ISAAC Data Reduction Guide 1.5

P. Amico, JG. Cuby, N. Devillard, Y. Jung, C. Lidman

January 30, 2002

Change Record

Issue/Rev.	Date	Section affected	Reason and Remarks
1.0	Dec 1999	All	Initial release
1.1	07/03/2000	All	Updated recipes
1.2	06/04/2001	All	Updated recipes
1.3	10/08/2001	SW	Web Release
1.4	01/11/2001	SW	Updated
1.5	29/01/2001	LW sections	LW Added

Contents

Chapter 1

Introduction

1.1 Purpose

The document is intended for astronomers who want to reduce ISAAC data. It describes the various data formats delivered by ISAAC, observational scenarios and reduction procedures. This document concentrates on the methodology rather than individual routines that are available in either IRAF or MIDAS. However, from time to time, we will mention some routines that the authors have found useful.

The document describes the algorithms implemented in the **eclipse** data reduction package. The eclipse package is an integral part of the ISAAC pipeline, which produces calibration products, quality control information and reduced data.

The pipeline does produce reduced data; however, this is not meant to replace more general reduction packages such as IRAF, MIDAS or IDL. The pipeline does not replace interactive analysis and cannot make educated choices. Thus, the data that the ISAAC pipeline produces should be considered as a way of quickly assessing the quality of the data (a quick look if you like) or a first pass at the reduction of the data.

Throughout this document we will list the **eclipse** routines that are used to reduce ISAAC data and we will give a short description of how they can be used. We will also list the shortcomings these routines have, so that users can decide if they need to reduce their data more carefully. For completeness, we have also included a description of the eclipse routines whose primary aim is to provide quality control information. These routines are probably of little interest to astronomers.

This document does *not* describe the ISAAC instrument, its modes of operations, how to acquire data, the offered templates, or the various issues attached to Phase II Proposal Preparation. The reader is assumed to have read the

ISAAC User's Manual beforehand, and have a basic knowledge of infrared data reduction in imaging and spectroscopy.

This document is a living document, and as such follows the evolution of the recipes, the implementation of new algorithms, enhancements, new supported modes, etc. Anything related here has to be understood to be valid **at the date of writing**.

1.2 Reference Documents

1 ESO DICB - Data Interface Control Document - GEN-SPE-ESO-00000-0794
2 ISAAC Calibration Plan - VLT-PLA-ESO-14100-1384
3 ISAAC User Manual

1.3 Abbreviations and Acronyms

Dec	Declination
DICB	Data Interface Control Board
DIT	Detector Integration Time
DPR	Data Products (FITS identifier)
eclipse	ESO C Library Image Processing Software Environment
ESO–MIDAS	ESO's Munich Image Data Analysis System
FITS	Flexible Image Transport System
IRAF	Image Reduction and Analysis Facility
ISAAC	Infrared Spectrometer And Array Camera
LSS	Long Slit Spectroscopy
PAF	PArameter File
RA	Right Ascension
UT	Unit Telecope
VLT	Very Large Telescope

1.4 The ISAAC Instrument

A full description of the capabilities of this instrument can be found in the ISAAC user manual available on the ISAAC web site. Please refer to this document for the latest user information about ISAAC.

1.5 The ISAAC Pipeline

The ISAAC pipeline as understood in the DataFlow context is described in its own document [1] describing the design and installation procedures for the

[1] ISAAC pipeline and quality control User's Manual, not public

ISAAC pipeline. Bear in mind that this manual is intended for DataFlow operation people who want to get a better insight about how the recipes are being implemented in the control-room in automatic mode. The current document, on the other hand, is aimed at describing the algorithmic concepts present in the recipes, and provide help for users who want to analyze data from ISAAC.

Chapter 2

Data Reduction Software

2.1 The ISAAC Pipeline

2.1.1 Raison d'être

The ISAAC pipeline is an ESO development to provide reduced data to service mode users, visitors at the telescope and operators who execute service mode observations. The pipeline runs in two modes: on-line (for both service and visitor mode programs) and off-line (for service mode programs).

On-line data reduction happens in the control-room during the night without any human intervention. The goal here is to provide enough information to assess the quality of the acquired data in the shortest possible time. This quick feedback is often useful e.g. to check the image quality in imaging modes. On-line reduced data are not automatically distributed to observers.

Off-line data reduction happens before the data are packed and distributed. The main difference between the on-line and off-line versions is that calibration frames, such as flats and darks, are used.

In both cases, it should be understood that the pipeline does not replace careful data analysis and the trained eye of the astronomer. They pipeline implements recipes that are meant to work in most cases. Most astronomers choose to take the pipeline input as a first guess, then refine some points where the default method is not providing the best results for what they are doing.

2.1.2 Downloading and installing

The ISAAC pipeline is entirely based on the **eclipse** library. You need to download the library to compile the pipeline. Both packages can be found at:

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

Installation instructions are provided with the package and on the Web site. After installation is complete, you should end up with a number of executables in `eclipse/bin`. The main program you are looking for is called `isaacp`, it contains the recipes for most of the ISAAC data reduction modes, together with some basic documentation for the command usage.

2.1.3 Pipeline usage

In **eclipse** version 4.0 and later versions, most of the ISAAC pipeline is contained in a single Unix command called `isaacp`. Additional commands are `jitter` and `lwjitter` for imaging jitter, they are bound to be included into `isaacp` sooner or later for consistency.

Once you have compiled `isaacp`, you can start launching it with no argument to get a list of supported recipes. To get more information about the command itself, try `isaacp man`. To get more information about one recipe, try `isaacp man` *recipe*, where *recipe* is the name of the recipe you are interested in.

2.2 Data Analysis Tools

You will probably need access to one of the standard data analysis packages for astronomy to help reduce your ISAAC data: IRAF, MIDAS, IDL, etc. You may want to implement your own routines and scripts to repeat the same algorithm on all your frames, or automate some of the data reduction process. This guide assumes that you have some basic knowledge of such packages.

Chapter 3

ISAAC FITS Information

Ancillary data attached to ISAAC files are all written into FITS headers. The following chapter lists the most important keywords and describes commands on how to retrieve them.

For ease of reading, keywords are shortened from `HIERARCH ESO A B C` to `A.B.C` (shortFITS notation).

Notice that all of this information is present in the ESO dictionaries, available from the ESO archive Web site. The following chapter only tries to summarize the most important information.

3.1 Extracting FITS Information

There are many tools to extract and parse FITS headers. One convenient way of extracting FITS information and displaying it on a terminal or re-directing it to a text file, is to use two stand-alone programs called **dfits** and **fitsort**. Both are included into the **eclipse** distribution.

`dfits` dumps a FITS header on stdout. You can use it to dump the FITS headers of many files, to allow the parsing of the output. Example:

```
% dfits *.fits | grep "TPL ID"
```

Usually, you want to get the value of a list of given FITS keywords in a list of FITS files. `fitsort` reads the output from `dfits`, classifies the keywords into columns, and prints out in a readable format the keyword values and file names. Example:

```
% dfits *.fits | fitsort NAXIS1 NAXIS2 BITPIX
```

`fitsort` also understands the shortFITS notation, where e.g. `HIERARCH ESO TPL ID` is shortened to `TPL.ID`. A classification example could be (both commands are equivalent, since `fitsort` is case-insensitive):

```
% dfits *.fits | fitsort TPL.ID DPR.TYPE
% dfits *.fits | fitsort tpl.id dpr.type
```

The output from this combination is something like:

```
FILE                                   TPL.ID                  DPR.TYPE
ISAAC.1999-04-09T19:16:32.047.fits     ISAACSW_img_cal_Darks   DARK
ISAAC.1999-04-09T19:56:51.011.fits     ISAACSW_img_cal_Darks   DARK
ISAAC.1999-04-09T20:37:09.058.fits     ISAACSW_img_cal_Darks   DARK
ISAAC.1999-04-09T21:00:49.003.fits     ISAACSW_img_cal_Darks   DARK
ISAAC.1999-04-09T21:24:27.081.fits     ISAACSW_img_cal_Darks   DARK
ISAAC.1999-04-09T21:37:26.086.fits     ISAACSW_img_cal_Darks   DARK
```

This kind of table is useful in getting an idea of what is present in a directory or list of directories. Loading such a summary table into a spreadsheet program also makes it conveniently readable.

3.2 Visualizing FITS Tables

Similarly, a FITS table can be visualized on the command-line without the need to call a full-fledged FITS-capable package. The `dtfits` command has been written for precisely that purpose. You will find it useful for spectroscopic data reduction, if you need to check out the results of the pipeline recipes that produce FITS tables.

