compute the errors of the resulting spectra. The extracted spectra are stored in a 2-D frame, each line corresponding to one extracted object. In addition, the errors for the extracted spectra are stored in this frame, one line per spectrum.

To rebin the resulting frames to constant wavelength steps use:

Midas ...> `SET/MOS extobjec=in`

Midas ...> `SET/MOS rebstrt=start rebend=end rebstp=step`

Midas ...> `REB1D/MOS ? reb`

This will split the extracted frame in.bdf into individual frames with constant wavelength steps, named reb0001, reb0002, etc. using the dispersion coefficients of `LINFIT`.tbl.

If you want to avoid rebinning noise you can use `APPLY/MOS` to produce a table with the columns `:WAVELENGTH, :FLUX, :FLUX_ERROR,` and `:BIN`.

Midas ...> `SET/MOS extobjec=in calobjec=ap`

Midas ...> `APPLY/MOS`

This will take the frame in.bdf and convert it to tables ap0001.tbl, ap0002.tbl, etc. using the dispersion coefficients of `LINFIT`.tbl.

Appendix M

FEROS

M.1 Introduction

This appendix describes the use of commands which have been written to reduce spectra taken with the fiber-linked echelle spectrograph FEROS. Most of the commands, however, are not specific to FEROS so that they can be used to reduce data from other fiber-linked echelle spectrographs as well.

This chapter covers the following items:

- Brief description of FEROS
- General description of the FEROS data reduction software, i.e. command parameters and execution
- Description of the use of the batches for on-line reduction of FEROS data

M.2 Brief description of FEROS

FEROS is a bench-mounted fiber-linked echelle spectrograph built for the ESO 1.52m spectrographic telescope. With one exposure it covers the spectral range from 3600-9200 Å on a 2×4k EEV CCD chip. The main dispersion axis runs along the longer side of the CCD. Read-out direction is such that the main-dispersion is along the y-axis. Due to the use of a prism cross-disperser the spectral orders are strongly curved on the CCD. The images of the fiber are sliced in order to increase the spectral resolution to about 48,000. The price to pay for the enhanced resolution is a complicated and broad cross-order profile.

FEROS uses two fibers one of which is used as object fiber. The second fiber can be used in two modes.

- In the first mode the second fiber is used for recording a sky spectrum simultaneously with the object exposure.
- In the second mode, the spectrum of a ThAr comparison lamp is recorded simultaneously with the object spectrum. This mode is used for increased accuracy in

wavelength calibration and is typically used for planet search programs. The intensity of the ThAr lamp is attenuated (via a filter with variable attenuation) in order to allow long ThAr exposures. The sky spectrum cannot recorded simultaneously in this mode.

For the object-calibration mode it is especially important that there are no very strong lines in the calibration lamps, which could lead to blooming of the CCD and thus distort the stellar spectrum. Therefore, FEROS uses a filter to suppress the red part of the ThAr spectrum, where many strong lines of Ar are present. In order to have any useful calibration lines in the red part of the spectrum, a Ne lamp is used in addition to the ThAr lamp.

The spectrograph itself has no moveable parts, i.e. the spectral format is fixed. This allows relatively easy on-line reduction.

M.3 Requirement for the FEROS DRS

The FEROS data reduction software has been developed to take care of the peculiarities of the FEROS data format. However, it should be possible to use the software for similar instrument with no or only minor modifications. Also, the software was written to easily allow batch processing and is used at the telescope to provide a full on-line data reduction.

The basic steps in the reduction are as follows:

M.4 Order definition

Due to the use of an image slicer for the two fibers, the FEROS order definition uses a special technique for finding and centering orders. The definition of the orders is done in several steps.

- First, the position of the orders is defined near the middle of the frame. A central cut through the orders is cross-correlated with a template which should match as good as possible the cross-order profile. Centering the peaks of the correlation function gives the centers of the orders near the center of the CCD. If the template consists of the profile of a double fiber, both fibers are centered simultaneously. This is the normal way of operation.

- Second, the orders are followed along the main dispersion direction by cross-correlating cuts taken at varying y-positions with a secondary template extracted from the central cut.

- Finally, the detected x-y positions of all orders are fitted with a polynomial. This is done for each order individually. The polynomial coefficients are kept in a table for the further steps.

Order definition is done with the command DEFINE/FEROS.