`dfits` and `fitsort` will help you classify tables and see ancillary data attached to them, but `dtfits` will display all information contained in the table itself, in ASCII format on the command-line. There are various options to help make the output readable on a terminal, or by a spreadsheet program.

See the `dtfits` manual page to get more information.

3.3 File Names

- `PIPEFILE` (if set) contains the name of the product using the official naming scheme for ISAAC products. This name can be set using the renaming recipe (`is_rename`).
- `ORIGFILE` contains the name of the file on the instrument workstation.
- `ARCFILE` is the archive file name.

3.4 Telescope Keywords

Here is a non-exhaustive list of telescope keywords.

- `RA` Right ascension (J2000) in degrees. Notice that the comment field indicates the value in hh:mm:ss.s format.

- `DEC` Declination (J2000) in degrees. Notice that the comment field indicates the value in hh:mm:ss.s format.

- `ADA.POSANG` Position angle on sky as measured from North to East.(degrees).

- `TEL.AIRM.START` Airmass at start.

- `TEL.AIRM.END` Airmass at end.

- `TEL.AMBI.FWHM.START` Astronomical Site Monitor seeing at start. Note that this value might differ significantly from the ISAAC image quality, which is usually better.

- `TEL.AMBI.FWHM.END` Astronomical Site Monitor seeing at start. Note that this value might differ significantly from the ISAAC image quality, which is usually better.

3.5 Instrument Keywords

Here is a non-exhaustive slit of instrument keywords.

- `INS.MODE` The instrument mode. It is a label codifying the arm (short or long) and the mode (imaging, spectroscopy, etc.) used for the current frame. See the ISAAC User's Manual for more information about possible values of this keyword.

- `INS.FILT1.ID` SW filter on filter wheel 1.

- `INS.FILT2.ID` SW filter on filter wheel 2.

- `INS.FILT3.ID` LW filter on filter wheel 1.

- `INS.FILT4.ID` LW filter on filter wheel 2.

- `INS.OPTI1.ID` Mask or slit in position (e.g. slit_1).

- `INS.GRAT.NAME` Grating in position (LR or MR).

- `INS.GRAT.WLEN` Central wavelength (microns). Note however that the value recorded here – in spite of the number of digits – is a rough estimate of the central wavelength.

- `INS.GRAT.ORDER` Grating order.

- `INS.OPTI2.NAME` Name of the objective.

- `INS.LAMP1.ST` Argon lamp status.
- `INS.LAMP2.ST` Xenon lamp status.
- `DET.DIT` Detector Integration Time (seconds)
- `DET.NDIT` Number of averaged DITs.
- `DET.MODE.NAME` Detector readout mode.
- `INS.PIXSCALE` Pixel scale in arcseconds per pixel.

3.6 Observation Block Keywords

- `OBS.PROG.ID` Program ID.
- `OBS.NAME` Name of the OB (as prepared with P2PP).
- `OBS.TARG.NAME` Target package name (as prepared with P2PP).

3.7 Template Keywords

3.7.1 Keywords common to all templates

- `TPL.ID` contains an unique identifier describing the template which was used to produce the data. Frame selection in the pipeline is mostly based on this keyword value.
- `DPR.CATG` Data Product category (SCIENCE, CALIB, ...).
- `DPR.TYPE` Data Product type (OBJECT, SKY, ...).
- `DPR.TECH` Data Product acquisition technique (e.g. IMAGE, SPECTRUM).
- `TPL.NEXP` Number of scheduled exposures within the template. [1]
- `TPL.EXPNO` Exposure number within template.

A template may produce several different frame types. Frames are discriminated by the value of the `DPR` keywords: `DPR.CATG`, `DPR.TYPE`, and `DPR.TECH` take different values depending on the observed frame type.

[1] Templates might have been aborted, so it is possible to have fewer frames than `TPL.NEXP` in a batch.

3.7.2 Keywords common to the jitter templates

The offsets sent to the telescope for jitter observations, both in imaging and spectroscopy, are stored into 8 keywords. This applies to AutoJitter, AutoJitterOffset, and GenericOffset templates.

- `SEQ.CUMOFFSETX` and `SEQ.CUMOFFSETY` for cumulative offsets in pixels.
- `SEQ.CUMOFFSETA` and `SEQ.CUMOFFSETD` for cumulative offsets in arcseconds (alpha, delta).
- `SEQ.RELOFFSETX` and `SEQ.RELOFFSETY` for relative offsets in pixels.
- `SEQ.RELOFFSETA` and `SEQ.RELOFFSETD` for relative offsets in arcseconds (alpha, delta).

Cumulative offsets are always relative to the **first** frame in the batch (`TPL.EXPNO`=1). Relative offsets are always relative to the **previous** frame (`TPL.EXPNO`-1) in the batch.

If the same guide star is used before and after an offset, the offsetting accuracy is about 0.1 arc seconds. All recipes looking for offset information take this into account and will use the header offset information as a first guess and will refine the offset through cross-correlation techniques.

In AutoJitter mode, the jitter offsets are generated using a Poisson distribution. `SEQ.POISSON` is an integer describing the Poisson homogeneity factor used for this distribution. See the eclipse web page (http://www.eso.org/eclipse) for more information about this factor.

The **jitter** recipe from **eclipse** always expects offsets to be given in pixels, not in arcseconds. If your headers do not mention the offsets in pixels, you must translate arcseconds to pixels yourself and feed the information back into the **jitter** command. The input offsets are then given by an ASCII file instead of being read from the FITS headers.

3.8 Pipeline Products

To allow identification of pipeline products, some keywords are inserted in the output FITS headers.

- `PIPEFILE` is a standard 8-char FITS keyword which contains the name of the file as set by the pipeline when creating the product, it is useful as a label to identify the file. If this keyword is not set in your products, you can use the **`is_rename`** recipe to rename your files and set this keyword accordingly. Nothing requires this name to be set, it is only here for convenience.

- `PRO.DID` contains the version number of the dictionary used for all product keywords.
- `PRO.TYPE` contains the type of the data products as one of 'TEMPORARY', 'PREPROCESSED', 'REDUCED', or 'QCPARAM'.
- `PRO.REDLEVEL` contains the reduction status of the file, its possible values are not clearly defined yet for infrared instruments.
- `PRO.CATG` is probably the most important product keyword, since it labels each frame with a product ID unique to the recipe. It qualifies files with hopefully understandable product labels.
- `PRO.STATUS` specifies whether the reduction performed OK or whether some warnings were issued by the algorithm.
- `PRO.DATE` gives the date and time of the pipeline execution for the frame creation.
- `PRO.REC.ID` identifies the recipe that generated the file, with a unique name.
- `PRO.REC.DRS.ID` identifies the Data Reduction System that was used to produce the file (**eclipse** for all ISAAC recipes).
- `PRO.DATANCOM` specifies the number of raw frames that were combined to generate the product. Its exact meaning depends on the recipe, see each recipe documentation to learn what it refers to.

ISAAC files from which the electrical ghost has been removed, contain two additional keywords:

- `GHOSTREM` is a FITS boolean set to T(rue) if the electrical ghost removal procedure has been applied to the frame.
- `GHOSTVER` is an ID of the algorithm used for ghost correction.

3.9 Quality Control Products

Quality control outputs are produced into PAF files. A PAF file is a PArameter File containing formatted ASCII data looking like a FITS header. They are made so that human beings can read them easily, and so that they can also be parsed by software means.

Keywords belonging to the realm of quality control products are currently being defined in a dictionary for later acceptance by the DICB board at ESO. Expect changes in the months following the release of this document.

See the chapter dedicated to ISAAC pipeline products for a full description of all possible output keywords.

3.10 Summary Table

The following table summarizes all keywords introduced in this chapter, in alphabetical order.

FITS keyword	Meaning
ADA.POSANG	Position angle on sky (degrees)
ARCFILE	Archive file name
DEC	Declination (J2000) in degrees
DET.DIT	Detector Integration Time
DET.NDIT	Number of (averaged) DITs
DPR.CATG	Data product category
DPR.TYPE	Data product type
DPR.TECH	Data product acquisition technique
GHOSTREM	Flag for ghost removal
GHOSTVER	Ghost removal algorithm ID
INS.FILT1.ID	SW Filter ID in first wheel
INS.FILT2.ID	SW Filter ID in second wheel
INS.FILT3.ID	LW Filter ID in first wheel
INS.FILT4.ID	LW Filter ID in second wheel
INS.GRAT.NAME	Grating in position
INS.GRAT.ORDER	Grating order
INS.GRAT.WLEN	Grating central wavelength (microns)
INS.LAMP1.ST	Status of the Argon lamp
INS.LAMP2.ST	Status of the Xenon lamp
INS.MODE	Instrument acquisition mode
INS.OPTI1.ID	Mask or slit in position
INS.OPTI2.NAME	Optical objective name
INS.PIXSCALE	Pixel scale in arcseconds per pixel
OBS.PROG.ID	Program ID
OBS.NAME	Observation Block name
OBS.TARG.NAME	Target package name
ORIGFILE	File name given during acquisition
PIPEFILE	File name as attributed by the pipeline
PRO.CATG	Data product category
PRO.DATANCOM	Number of frames combined into this product
PRO.DATE	Pipeline reduction date
PRO.DID	Dictionary version for PRO keywords
PRO.RECi.ID	Applied recipe ID
PRO.RECi.DRS.ID	Data Reduction System ID
PRO.REDLEVEL	Data product reduction level
PRO.STATUS	Data product reduction status
PRO.TYPE	Data product type

`RA`	Right ascension (J2000) in degrees
`SEQ.CUMOFFSETX`	Cumulated offset in X, in pixels
`SEQ.CUMOFFSETY`	Cumulated offset in Y, in pixels
`SEQ.CUMOFFSETA`	Cumulated offset in alpha, in arcseconds
`SEQ.CUMOFFSETD`	Cumulated offset in delta, in arcseconds
`SEQ.POISSON`	Poisson homogeneity in jitter throw
`SEQ.RELOFFSETX`	Relative offset in X, in pixels
`SEQ.RELOFFSETY`	Relative offset in Y, in pixels
`SEQ.RELOFFSETA`	Relative offset in alpha, in arcseconds
`SEQ.RELOFFSETD`	Relative offset in delta, in arcseconds
`TEL.AIRM.START`	Airmass at start of exposure
`TEL.AIRM.END`	Airmass at end of exposure
`TEL.AMBI.FWHM.START`	ASM seeing at start
`TEL.AMBI.FWHM.END`	ASM seeing at end
`TPL.ID`	Template ID
`TPL.NEXP`	Number of exposures expected in the template
`TPL.EXPNO`	Exposure rank in the template

Chapter 4

An Overview of Short-wavelength Data

4.1 Introduction

The SW arm of ISAAC uses a Hawaii 1024x1024 array from Rockwell. When reducing data from the SW detector you should keep in mind the following points.

- The array is read out simultaneously in four quadrants.
- Within each quadrant, the read starts in the bottom left hand corner, reads along a single row before moving to the next row and then ends up in the top right hand corner.
- For data taken before 24/07/2001, the detector read out time is 1.7728 seconds. This also corresponds to the minimum integration time.
- For data taken after 24/07/2001, the detector read out time is 3.5451 seconds. This also corresponds to the minimum integration time.
- There is no shutter, so the array is continuously exposed to light.

The ISAAC IRACE controller and the associated DCS software control the readout of the array. They are extremely flexible and allow many different readout schemes to be used. Currently, only two readout schemes are supported: DCR (Double Correlated Read) and NDR (Non-Destructive Read). In the near future we may also offer other readout modes such as FDCR (Fast Double Correlated Read).

DCR first resets the array and then performs two reads, one at the beginning of the integration and one at the end. The difference between these two reads is the image.

NDR first resets the array and then reads the array N+1 times, where N depends on the DIT. Longer DITs will result in more reads. A line is fit to N reads (the first read is discarded) and the slope of the fit gives the image.

For a given DIT, DCR is faster, but NDR results in better noise performance. For these reasons, DCR is used when the background is high (imaging) and NDR is used when the background is low (spectroscopy).

It is important to appreciate that different parts of the detector are read out at different times. If we take DCR as an example, set the DIT to the minimum, i.e. 1.77 seconds[1], and set the clock at 0 seconds when the first row is read out, then row 1 is the difference of two reads that occur at 0 and approximately 1.77 seconds and row 512 is the difference between two reads that occur at approximately 1.77 and 3.54 seconds. During the first 1.77 seconds, row 512 is still collecting photons. In other words, different parts of the detector sample different parts of the integration ramp, and because the integration ramp is non-linear, a source at row 500 will always have fewer counts than the same source at row 100. If the peak flux of source and sky is less than 10,000 ADUs, then you do not need to worry about these effects, unless you wish to obtain photometry which is better than 1%. If it is greater then 10,000 ADUs and DITs are small, then these effects become important.

4.2 EXPTIME, NDIT and DIT

Up until March, 2000, the EXPTIME keyword was defined as NDIT $\times$ DIT. This caused some confusion as the frames are the average and not the sum of NDIT exposures. After March, 2000, the EXPTIME keyword is set to the DIT.

4.3 General Features of the ISAAC SW Infrared Detector

4.3.1 Bias variations

The ISAAC SW infrared detector bias is a function of the detector integration time (DIT) and the detector illumination. It also varies in time, most notably at the rows where the reading of the detector starts, that is rows 1, 2, 3, ... and rows 513, 514, 515, ...

It is therefore common to observe bias variations from one image to the next. This is particularly the case for the first image in a sequence (template) of images. These bias variations are non-uniform across the array, but are uniform along most rows. These variations are usually not a serious problem, and do not prevent one from using all the images, but they may require some

[1] For data taken after 24/07/2001, replace 1.77 with 3.55 seconds and 3.54 with 7.09 in the above text.

special treatment, e.g. fitting all lines with rejection of positive signal so as to derive the vertical pattern of the bias (or of its variation between images). The effect is strongest in DCR mode and is more easily seen in low background conditions.

4.3.2 Shift register glow

The detector shift registers generate light which is in turn detected by the detector. These glows are visible on 4 quadrants of the detector, at the bottom and top of the image. Although strongest in NDR, they can also be seen in DCR. The glow subtracts out perfectly when subtracting sky or dark frames that have been taken with the same DIT. Moreover, most of the glow is outside the useful part of the image in spectroscopy, and the noise induced by the glow is negligible in the useful part.

4.3.3 Electrical ghosts

The HAWAII detector generates 'electrical ghosts' which seem to have a reproducible behavior. The effect basically consists of an additional signal, which, on one row, is proportional to the sum of the intensity along this row *and* the row 512 rows away.

This effect is mostly an issue in spectroscopy of relatively bright sources (e.g. standard stars), as it might affect the level of the continuum, and e.g. the line equivalent widths.

The procedure to remove the electrical ghost is to use a dedicated **eclipse** recipe called `ghost`. The simplest way is probably to use it as:

```
% isaacp ghost *.fits
```

On some Unix systems, `*.fits` refers to too many file names to be handled correctly on the command-line. The solution is to use a loop, depending on the shell you use. In csh/tcsh this would be:

```
% foreach i (*.fits)
% foreach? isaac ghost $i
% foreach? end
```

The ghost recipe writes into any processed file header two keywords: `GHOSTREM` is a flag indicating that the process has been applied to the file, and `GHOSTVER` indicating what the algorithm version ID is. This way, the algorithm cannot be applied twice to the same file (unless you use the `--force` option).

The implemented algorithm is the following:

- Create a 1d signal in which each element contains the sum of all pixels along a line of the input image. In other words, collapse the 2d image along the x axis.

- The signal in [1..512] is added to the signal in [513..1024] and visa-versa.
- Multiply by 1.35e-5.
- Subtract element N of this 1d signal from line N in the input image.

4.3.4 Detector non-linearity

Firstly, you should try to set the exposure times so that the exposure level of the source and the sky is below 10,000 ADU. If this is the case, then you do not need to worry about non-linearity. If this is not the case, then you should think about how non-linearity may effect your data and if you really think that you should apply a non-linearity correction, then you are on your own at this point, because we have not developed any software to do this. The reason for this are twofold. Firstly, different parts of the array are read out at different times, so the non-linearity correction at row 512 is always going to be larger than the correction at row 1. Secondly, since the bias is a function of the flux, it is difficult to know the precise value of the signal to correct. This second obstacle is not as severe as you may at first think as the error you would introduce by ignoring the zero level offset will in most case be much smaller than the correction itself. If this does not deter you and you develop a way of correcting the non-linearity, then we'd be happy to hear from you.

The non linearity of the array can be fitted with the following function.

$$f_{\mathrm{true}} = f_{\mathrm{meas}} + 4.75e - 11 * f^3_{\mathrm{meas}}$$

Since we have not developed software to correct for non-linearity, this formula has not been tested.

4.3.5 Odd-even column effect

This effect can be seen as an offset between the odd and even columns of the array. It is a function of the flux of the array and it evolves with time. The amplitude also depends on the quadrant and can be as large as 6%.

Before March 30th, 2001, the effect was only visible when the flux on the array was above 10,000 ADUs. It was possible to avoid it by ensuring that the flux was never above this. However, after this date and for reasons which are unknown, the effect became visible at much lower flux levels. On July 24th 2001, the read speed of the array was slowed down. The odd-even column effect was reduced to less than 1% over the flux range 0-20000 ADU. As a direct consequence of this, the minimum integration time of the array was set 3.55 seconds.