M.5 Background subtraction

The background of the FEROS spectra consists of several components, mainly

- an electronic bias level, which can be determined from the overscan region of the CCD or from bias exposures,
- the CCD dark current, which can be determined from a series of long darks, and
- scattered light, which is smoothly varying over the CCD.

The latter contribution is determined by measuring its level outside the spectra. In the case of FEROS, this is the region between the orders and between the fibers. The region between the fibers is small, but is independent of order number. The distance between the orders is strongly depending on order number and is largest in the blue region. The background level is measured by taking the median in small regions at regular intervals in the y-direction. The measured values are approximated by fitting a 2-D smoothing spline function. Since the 2-D spline requires a rectangular grid, the median values are interpolated in x-direction before the spline fit.

The dark current should be subtracted before scattered light is subtracted, since it may have strong spatial variations which can not be taken into account by the spline fit. Background subtraction is done with the command BACKGR/FEROS.

M.6 Order extraction

After background subtraction, the spectral orders can be extracted. This is done by defining offsets in x-direction (with respect to the center of both fibers) and summing up the flux within a defined slit-width around these offsets. Both fibers are treated completely separately.

For reason of efficiency and simplicity, this is done in two steps.

- First, the pixels are re-ordered (rectified) in a new 2-D frame. It is important to note that this step involves *no* resampling. The fractional pixel offsets are kept in a separate file. The corresponding command is RECTIFY/FEROS.
- Second, the fluxes are summed up, producing a new 2-D frame (pixel-order space) for each fiber. The corresponding command is EXTRACT/FEROS.

The extraction can be done by straight summation or by using an optimum extraction algorithm which also detects and removes cosmic ray events. Optimum extraction requires a good knowledge of the cross-order profile (COP). For FEROS, this profile is – due to the image slicer – very complicated. It is approximated by fitting the fractional flux per pixel at several distances from the center of the COP as a function of position along the order with a polynomial. Since rebinning has to be avoided, this step is complex and CPU intensive. Thus, the COP can be saved for further use. For FEROS, the COP is typically determined from a flat-field exposure and later used for extraction of object spectra. This assumes that the COP does not change its profile, which can reasonably be assumed only for fiber-linked spectrographs.

M.7 Flat-fielding

Since the flat-fields are also taken through the fiber, the FEROS spectra cannot reasonably be flat-fielded in a 2-D way. Instead the extracted spectra of the object and the flat-field lamp are divided to remove the pixel-to-pixel variations. This division also removes the blaze function of the echelle grating with good precision. However, this requires that the background subtraction is sufficiently accurate. The command corresponding is FLAT/FEROS.

M.8 Wavelength calibration

The wavelength calibration uses the ThArNe calibration frames. The orders are first extracted and then searched for emission lines. This is done with the command SEARCH/FEROS. Wavelengths are assigned to these lines iteratively using a catalog of wavelengths and starting with a preliminary dispersion relation. A global dispersion relation for the whole wavelength rane is used for fitting the line positions as a function of y-position and order number. The global formula used is of the form

$$\lambda m = \sum \sum a_{ij} x^i m^j \qquad \text{(M.1)}$$

where x and m are the position along the dispersion axis and the order number, respectively, and a_{ij} are the coefficients to be fitted. For the order of the polynomial $i, j \leq 4$ is sufficient for FEROS. Most of the higher terms are set to zero, so that only 15 free parameters are used. With this formula we can reach residuals of the order of 3mÅ rms over the full wavelength range. The wavelength calibration is done with the command CALIB/FEROS.

M.9 Rebinning

The derived dispersion relation can be used to rebin the spectra in constant steps in wavelength on a linear or logarithmic scale. The barycentric correction can (and should) be applied in the same step. The rebinned spectra are still kept in 2-D frames (wavelength-order space). Different options for the rebinning are provided. The corresponding command is REBIN/FEROS.

M.10 Order merging

¿From the rebinned quasi 2-D spectra, individual orders can be extracted in individual files or all orders can be merged in a single 1-D spectrum. In the latter case, the spectra are averaged in the overlapping regions. The edges of the orders containing too little signal have to be removed before the averaging. This is especially important for FEROS, since the blue orders are much shorter than the CCD. A table with the limits of the orders is supplied for FEROS spectra. The corresponding command is MERGE/FEROS.

M.11 Description of FEROS keywords

This is the full list of FEROS keywords, with a short description of their meaning.