On August 24th, 2001, the effect re-appeared, even with the slower read-speed.

On October 20th, 2001, during an instrument intervention, the problem disappeared. On December 15th, 2001 the effect re-appeared.

Since the effect depends on the quadrant, each quadrant has to be treated separately. A very effective way of removing the effect is to take a Fourier Transform of individual quadrants and mask the one pixel in Fourier Space that corresponds to a spatial frequency of one pixel. The effect of applying the correction on the photometry is less that a hundredth of a magnitude. This works very well for the lower right, upper left and upper right quadrants. It works less well for the lower right quadrant where there appears to be a y dependence to the effect. The y dependence can be removed by masking additional pixels.

An eclipse routine to remove the effect in imaging data has been created. It is called isaacp odd-even. The routine works less well for spectroscipic data.

The effect is harder to remove from spectroscopic data, and this is especially true for the flat fields. We have retaken all flat fields for all the spectroscopic set ups that were used between May 12th and July 24th 2001. If you suspect that your spectroscopic flats are contaminated by the odd-even column effect, then you should search the archive for data that was taken with the same instrument setup after July 24th.

4.3.6 Remnants from bright sources

If the array is exposed to a bright source, a remnant becomes visible. The amplitude of the remnant and the time it lasts depend on the brightness of the source. In broad band imaging, these remnants are not a problem as they are much fainter than the background. However, in spectroscopy, they are a nuisance, and, by Murphy's law, they are bound to land on your spectrum. We ask astronomers to check the brightness of their sources and in case where there are bright sources, we will arrange the schedule so that these sources are observed at the end of the night. However, there is little we can do if the acquisition image contains a bright source.

4.3.7 QE reduction from really bright sources!

If the source is really bright, then the QE of the array decreases. We do not fully understand this effect, but it would appear that the effect is strongest in the blue and that the only way of removing it is to warm up the detector.

4.3.8 50Hz pickup

The Hawaii array suffers from 50Hz pickup, which appears as lines that are almost aligned with detector rows. The strength of the pickup depends on how the array is readout and is a function of time. It is strongest in DCR and for some DITs in NDR. For NDR we have come up with a recommended list of

DITs (See the ISAAC web page) and users are encouraged to use these DITs. As an example, before April 2001 DITs of 60 and 180 seconds were strongly affected by the pickup. After this date, we tuned the number of reads so that the pickup with these DITs is weak.

If most cases, the 50 Hz signal will not be a problem, but if you find that it is, a very simple methyod to remove it was recently developed by Fernando Comeron. It turns out that the periodicity of the 50Hz signal is very close to 6.5 pixels in the vertical direction. To see how this is useful, take an image where the 50Hz signal is strong, shift it by 13 pixels in the vertical direction and subtract it from itself. The 50 Hz disappears. This simple procedure does not account for objects so a more sophisticated procedure is required. Such a procedure may work as follows.

- Divide the array into four quadrants and treat each quadrant separately.
- For each row, which we label n, average rows n-52, n-39, n-26, n-13, n+13, n+26, n+39, n+52, ... with suitable clipping to remove objects and subtract this average from row n. One may chose to average more rows that the row listed here.

This method is probably only applicable to images which already have been sky subtracted and special attention has to be paid when the row is near the quadrant borders.

For data taken after 24/7/2001, when the detector readout speed was changed, the periodicity of the 50 Hz pickup is about 3.25 pixels. The procedure to remove it remains unchanged.

4.3.9 Fringing

It is possible to see fringing on the detector at about the one percent level in spectra taken with the medium resolution grating. The fringes can be partially removed by taking flats immediately after the observations and/or by observing a telluric standard in the same place on the array as the science target. If neither of these strategies were employed, then fringing is difficult to remove.

Chapter 5

SW Imaging

This chapter is split into two sections. It first outlines how to reduce SW imaging data and then discusses the various frames that are needed for the reduction.

5.1 Reducing Science Data

The basic steps in reducing SW imaging data are:

- Removal of the odd-even column effect.
- Ghost removal
- Dark subtraction (*)
- Flat Fielding (*)
- Illumination Correction
- Flagging bad pixels, removing vignetted regions. (*)
- Sky subtraction (*)
- Removing the 50 Hz pickup
- Registration and stacking (*)
- Removal of residual bias variations (*)

Steps indicated with a star are supported by the `jitter` command.

5.1.1 Odd-even column effect

In most cases, the odd-even column effect is small enough that it can be ignored. However, if the flux level of the exposure is above 10,000 or if the data was taken during those occasions when it was strong, users should think seriously about removing it. See Sec. 4.3.5.

The `isaacp` command features a recipe called `oddeven`, that takes care of removing the odd-even effect from a frame. While it works reasonably well to cover for most of the effect, it is no silver bullet. Check out the results coming out of this recipe. This recipe has been made available in eclipse since version 4.1.1 (December 2001).

5.1.2 Ghost removal

The ghost that was described in the general introduction should be removed, especially if there are bright objects in the field of view. This step can be done with the **eclipse** recipe `ghost`.

5.1.3 Dark subtraction

One subtracts the dark frame from all images. Since the zero level offset (or bias) is a function of the flux and since the bias is somewhat variable, this subtraction is not perfect. The residual is removed in the subsequent steps.

The behavior of the bias as a function of the flux and DIT is very complex. We are investigating ways to model the bias so that this step is more precise.

5.1.4 Flat fielding

One divides all images with a flat field which has been normalised to unity. A more detailed discussion on how the flat field can be created is given in Section 5.3. This will result in photometry that is consistent to the 2% level over the ISAAC field of view. If more accurate photometry is required, then an illumination correction should be applied.

If one was to look at the flat fielded images, one would see that they are far from flat. One would see a jump between the two halves of the array, and one would perhaps see structure at intermediate (5-10 pixels) and large (several hundred pixels) scales. These structures have a variety of causes. The jump in the middle is caused by the fact that we have not removed the zero level offset perfectly. The structures at intermediate scales are probably caused by pupil ghosts and by dust that has moved. The structures at large scales are probably caused by scattered light. Most of these features are additive, so they are removed at the sky subtraction stage.

5.1.5 Illumination correction

This can be applied directly to the data or to the flat field itself, in which case it is the inverse correction and it is applied to the flat before the data is flatfielded. See Section 5.4 about how to create and use illumination corrections.

`jitter` does not take into account illumination frames during the reduction process so you would have to correct the flat-field beforehand to take this into account.

5.1.6 Bad pixels and vignetted regions.

Before April 2001, the top 20-30 pixels of the ISAAC image were vignetted. This vignetting was removed during the March 2001 intervention. So, for data taken before April 2001, this region can be trimmed without any loss of information.

There are a lot of bad pixels on the right hand edge of the array. Users may want to trim the 10 last columns of the ISAAC image or alternatively flag them as bad.

The `jitter` command can replace bad pixels by an average of their valid neighbors. It cannot trim whole regions.

5.1.7 Sky subtraction

This is the most important step and great care and a good understanding of the technique are necessary if good results are required. This is particularly important for deep imaging as an error at the 0.01% level will significantly effect the photometry of the faintest sources.

There are two cases. In relatively blank fields, the sky is created from the object frames themselves. For crowded fields or large objects, the sky is created from frames that were specifically taken to measure the sky.

For deep exposures, the sky is computed from a subset of exposures and there will be one sky frame for each object frame. For accurate photometry, it is very important that the object frame is not included in the frames that are used to compute the sky. This is a weakness of the current **eclipse** recipe `jitter`. For H and K band observations the sky frame should be computed from frames that were taken 5-10 arc minutes either side of the object frame. For J band observations, these numbers can be doubled. For conditions where the sky background is varying rapidly (clouds or observations taken just after evening twilight) a more rapid sampling of the sky is necessary.

All sky frames contain objects, so one has to combine them with suitable clipping. A robust method is to first scale frames to a common median and then remove objects by rejecting the highest and lowest pixels. Rejecting the two highest and two lowest pixels would produce even more robust results. The remaining pixels are then averaged (the median can also be used, but it is a noisier statistic). The resulting sky frame is then scaled (to match the median of the object frame) and subtracted.

A more sophisticated approach is to do the sky subtraction in two steps. The first step reduces the data as described above, produces the combined image and then locates all the objects. In the second step, the data is rereduced with the knowledge of where the objects are. These objects are then excluded when the sky is estimated in the second pass. This is the approach used by the XDIMSUM package in IRAF and for very deep imaging it is the recommended package.

5.1.8 Removal of the 50Hz pickup

In cases where the background level is low, it may be possible to see the 50Hz pickup. In most cases, the pickup will average out when the images are registered and stacked. If you want to remove the pickup, then it is at this point of the reduction sequence it should be done. A description of how the pickup could be removed is provided in 4.3.8.

5.1.9 Image registration and stacking

To register the sky-subtracted images to a common reference, it is necessary to precisely estimate the offsets between them. `jitter` applies a 2d cross-correlation routine to determine the offsets to an accuracy of 1/10th of a pixel. There are other ways to find out offsets between frames: with many point-sources, point-pattern matching is a possibility. Identifying the same objects in all consecutive frames would also yield a list of offsets between frames. An initial estimate of the offsets between frames can be found in the FITS headers. `jitter` assumes that the offsets found in the input FITS headers have a certain accuracy. If there are no input offsets, they are all initially estimated to be zero in both directions.

Registering the images is done by resampling them with subpixel shifts to align them all to a common reference (usually the first frame). Resampling can make use of any interpolation algorithm, but be aware that using cheap and dirty algorithms like nearest-neighbor or linear interpolation can degrade the images by introducing aliasing. `jitter` offers many higher-order interpolation kernels that introduce few or no artifacts; however, the noise (high frequencies) will be smoothed a little bit.

Stacking the resulting images is done using a 3d filter to remove outliers and `jitter` gives you a choice between 3 different filters. *Linear* means that all frames are actually averaged without filtering (pass-all filter). This is not recommended as this is likely to keep cosmic rays and other outliers in the final frame. *Median* means that the final frame is the median of all resampled frames. The last filter (default) scales all frames by their medians and removes the highest and lowest pixel values before taking an average. See the `jitter` documentation for more information.

Notice that in versions later than eclipse version 4, `jitter` resamples and stacks in one step to speed up the process. Also added since version 4, the final frame is a union of all input images (as opposed to an intersection for previous versions), which means that it is bigger than any of the initial input frames.

5.1.10 Removing the remaining bias residuals

In most cases, the final combined image will contain small but noticeable jumps in the vertical direction. This is caused by the imprecise removal of the bias when the dark was subtracted. The jumps can be removed very effectively by averaging the image along rows and subtracting the resulting one dimensional image from each column of the original image. To make sure that objects do not bias the result, one clips the 200 highest and lowest pixels from the computation of the one dimensional image.

5.2 Darks

Dark frames are exposures without detector illumination. The dark current of the ISAAC SW detector is small, so the dominant feature in these frames is the detector bias, which is also called the zero level offset, since it is not possible to take a zero second exposure with the SW array.

Usually one takes at least three darks and combines them with suitable clipping to create the dark frame that is subtracted from the science data. As the bias is a function of the DIT, the DIT of the science data and that of the dark must match.

Dark frames are acquired through a dedicated template, which obtains (usually at the end of the night) at least three dark frames for each DIT that was used during the night.

The darks can be reduced with the **`dark`** recipe (in **`isaacp`**). This recipe will produce one dark frame for each DIT. The readout noise is also measured from these frames and the information is written to a readable ASCII file in PAF format.

The **`dark`** recipe is actually separated into two sub-recipes, the first one creates the dark and the second one computes the readout noise. Each sub-recipe can be called individually by using the **`dark-avg`** and **`dark-ron`** recipes in **`isaacp`**.

Currently, the dark created by **`dark-avg`** is nothing more than an average of the input files (after having sorted them by identical DIT). Cosmic ray rejection will be implemented later.

5.2.1 Read-out noise computation

This particular recipe is used for quality control and instrument monitoring.

The input list of frames is browsed in the order they are received. For every pair of consecutive frames that share the same DIT, NDIT, and readout mode, the readout noise is computed. Notice that the recipe does not try to make all possible pairs from the input list, so it is important to make sure that the files are correctly sorted. If the frames are kept in the order they were acquired, this should not be a problem.

For each pair, the second frame is subtracted from the first. The resulting image is then cut out into 4 quadrants and on each quadrant the following measurement is applied:

- Generate 100 9x9 windows on the input pixel surface. These windows are optimally scattered using a Poisson distribution to make sure they sample the whole area with as little overlap as possible.
- Compute the pixel standard deviation in each window.
- The readout noise is the median of all these measured standard deviations.
- The error on the readout noise is the standard deviation of all measured standard deviations.

The results are written into a PAF file (readable ASCII). One value per quadrant is printed out.

Notice that the algorithm is identical in LW mode, but the whole detector is considered instead of cutting it into four quadrants.

5.3 Twilight Flats

ISAAC imaging data from the SW imaging arm are flat fielded with twilight flats. The flats are derived by imaging a region of the sky relatively free of stars. Between 10 to 25 exposures with constant DIT and NDIT are taken for each filter and a robust linear fit between the flux in each pixel and the median flux of all pixels is used to produce the flat field.

UT1 does not have a flat field screen, so it is not possible to take dome flats.

5.3.1 Eclipse implementation

The implemented algorithm is the following: for each pixel on the detector, a curve is plotted of the median plane value against the individual pixel value in this plane. This curve shows the pixel response from which a robust linear regression provides a pixel gain. The image showing all pixel gains (i.e. the flat-field) is normalized to have an average value of 1.

By-products of this routine are: a map of the zero-intercepts and an error map of the fit. Both can be used to visually check the validity of the fit. A bad pixel map can also be produced by declaring all pixels whose value after normalization is below 0.5 or above 2.0 as bad.

The **eclipse** recipe is called `twflat`. Check out the manual page (`isaacp man twflat`) for more details about how to use the command.

5.3.2 Limitations

The method of using twilight flats to flat field the data suffers from several limitations.

Non-linear change in the intensity of the twilight sky with time

Since the sky is either decreasing in intensity (evening twilight flats) or increasing in intensity (morning twilight flats), since these changes are non-linear in time and since the array takes some time to read-out, one can get a discontinuity in the flat field between the two halves of the array which has an amplitude of 1% with the current DIT and NDIT settings.

The effect can be removed rather well by averaging the morning and evening twilight flats as the effect inverts itself between the two times.

Odd-even column effect

The odd even column effect should be removed from the individual flat fields. However, this only needs to be done for flat fields that were taken when the effect was strong. (See Sec. 4.3.5).

The variable zero level offset

The zero level offset of the Hawaii array is a function of the flux, so when the flux of the array changes so does the offset. The change appears to be complex. As the flat is made from exposures of varying flux and since the zero level offset changes with flux, we are not measuring the relative sensitivity of a pixel, but more the relative sensitivity plus the change in the zero level offset.

At this point in time, we cannot remove the change in the zero level offset and this ultimately limits the accuracy at which the relative sensitivity of pixels and hence the flat field can be measured. Nevertheless, we can quantify the effect this has on the flat field by dividing one flat by another. The typical pixel to pixel RMS in the flat field is of the order of 6%. The typical pixel to pixel RMS of an image which is the division of one flat by another is 0.4-0.6%. This translates to an accuracy of 0.3-0.4% in the flat itself.

Saturation

If the array is heavily saturated, two affects occur. Firstly, a remnant, whose timescale depends on the level of saturation, appears and, secondly, the QE of the array decreases. This has a much longer timescale, and appears to be a strong function of wavelength. One sees it strongest at J and weakest at Ks.

The solution to this problem is to not saturate the array, but this will occur from time to time, so flats need to be taken often enough so that the scientific data is not affected. Since the timescale of the effect is long, the current practice of taking flats as required is satisfactory.

5.3.3 Accuracy and period of validity

When describing the accuracy of the flat field, one describes it over two length scales: on the scale of individual pixels and on the scale of the entire array. As discussed above, the pixel-to-pixel accuracy is around 0.2-0.3%. On the scale of the entire array, the accuracy of the flat field is of the order of 1-3%. This is measured by placing a bright standard over several positions of the array and by measuring the flux of the standard after the data has been flat fielded (this test is sometimes called "the thousand points of light test"). Alternatively one can use the science data itself, if dithering has been used and if there are enough sources that are sufficiently bright. The large scale accuracy of the flat field can be improved to the 1% level by using an illumination correction.

Scientific data should be normalised with flats that were taken within one week of the scientific data. Beyond this, the relative accuracy at which pixels can be normalised slowly degrades.

5.4 Illumination Corrections

The illumination correction takes into account low-frequency differences between the true flat and the twilight flat. These variations have an RMS of 2%. If this is accurate enough, then an illumination correction is not needed. To compute these variations, a dedicated calibration template is offered. The procedure images a standard star over a grid of 17 positions (1 central and a 4x4 grid), with the convention that the first frame has the star at the center of the image ("center" means up to 50 pixels from the true image center), and subsequent positions are indicated through keywords in the FITS header. Usually, the procedure is run once every six months and after major instrument interventions.