RAW_IMG/C/1/80 – Name of the image used for order definition.

WLC_IMG/C/1/80 – Name of the raw wavelength calibration frame. This keyword is used only for the on-line version of the FEROS software.

GUESS_TBL/C/1/80 – Name of table with the approximate order position. Use in the order definition.

CENTER_TBL/C/1/80 – Name of the table where the detected order position are stored.

LOC_CUTSTEP/I/1/1 – Step size in pixel for the cuts used to follow the orders.

LOC_WINDOW/I/1/1 – Window size in pixel used for the cross-correlation for order definition.

FIT_DEG/I/1/1 – Order of the polynomial used for description of the order positions.

LOC_THRES/R/1/1 – Background level in the cross-correlation used to follow the orders.

LOC_MODE/C/1/1 – Mode used for order definition. Valid values are S for peak searching and G for using a guess table.

LOC_METHOD/C/1/1 – Centering method for order definition. Valid values are R for center of gravity and A for gaussian centering.

CUTS_IMG/C/1/80 – Name of the image where the cuts through the order definition frame are stored.

FOLD_IMG/C/1/80 – Name of the image where the cross-correlations are stored.

TEMPL_IMG/C/1/80 – Name of the template image for the first step in order definition.

TEMPLT_IMG/C/1/80 – Name of the frame where the secondary templates extracted from the order definition frame are stored

FIT_IMG/C/1/80 – Name of the frames where the fitted cross-order profiles are stored

MASK_IMG/C/1/80 – Name of the mask frame where the detected cosmics are stored

INIT_TBL/C/1/80 – Name of the table with saved FEROS keywords

FLAT_IMG/C/1/80 – Name of the background subtracted image

FLATEXT_IMG/C/1/80 – Name of the extracted flat-field image. This is a pixel-order frame.

UNBLAZED_IMG/C/1/80 – Name of the flat-fielded frame. This is a pixel-order frame.

BG_MODE/C/1/1 – Mode of background subtraction. Valid values are B for subtract background and N for storing fitted background.

BG_STEPX/I/1/1 – Step size in x for background determination. This is used for grid for the 2-D spline fit.

BG_STEPY/I/1/1 – Step size in y for background determination. This is used for grid for the 2-D spline fit.

BG_WIDTHX/I/1/1 – Width in x for background determination.

BG_WIDTHY/I/1/1 – Width in y for background determination.

BG_MEDIANX/I/1/1 - Size of median window in x for median filtering used in background subtraction.

BG_MEDIANY/I/1/1 - Size of median window in y for median filtering used in background subtraction.

BG_DIST/I/1/1 – Minimum distance between orders where background is determined between orders. In two-fiber mode, the background is determined in the middle between fibers and between orders.

STRAIGHT_IMG/C/1/80 – Root name of rectified image. A 1 or 2 is appended for the two fibers.

PROFILE_W/I/1/1 – Width of spatial profile for order extraction.

FIBER_OFF1/I/1/1 – Offset of fiber 1 with respect to the fitted center of the order.

FIBER_OFF2/I/1/1 – Offset of fiber 2 with respect to the fitted center of the order

FIBER_MODE/I/1/1 – Number of fibers. Valid values are 1 and 2. This parameter affects order extraction and background determination.

IMG_WRITE/C/1/1 – Parameter to indicate if additional images should be generated during order extraction. This can used for debugging purposes.

EXT_IMG/C/1/80 – Root name of extracted image. The character 1 or 2 is appended for fiber 1 and 2, respectively.

SPECTR_TYPE/C/1/ – Spectrograph type. Valid values are F for single fiber and G for two fibers. This parameter affects order extraction.

EXT_MODE/C/1/1 – Extraction mode. Valid values are S for standard extraction, O for optimum extraction and and M for standard extraction with masking of cosmics.

PROFILE_GET/C/1/1 – Get spatial profile from spectrum. Valid values are Y or N.

EXT_ITER/I/1/1 – Number of iterations in optimum extraction.

CCD_RON/R/1/1 – Read-out-noise of CCD in electrons. Used for optimum extraction.

CCD_GAIN/R/1/1 – Gain of CCD in electron/ADU Used for optimum extraction.

CCD_THRES/R/1/1 – Threshold for clipping of cosmics Used for optimum extraction.