These images need to be sky subtracted and flat fielded. The easiest way to remove the sky (and the dark) is to subtract one frame from another. For flat fielding, one can either use the flat that will be later used to flat field the science frames or the flat that was taken closest in time to when the standard star was observed. The advantage in using the second flat is that pixel to pixel variations are better normalised, but one has to take into account the large scale differences between the two flats by smoothing the flats over a suitably large scale, say 100 pixels.

The next step is to compute the flux (not magnitude) of the stars through some fixed aperture and to fit a two-dimensional, 2nd or 3rd order polynomial to the flux values as a function of position on the array. The IRAF routine surfit is particularly useful for doing this. The surface which is fit is the illumination correction (after it has been normalized to a mean of unity).

The illumination correction can be applied to either the data or the flat field itself; however, some care is required in determining if one needs to multiply or divide. It depends on how the fit was done.

We are creating an eclipse routine, called `isaacp illum`, which will do all of this; however, the routine is still under development.

5.5 Zero Points

Standard stars are observed every night in the J, Js, H and Ks filters. For the NB filters, standards are observed as required.

Standard stars are imaged over a grid of five positions, one just above the center of the array and one in each quadrant. The recipe finds the standard (it assumes that the star in the first image is near the center), computes the instrumental magnitude, and then uses the standard star database to determine the ZP, which is uncorrected for extinction.

The standard star database contains about 1000 stars with magnitudes in the J, H, K, Ks, L and M bands, although most stars only have magnitudes in a subset of these filters. The standard star database is currently accessible through an eclipse command called `std`. In the (near) future, we will provide a web interface for accessing this list.

If you want to check out which stars are present in the ISAAC database, use the `std` command. You can search the database by name, position, magnitude range, catalog name, etc. Stars are currently taken from the following catalogs:

- Arnica
- ESO Van der Bliek
- LCO Palomar
- LCO Palomar NICMOS red stars
- MSSSO Photometric
- MSSSO Spectroscopic
- SAAO Carter
- UKIRT extended
- UKIRT fundamental

The implemented recipe is the following:
For any couple of consecutive images `(image1, image2)`:

1. `diff = image1 - image2`
2. Locate in `diff` the star around the expected pixel position (provided by the FITS header or by an external offset list).
3. Compute the background around the star, and the star flux.

4. Store the flux result in an output table.

Apply steps 2 to 4 to the inverted image `image2-image1`. This yields 2(N-1) measurements for N input frames. From this statistical set, the highest and lowest values are removed, then an average and standard deviation are computed. The conversion formula from ADUs to magnitudes is:
$zmag = mag + 2.5 * log_{10}(flux) - 2.5 * log_{10}(DIT)$
where:

- $zmag$ is the computed zero-point.
- mag is the known magnitude of the standard star in the observed band.
- $flux$ is the measured flux in ADUs in the image.
- DIT is the detector integration time.

Note that neither the extinction nor the colour correction are included in the ZP. The average airmass is given in the output result file, together with individual airmass values for each frame.

The average extinction on Paranal for the J, Js, H, Ks and NB_M filters, is available from the ISAAC web pages.

The correspondence between the filter in which the observations were taken and the filter in the standard star table is listed below. The two filters are reported in the QC parameters: `QC.FILTER.OBS` and `QC.FILTER.REF`. This correspondence completely ignores corrections due to filter mismatch, and, in some cases, these corrections are substantial.

ISAAC filter	Reference filter
Z	J
SZ	J
Js	J
J	J
SH	H
H	H
K	K or Ks
Ks	Ks or K
L	L
M	M

ISAAC narrow band filter	Reference filter
NB_1.06	J
NB_1.08	J
NB_1.19	J
NB_1.21	J
NB_1.26	J
NB_1.28	J
NB_1.64	H
NB_1.71	H
NB_2.07	Ks
NB_2.09	Ks
NB_2.13	Ks
NB_2.17	Ks
NB_2.19	Ks
NB_2.25	Ks
NB_2.29	Ks
NB_2.34	Ks

5.5.1 Colour terms

The match between the ISAAC filters (J, H and Ks) and those used to establish the faint IR standard star system of Persson et al. (AJ, 116, 2475) are quite good, so we expect the colour terms to be small. If the zero point is expressed in the form

$$ZP_X = -x_{\mathrm{ISAAC}} + X_{\mathrm{LCO}} + a * (J - K)_{\mathrm{LCO}}$$

where x_{ISAAC} is the ISAAC instrumental magnitude, X_{LCO} is the catalogue magnitude and, for simplicity, we have ignored extinction, we expect that a will differ from 0 by less than 0.01 for X being one of J, H or Ks. The Js filter of ISAAC has a slightly redder effective wavelength than J, so we expect a non-negligible colour term. Theoretically, we expect $a = -0.04$.

However, the colour transformation between ISAAC magnitudes and those of LCO have never been experimentally verified, so the results quoted here should be considered with this in mind.

Chapter 6

SW Spectroscopy

6.1 Introduction

This chapter is split into two sections. The first section describes the basic steps in going from the raw data to the wavelength and flux calibrated spectrum. The second section describes the eclipse routines that are available for producing calibration products and for reducing the data.

6.2 SW Reduction - The Basic Steps

The most basic way of taking IR spectra is to observe the target along two slit positions. The sky is then removed by a process which is sometimes called double sky subtraction. The basic steps of how to reduce these type of data are:

- Ghost Removal
- First subtraction
- Removal of the 50 Hz pickup
- Flat fielding
- Slit Curvature Correction and wavelength calibration
- Combining spectra
- Extraction
- Telluric line correction
- Flux calibration

6.2.1 Ghost removal

The ghost that was described in the section 4.3.3 needs to be removed, as the equivalent width of absorption lines (both those caused by the atmosphere and those intrinsic to the object) and the continuum are affected. This step can be done with the **eclipse** recipe `ghost`.

6.2.2 First subtraction

There are two techniques observers use in taking IR spectra. There is the classic sequence where one observes the object at two slit positions. i.e. ABBA, etc, and there is the more complex case where one has observed the target along several slit positions.

In the classical case, one simply subtracts frames taken at different slit positions. So one needs to form A-B, B-A, etc. This simple step removes the bias and results in an image with two spectra, one positive and one negative.

In the more complex case, one could build a sky frame from several spectra, as one does when building the sky frame in imaging. This results in an image with only one positive spectrum.

6.2.3 Removal of the 50 Hz pickup

In cases where the background level is low, it may be possible to see the 50Hz pickup. If you want to remove the pickup, then it is at this point of the reduction sequence it should be done. A description of how the pickup can be removed is provided in section 4.3.8.

6.2.4 Flat fielding

As in imaging one divides by the flat field.

For data taken between May 12th and July 24th 2001, the odd-even column effect is strongly present. It is difficult to remove from the spectroscopic flat fields, so the observatory has retaken flats for all spectroscopic setups that were taken during these dates. These flats are available from the ESO archive.

Observations with the medium resolution grating show fringing at the one percent level. If the flat was not taken immediately after the observations and if the telluric standard was not observed at the same place on the array, then fringing can be difficult to remove.

6.2.5 Slit curvature correction and wavelength calibration

ISAAC spectra are strongly curved and tilted. Before the 2d spectra are combined, they need to be straightened. It is useful to do the wavelength calibration at the same time, so that the horizontal axis is in wavelength units.

The wavelength scale can be calibrated with either arc frames or the OH lines that are imprinted on each spectrum. The advantage of the arcs is that there are lines covering the entire 0.9 to 2.5 micron range. The disadvantage is that the arcs are taken separately and, in most cases, this means that the grating has moved between the time the target was observed and the arcs were taken. One can use the OH lines to cross check and correct the zero point of the wavelength calibration, which will be a necessary step in most cases.

The advantage of the OH lines is that they are numerous and that they lead to a slightly more accurate wavelength calibration. The disadvantages are that: in some regions, particularly beyond 2.2 microns, there may be too few lines to do a good fit; in standard star observations, where exposure times are short, the OH lines may be too faint; and in LR observations, where the resolution is low, the OH lines may be heavily blended.

For both arcs and OH lines a 3rd order Legendre (4 terms) gives a good description of the dispersion.

6.2.6 Combining 2d spectra

For the classical ABBA technique, one multiplies each image by -1 and adds it back to itself after a suitable shift. This method of combining data is often called double sky subtraction as it effectively removes any residual sky that remains after the first subtraction. It results in an image that has one positive spectrum and two negative spectra either side of the positive spectrum.

In the more complex cases, one combines the individual spectra after suitable shifts have been applied.