CCD_ROT/R/1/1 – Rotation angle of CCD in degrees. This affects calibration only in mode G, which is normally not used.

CCD_SHIFT/R/1/1 – Shift of CCD in x direction with respect to blaze center. This affects calibration only in mode G, which is normally not used.

COEF_COP/C/1/80 – Root name of cross order profile frame.

COEF_WLC/C/1/80 – Root name of wavelength calibration coefficient table.

INIT_WLC/C/1/1 – Get start values for dispersion coefficients. Valid values are G for grating relation, C for coefficients table and W for wavelengths of identified lines.

ORDER_FIRST/I/1/1 – Order number of first order on CCD.

ORDER_LAST/I/1/1 – Order number of last order on CCD.

LINE_W/I/1/1 – Estimated width of spectral lines in pixel.

LINE_THRES/R/1/1 – Threshold value for line searching.

LINE_POS_TBL/C/1/80 – Root name of table for storing detected lines.

LINE_MTD/C/1/80 – Centering method for line search. Valid values are GRAVITY, MAXIMUM, MINIMUM and GAUSS. Normally, GAUSSIAN should be used.

LINE_TYPE/C/1/80 – Type of spectral lines. Valid values are EMISSION and ABSORPTION. Should be EMISSION.

LINE_REF_TBL/C/1/80 – Name of laboratory wavelength table. A table with the name ThAr50000 is delivered with FEROS. This is based on a table of wavelengths optimized for a resolution of 50,000 and was kindly supplied by Herman Hensberge. It was extended with lines of Ne, since FEROS also has a Ne lamp and Ne has important lines in the red part of the spectrum.

ID_ALPHA/R/1/1 – alpha value for line identification.

ID_TOL/R/1/1 – Tolerance value for line identification.

ID_THRES/R/1/1 – Intensity threshold for first step of line identification. Only stronger lines will be used for this first step.

SPECTR_G/R/1/1 – Grating constant in lines/mm. This parameter affects calibration only in mode G, which is normally not used.

SPECTR_F/R/1/1 – Focal length of camera (mm) This affects calibration only in mode G, which is normally not used.

SPECTR_THB/R/1/1 – Blaze angle of spectrograph (degrees) This affects calibration only in mode G, which is normally not used.

REBIN_IMG/C/1/80 – Root name of rebinned spectrum.

REBIN_SCL/C/1/1 – Scaling of rebinned spectrum valid values are I for lInear and O for lOgarithmic.

REBIN_STEP/R/1/1 – Wavelength step size for rebinned spectrum.

REBIN_MTD/C/1/1 – Rebinning method. Valid values are L, Q, S for linear, quadratic and spline, respectively.

MERGE_IMG/C/1/80 – Root name of merged spectrum.

MERGE_MTD/C/1/12 – Merging method. Valid values are NOAPPEND, AVERAGE, SINC. The method AVERAGE is not recommended. SINC takes the length of the orders to be averaged into account. It needs a table with the fixed name BLAZE, where the order limits are stored. Method NOAPPEND is used to extract orders into individual files.

MERGE_DELTA/R/1/1 – Wavelength interval at the edges of orders to be skipped for averaging. Used for MERGE_MTD=AVERAGE.

MERGE_ORD/I/1/2 – Order numbers Ord1, Ord2 to be extracted. Used for MERGE_MTD=NOAPP.

M.12 Using the FEROS software on-line at the telescope

This section briefly describes the on-line operation of the FEROS software at the ESO 1.52m telescope. For a full documentation of FEROS see:

`http://www.ls.eso.org/lasilla/Telescopes/2p2T/E1p5M/FEROS/docu/pages/frames.html`

The FEROS on-line DRS allows a complete reduction of the science spectra taken during the night from the CCD system. The on-line DRS is based on the MIDAS context feros.

To install the on-line software after the DRS is installed, the MIDAS programs of the directory

`$MIDASHOME/$MIDVERS/stdred/feros/locproc`

have to be copied to the local midwork directory.

After CCD readout, the BIAS program

- includes the status informations from the CCD, the Telescope Control System (TCS), and the Instrument Control System (ICS) in the FITS header
- transfers the 2-D spectra to the instrument workstation (IWS) if the remote autosave is turned on (BIAS command remsave+).
- starts on the IWS the MIDAS programm `@@ loadccd fero [filenum]` where `filenum` is the running 4-digit filenumber of the CCD frame.

The `loadccd` program itself

- loads the frame `fero[filenum].mt` into the display,
- adds the incoming file to the catalogue `Feros.cat`,
- writes the FITS file to the DAT drive `/dev/rmt/1mn (BS 2880)`
- starts the automatic reduction via
- `@@ autoreduce fero [filenum]`

According to the four possible exposure types (FLATFIELD, CALIBRATION, DARK, and SCIENCE) given in the descriptor EXPTYPE, the autoreduce program starts the following actions:

- FLATFIELD: adds the incoming file to the catalogue `FF.cat`
- CALIBRATION: adds the incoming file to the catalogue `ThAr.cat`
- DARK: adds the incoming file to the catalogue `Dark.cat`
- SCIENCE:
  - adds the incoming file to the catalogue `Objects.cat`
  - start the pre-reduction of the the file (`@@ prered [filenum] raw_image`) where raw_image is the name of the input file for the following on-line reduction.
  - computes the barycentric velocity according to the telescope position and writes the result to the descriptor BARY_CORR
  - computes and subtracts the interorder background of the echelle spectrum (BACKGR/FEROS)
  - extracts the echelle orders (RECTIFY/FEROS, EXTRACT/FEROS)
  - removes the blaze function and the pixel-to-pixel variations (FLAT/FEROS)

- rebins the echelle orders to wavelengths (REBIN/FEROS) according using pre-determined dispersion coefficients. In this step also the barycentric correction is applied.
- merges the echelle orders (MERGE/FEROS) into two 1-D spectra named `f[filenum]1` and `f[filenum]2` where the spectrum with the ending 1 refers to the spectrum recorded on the object fiber and the spectrum with the ending 2 to the spectrum recorded on the sky/calibration fiber.

This standard reduction is controlled by the FEROS context keywords which can be listed together with their current contents by the command SHOW/FEROS and are set with the command SET/FEROS key=[value]. See below for useful keywords to be used during the observing session.

M.12.1 Initialization of the DRS at the beginning of the night

To use the automatic data reduction as described above, the DRS has to be initialized at the beginning of the night.

For this purpose several flat-field and wavelength calibration exposures have to be taken in the Object-Sky mode of FEROS before the beginning of the night following the following sequence:

- Reset the image catalogues `FF.cat`, `ThAr.cat`, `Dark.cat`, `Object.cat` with the command `@@ init ?  reset`
- Insert a new write-enabled DAT in the drive `/dev/rmt/1m`. Note that this DAT will be overwritten from the beginning erasing any previous contents! One 60m DAT can carry about 70 files. If you are to take more than 70 full frames, it is advisable to change the DAT before it is full.
- Use the instrument control software XFCU running on the CCD control PC next to the ISW to turn on the wavelength calibration lamp and use the BIAS CCD control software to take several (typical 2) exposures of 15 sec each. For details on the XFCU software and the BIAS software see the FEROS documentation. The resulting frames are automatically transfered to the IWS and added to the catalogue `ThAr.cat`.
- Switch with the XFCU program to the flatfield lamp and take, depending on the S/N needed for a appropriate reduction of the planned science exposures, 3 to 10 exposures of 30 sec. The frames are automatically transfered to the IWS and added to the catalogue `FF.cat`.
- Initialize the DRS for the night with the command `@@ init [guess]` where guess is the name of a previously saved guess session. Typically this is the session saved in the night before. The session names are formed automatically from the file number of the first calibration exposure in the catalogue `ThAr.cat` and the prefix ThAr, e.g., `ThAr0741`.

Now the following initialization steps are performed

- Initialization of the session keywords and tables (INIT/FEROS)
- Averaging of the frames of the respective catalogues `FF.cat`, `ThAr.cat`.
- Setting of the CCD gain keyword according to descriptor CCD_GAIN and the values specified in `init.prg`
- Definition of the echelle orders in the averaged flatfield (DEFINE/FEROS); the fitted positions are shown in the display window.
- Standard reduction of the flatfield (BACKGR/FEROS, RECTIFY/FEROS, EXTRACT/FEROS). The extraction is done twice: the first time, the cross-order profiles are determined for an optimum extraction with cosmic removal for the science exposures; the second time the flatfield orders are extracted. The name of the reduced flatfield is found in the keyword FLAT_IMG.
- Standard reduction of the wavelength calibration (BACKGR/FEROS, RECTIFY/FEROS, EXTRACT/FEROS). The name of the reduced calibration is found in the keyword WLC_IMG
- Search for emission lines in the reduced calibration frame (SEARCH/FEROS).
- Wavelength calibration by iterative fitting of the dispersion coefficients (CALIBRATE/FEROS). The resdiuals of the individual lines are plotted over the order number. The spread should not exceed an rms of 0.05 Å.
- The session parameters are saved as session WLC_IMG.