6.2.7 Extraction

For the classical ABBA technique, one should extract the spectrum without fitting the sky. Fitting the sky only adds noise. For more complex cases, a low order fit to the sky may be required.

6.2.8 Removing telluric lines

This is a critical step that requires some care and, possibly, some experimentation. The aim is to remove the large number of telluric lines that appear in IR spectra. This is done by dividing the object spectrum with that of a telluric standard. Since this is a division of one spectrum by another, it is important that the strength, shape and centroid of the telluric lines match.

First and foremost, the telluric standard and the object have to observed with the same instrument setup, with roughly the same airmass and, if possible, consecutively.

Secondly, the object and science data should be reduced in the same way and with the same calibration frames.

For the best results, one may have to modify the spectrum of the telluric standard so that the center and strength of the telluric lines match those of the object spectrum.

The next step is to remove spectral features that have been imprinted onto the object spectrum from the telluric standard itself.

Telluric standards are either hot stars or solar type dwarfs. Both types contain spectral features that should be removed. For solar type stars, one can use the observed solar spectrum to remove the features. This can be tricky if the spectral resolution of the instrument is a function of the wavelength (which is the case for ISAAC) as it means that the kernel for convolution also has to be a function of wavelength. The arc spectra and the OH lines can be used to estimate what this function is.

Hot stars usually contain helium and hydrogen lines. If the spectral regions around these lines are of interest, then one should think carefully about using these type of stars. If the resolution is high enough, which is certainly the case for MR observations, one can try to remove these lines by fitting them. Alternatively, one can use a second telluric standard that does not have helium or hydrogen lines so that these lines can be removed from the hot star.

The ISAAC FITS header does not always contain the full target name of the telluric standard and, sometimes, operators forget to include it in the night report. To find out which telluric standard was used, look at the RA and DEC of the target and consult either the ISAAC manual or the list of Hipparcos stars that are often used as telluric standards.

6.2.9 Flux calibration

The first step is to obtain a relative flux calibration. The second step is to do absolute flux calibration.

If the telluric standard was a hot star, then a blackbody curve can be used to model the continuum of the standard. The spectral type of the star can be used to give an idea of what temperature to use. The blackbody curve is then multiplied into the object spectrum.

For solar type stars, a blackbody curve is a good enough description of the spectral energy distribution above 1.6 microns. Below 1.6 microns, a more accurate description of the continuum is required. The spectral energy distribution of a wide variety of stars are available through the ISAAC web pages.

The second step is absolute flux calibration. If the magnitude of the target is known and if the spectra were taken with the LR grating, a reasonably accurate calibration can be obtained by convolving the spectrum with the filter curve and determining the appropriate scaling. If neither of the above are true, then determining the absolute flux calibration is more difficult and less certain. Most ISAAC standard star observations are done with the slit the object was observed with and either the 2" slit or slitless, so slit losses can be roughly estimated. If the standard was not observed with the 2 arc second slit or slitless, slit losses

Recipes to produce calibrations and to do quality control	
isaacp sw_spflat	Creates spectroscopic flat fields
isaacp sw_arc	Models the slit curvature
isaacp slitpos	Measures slit positions
isaacp sw_startrace	Models the trace of a star as a function of y
isaacp sw_respfunc	Measures the spectroscopic throughput
Recipes to reduce the data	
isaacp sw_spjitter	Reduces ISAAC spectroscopic data

can also be roughly estimated from the image quality in the acquisition image. Alternatively, if the seeing did not change too drastically while the target and the standard were observed, slit losses can be ignored.

6.3 Pipeline Recipes

This section describes the **eclipse** recipes that are used in the ISAAC pipeline. The recipes have a variety of purposes, from recipes that take quality control data to recipes that produce reduced data. Not all recipes will be of direct interest to people wishing to reduce ISAAC data.

The principle purpose of these recipes is to produce and use calibration products that can be used to reduce ISAAC data. These recipes produce products such as flat fields, fits to slit curvature, wavelength calibration, etc.

6.3.1 Spectroscopic flat fields

The flat is the normalised difference of two images, one with the lamp on and another with the lamp off. In general, three pairs of images are taken.

The **eclipse** recipe that produces the master flat fields that are given to the users is called *isaacp sw_spflat*. The input file(s) is an ASCII file containing an even number of FITS files. The list consists of a series of image pairs. Each pair consists of a exposure with the lamp on and one with the lamp off.

The recipe starts by classifying images by instrument setting. A setting is defined by the slit used, the central wavelength and the grating name, which are obtained from the FITS header. Each pair of on-off spectra are then reduced as follows:

- subtract the off frame from the on.
- normalize the whole frame with the computed mean of the center part of the difference frame.
- fit a polynomial in the slit direction, and divide the frame by the polynomial. The aim of this step is remove the lamp shape in the slit direction, since the illumination of the slit is not uniform.

- Average the results if there is more than one on-off pair.

The algorithm is illustrated in figure 6.1.

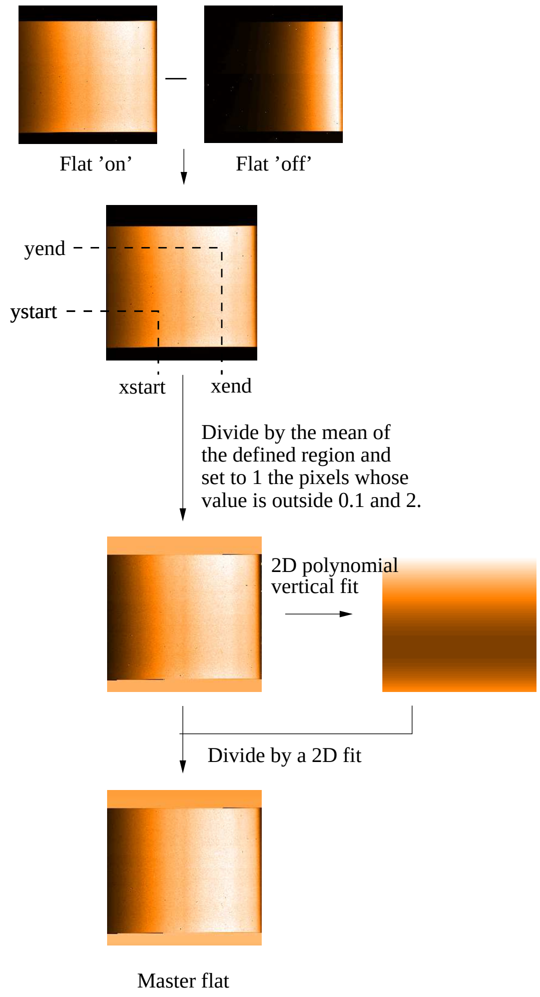

Figure 6.1: Flatfield reduction

The output consists of one master flat for each instrument setting and a PAF file. The master flats are named [outname]_X.fits where X goes from 1 to the number of settings and *outname* is the basename used for output files (specified by the user or defaulted to the input file basename).

The PAF file is an ASCII file containing keywords for quality control.

A weakness of the recipe is that a residual bias remains after the exposure with the lamp off is subtracted from the one with the lamp on. This is a fundamental property of the array - the bias is a function of the amount of flux that the array sees. It may be possible to use the parts of the array that are not exposed to light to partially remove the residual bias; however, this is not yet part of the recipe.

At some wavelengths, the spectral response of the lamp is a very strong function of wavelength, and it is possible, in extreme cases, for the frame with the lamp off to have a higher level in some regions that the frame with the lamp on. In these cases, it is better to estimate the bias from the regions which do not have direct flux, i.e. the 50 or so pixels at the top and bottom of the image.

The recipe does not remove the wavelength dependence of the lamp from the flat field. This is removed when the telluric standard is divided into the spectrum of the science target. Implicit in this statement is that the same flat was used for both the target and the standard and that the target and the standard were observed at roughly the same place on the array. If neither of these things are true then flux calibration (relative and absolute) will be less accurate.

6.3.2 Arc recipe

For a given instrument setting, arcs consist of two or three exposures; one with the arc lamps off and additional exposures with one or both of the arc lamps on. The lamps are Xenon and Argon. The arcs are used to model the slit curvature and to derive the wavelength calibration.

The **eclipse** recipe that analyses the arcs is called *isaacp sw_arc*. The input file(s) can either be a FITS file or an ASCII file containing a list of FITS files. If the input is a FITS file, one can select the line catalog. If the input is an ASCII file, then the value of specific keywords in the FITS headers are used to select the line catalog.

The recipe starts by classifying images based on instrument setting: a setting is defined by the resolution, the central wavelength and the slit.