With this step completed, the FEROS on-line DRS is initialized.
Every new incoming spectrum will be saved and reduced now as described above.

M.12.2 On-line reduction options during the night

The context keywords allow to control the parameters of the reduction process. The keywords can be listed together with their current contents by the command SHOW/FEROS and are set with the command SET/FEROS key=[value].

If the keywords are set to new values, they will only affect the automatic on-line DRS for next incoming files. If one of the files already transfered to the IWS (`fero[filenum].mt` should be reduced again according to the new settings of the keywords, this is easily achieved by re-starting the `@@ autoreduce` command manually as follows:

`@@ autoreduce fero [filenum]`

Useful keywords for the observing session might be:

- EXT_MODE controls the method used for the extraction of the spectra. The three options are:

  - SET/FEROS EXT_MODE=S the standard extraction is performed where the flux across the slit is just summed.
  - SET/FEROS EXT_MODE=M the standard extraction is performed as above but with clipping of cosmics
  - SET/FEROS EXT_MODE=O the optimum extraction is performed with clipping of cosmics

- MERGE_MTD controls the merging of the orders. The options are:
  - SET/FEROS MERGE_MTD=SINC the default merging into a 1-D spectrum with weighted adding of overlapping regions. The lengths of the orders are determined from table `BLAZE.tbl`
  - SET/FEROS MERGE_MTD=AVE should not be used
  - SET/FEROS MERGE_MTD=NOAPP the orders are not merged but written into individual 1D spectra; the order number is appended to the filename as 4-digit number.

- REBIN_MTD controls the rebinning of the spectra. The options are:
  - SET/FEROS REBIN_SCL=I the rebinning is done into a lInear wavelength scale. The stepsize has to be set in the keyword REBIN_STEP.
  - SET/FEROS REBIN_SCL=O the rebinning is done into a lOgartithnic wavelength scale. The stepsize has to be set in the keyword REBIN_STEP.

Table M.1: Overview of FEROS commands

Command	parameters
INITIALIZE/FEROS	name
SAVE/FEROS	name
SHOW/FEROS	param
SET/FEROS	param=value
SAVINIT/FEROS	filename accessmode
HELP/FEROS	param
KEYDEL/FEROS	
DEFINE/FEROS	inimage intable outtable locval thres locpar
BACKGR/FEROS	raw flat centab bgmode bgparam medparam fibmode
RECTIFY/FEROS	image straightimage centab profilepar
EXTRACT/FEROS	straightimage extimage extpar params ccdpar coptab
FLAT/FEROS	inspec flatspec outspec
SEARCH/FEROS	spec linepar linepos meth linetype
CALIBRATE/FEROS	linetab reftab instpar coefftab centertab getcoeffs
REBIN/FEROS	extimage linetab rebimage scale step meth
MERGE/FEROS	inimage outimage params method
TUTORIAL/FEROS	

Table M.2: List of FEROS keywords

Keyword	default value	description
RAW_IMG/C/1/80	flat2d	name of raw image
WLC_IMG/C/1/80	wlc2d	name of wlc image
GUESS_TBL/C/1/80	echpos	name of guess table
CENTER_TBL/C/1/80	centers	name of order position table
LOC_CUTSTEP/I/1/1	10	cut step for order definition
LOC_WINDOW/I/1/1	17	window size for order definition
FIT_DEG/I/1/1	5	order polynomial fit degree
LOC_THRES/R/1/1	0.9	background level in order definition
LOC_MODE/C/1/1	G	definition mode; valid values: S,G
LOC_METHOD/C/1/1	R	definition method; valid values: R,A
CUTS_IMG/C/1/80	cuts	cuts image
FOLD_IMG/C/1/80	fold	fold image
TEMPL_IMG/C/1/80	template	input template image
TEMPLT_IMG/C/1/80	templatet	output template image
FIT_IMG/C/1/80	fitted_img	root name if fitted profiles image
MASK_IMG/C/1/80	mask_img	root name of masked pixels image
INIT_TBL/C/1/80	init	init table
FLAT_IMG/C/1/80	nobg2d	name of flatfielded image
FLATEXT_IMG/C/1/80	flatxtrctd	name of extracted flatfield image
UNBLAZED_IMG/C/1/80	unblazed	name of unblazed image
BG_MODE/C/1/1	B	background determination mode; valid values: B,N,S
BG_STEPX/I/1/1	51	step in x for background determination
BG_STEPY/I/1/1	51	step in y for background determination
BG_WIDTHX/I/1/1	21	width in x for background determination
BG_WIDTHY/I/1/1	45	width in y for background determination
BG_MEDIANX/I/1/1	4	median filter size in x
BG_MEDIANY/I/1/1	11	median filter size in y
BG_DIST/I/1/1	50	minimum distance between orders
STRAIGHT_IMG/C/1/80	straightened	root name of straightened image
PROFILE_W/I/1/1	15	width of spatial profile
FIBER_OFF1/I/1/1	-9	shift of fiber 1
FIBER_OFF2/I/1/1	9	shift of fiber 2
FIBER_MODE/I/1/1	2	number of fibers
IMG_WRITE/C/1/1	N	write additional images
EXT_IMG/C/1/80	extracted	root name of extracted image

Table M.3: List of FEROS keywords (continued)

Keyword	default value	description
SPECTR_TYPE/C/1/1	G	spectrograph type; valid values: F,G
EXT_MODE/C/1/1	S	extraction mode; valid values: S,O,M
PROFILE_GET/C/1/1	N	get spatial profile from spectrum; valid Y,N
EXT_ITER/I/1/1	3	maximum iterations in optimum extraction
CCD_RON/R/1/1	3.5	readout noise of CCD (optimum extraction)
CCD_GAIN/R/1/1	0.66	gain of CCD (optimum extraction)
CCD_THRES/R/1/1	4.0	clipping threshold (optimum extraction)
CCD_ROT/R/1/1	2.4	rotation of CCD
CCD_SHIFT/R/1/1	0.0	shift of CCD in x direction
COEF_COP/C/1/80	cop_coeff	root name of cross order profile coefficient table
COEF_WLC/C/1/80	wlc_coeffs	root name of wavelength calibration coefficient table
INIT_WLC/C/1/1	C	get dispersion coefficients from spectrum; valid values: G,C,W
ORDER_FIRST/I/1/1	25	first order on CCD
ORDER_LAST/I/1/1	63	last order on CCD
LINE_W/I/1/1	5	width of spectral line
LINE_THRES/R/1/1	10000	background level
LINE_POS_TBL/C/1/80	found_lines	root name of found line table
LINE_MTD/C/1/80	gauss	line search method; valid values: GRA,MAX,MIN,GAUSS
LINE_TYPE/C/1/80	emission	line type; valid values: EMIS,ABSORP
LINE_REF_TBL/C/1/80	ThAr50000	name of laboratory wavelength table
ID_ALPHA/R/1/1	0.2	alpha value for line identification
ID_TOL/R/1/1	0.2	tolerance value for line identification
ID_THRES/R/1/1	0.0	intensity threshold for first step of line identification
SPECTR_G/R/1/1	79.0	grating constant (lines/mm)
SPECTR_F/R/1/1	410.0	focal length of camera (mm)
SPECTR_THB/R/1/1	63.4	blaze angle of spectrograph (degrees)
REBIN_IMG/C/1/80	rebinned	root name of rebinned spectrum
REBIN_SCL/C/1/1	I	scaling of rebinned spectrum (lInear or lOgarithmic) valid values: I,O
REBIN_STEP/R/1/1	0.03	wavelength step of rebinned spectrum
REBIN_MTD/C/1/1	S	rebinning method; valid values: L,Q,S
MERGE_IMG/C/1/80	merged	root name of merged spectrum
MERGE_MTD/C/1/12	SINC	Merging methods: NOAPPEND,AVERAGE,SINC (Associated command MERGE/FEROS)
MERGE_DELTA/R/1/1	1.0	Wav. interval to be skipped (MERGE_MTD=AVERAGE)
MERGE_ORD/I/1/2	0,0	Ord1,Ord2 (MERGE_MTD=NOAPP). 0,0:all orders