The slit curvature is modeled with a bivariate 2-d polynomial. If we let the distorted image be expressed in (u,v) coordinates, and the corrected image in (x,y) the curvature is modeled with:

$$u = a + bx + cy + dx^2 + ey^2 + fxy; v = y$$

The dispersion relation (1st order) is computed by matching a xenon and/or argon atlas with the corrected spectra, which can be produced with the -c option. This is a weakness of the current version of the routine as a 3rd order fit provides a better description of the dispersion.

Users familiar with IRAF and/or MIDAS would notice that the correction for slit curvature and the wavelength calibration are done in two steps with this recipe, whereas using IRAF or MIDAS users can do it in one step.

The procedure produces two output files: a FITS table, which contains the fit to the slit curvature and the linear dispersion relation; and a PAF file, which contains the same information as the FITS file as well as additional quality control information.

The output FITS table is named [outname]_arc_setX_Y_Z.tfits where X is the setting number, Y the resolution (LR or MR), and Z the arc lamp (Xe, Ar, Xe+Ar), e.g. infile_arc_set2_MR_Xe+Ar.tfits. The corresponding PAF file would be called infile_arc_set2_MR_Xe+Ar.paf.

With the -c option, the recipe will produce the curvature corrected spectrum. Figure 6.2 shows a raw image and the corresponding corrected one.

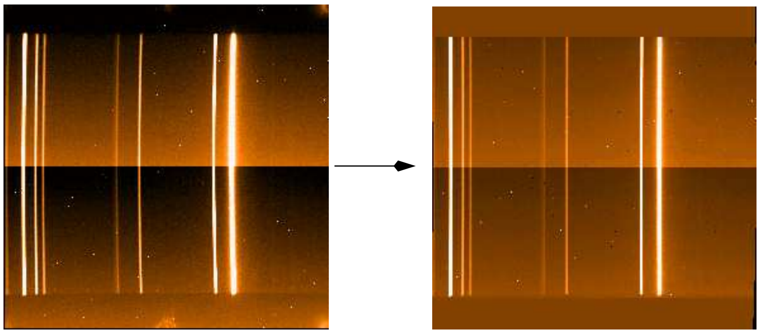

Figure 6.2: Slit curvature correction

6.3.3 Slit position recipe

On a regular basis, the observatory takes images of every slit through every filter. These images are processed with the eclipse recipe *isaacp slitpos* which measures the slit position.

The input file can be either a FITS file or an ASCII file containing a list of FITS files. If the input is a FITS file, it is assumed to be a slit image. In the latter case, the first file may be a *dark*, which would then be subtracted from all subsequent images.

The output consists of two files: a FITS table which records the location of the slit in great detail, and a PAF file that contains the slit angle and slit position. Figure 6.3 shows the how the FITS table is formatted.

6.3.4 Star trace recipe

On a regular basis, the observatory takes data that can be used to determine the tilt of spectra as a function of position along the slit and the relationship

SWI_slit_03_tilt_Js_S2_slitpos.tfits				
Sequence	Y	LEFT_POSITION	CENTER_POSITION	RIGHT_POSITION
1	5.6000e+01	5.2817e+02	5.3008e+02	5.3200e+02
2	5.7000e+01	5.2817e+02	5.3008e+02	5.3200e+02
3	5.8000e+01	5.2817e+02	5.3008e+02	5.3200e+02
4	5.9000e+01	5.2816e+02	5.3008e+02	5.3200e+02
5	6.0000e+01	5.2816e+02	5.3008e+02	5.3200e+02
6	6.1000e+01	5.2816e+02	5.3008e+02	5.3200e+02
7	[illegible]	[illegible]	[illegible]	[illegible]

Figure 6.3: Table created by the slit detection

between the position of an object along the slit and the corresponding location of the spectrum on the array.

The data consists of n images of a bright star stepped along the slit and spectra of the star in the MR and LR gratings at each of these positions. In total, n images and $2 * n$ spectra are obtained.

The eclipse recipe that processes this data is called *isaacp sw_startrace*. The input file consists of an ASCII file containing the data described above. The input frames can be corrected for slit curvature at the very beginning via a command line option. In this case, the user has to provide the ARC table containing the model of the slit curvature.

The recipe produces several files.

1. Positions table. The location of the stellar images and spectra are found and stored in a table named [outname]_positions.tfits.

2. Positions correspondence table. The relationship between the position of the star on the slit and the location of the corresponding spectra are fitted with a polynomial. There is one polynomial for LR and one for MR and the co-efficients of both are stored in the table called [outname]_corresp.tfits.

3. Extracted spectra. The spectra are traced, extracted, wavelength calibrated and stored in a table named [outname]_extracted.tfits. The spectra are traced with a 3rd order polynomial and the results are stored in [outname]_shapes.tfits.

4. A table with 2d polynomial modelling the slit tilt. For each grating, the spectral tilt is modelled with a 2d polynomial, which is written to a table named [outname]_poly2D_XR.tfits. This table is the one used by *spjitter* to correct the spectral tilt.

An example of each table is given in figure 6.4 and a flowchart of what the recipe does is given in figure 6.5.

6.3.5 Response Function

On a regular basis, the observatory observes a standard star that has a reasonably well characterised spectral energy distribution. The purpose of the *sw_respfunc* recipe is to use this data to calculate the instrument throughput. It is used for quality control purposes and observers will have little use for this routine.

The recipe takes the FITS image created by *isaacp sw_spjitter*, extracts the spectrum, correlates the RA and DEC in the FITS header with a star in the catalogue and calculates the efficiency and the conversion curves.

The output consists of four files:

- [outname]_extr.tfits is the extracted spectrum
- [outname]_back.tfits is the background spectrum
- [outname]_conversion.tfits is the conversion curve
- [outname]_efficiency.tfits is the efficiency curve

6.3.6 SW_SPJITTER

The pipeline recipe that reduces SW spectroscopic data is called *isaacp sw_spjitter*. The recipe uses an initialisation file, which contains parameters that define how the recipe runs. To generate one, type *isaacp sw_spjitter -g spjitter.ini* at the prompt and this will create an initialisation file called spjitter.ini. This file can be edited and modified by the user to setup some parameters.

The initialisation file contains parameters which indicate

- which table to use to correct for slit curvature
- which table to use for wavelength calibration
- which table to use to correct for spectral tilt (the one produced by the startrace recipe) and
- which flat field to use

The input file should be an ASCII file listing the raw FITS files.

The recipe starts by classifying the input images according to the cumulative offsets in the headers. The classic way of taking IR spectroscopic data is to observe the target along two positions along the slit, which we will call A and B. An example, may be the sequence AAABBBBBBAAA. After flat fielding all the data, the recipe will take the first three A frames and average them, take the first three B frames and average them, etc. The recipe then subtracts one average from the other, corrects for slit curvature, spectral tilt and wavelength calibrates.

If the tables for the wavelength calibration and the correction for slit curvature are missing, the recipe will use the OH night sky lines. If there are too few of these, the recipe will use a model to do the wavelength calibration and will skip the correction for slit curvature.

If the flat field is missing, the recipe skips the flat fielding step.

The subtracted frames will contain positive and negative spectra. The two spectra are combined by multiplying the image by -1 and adding it to the original after a suitable shift. The resulting frames are then added together to give the final result.

At the end, a spectrum can be extracted. Either the user specifies the position of the spectrum they want to extract in the initialisation file, or the spectrum of the brightest object is extracted.

Figure 6.6 shows the individual steps of the recipe.

The output consists of the reduced FITS image, an ascii file which logs the reduction steps (and informations on how they were performed) and the extracted spectrum (TFITS), if it was found.

Correspondance table:

Sequence	Imaging_LR	Imaging_MR	
1	9.3040e+02	9.5946e+02	→ a
2	-9.8629e-01	-9.8070e-01	→ b
3	-2.7470e-05	-3.2096e-05	→ c

Y: star position
X: spectrum position

$Y = a + b.X + c.X^2$

Positions table:

Sequence	Star_positions	Spec_LR_positio	Spec_MR_positio
1	7.5099e+02	-1.0000e+00	2.0450e+02
2	6.2755e+02	7.9950e+02	3.3200e+02
3	5.0372e+02	6.7550e+02	4.5750e+02
4	3.8008e+02	5.5150e+02	5.8150e+02
5	2.5679e+02	4.2700e+02	7.0550e+02
6	1.3211e+02	3.0000e+02	8.2950e+02
7	1.7856e+01	1.7450e+02	-1.0000e+00

-1.0: Not detected

Extracted spectra table:

Sequence	Wavelength_LR	LR_1	LR_2		LR_7	Wavelength_MR	MR_1	MR_2	
1	0.0000e+00	0.0000e+00	7.9950e+02		1.7450e+02	0.0000e+00	2.0450e+02	3.3200e+02	
2	1.8380e+03	0.0000e+00	2.2830e+03		3.3356e+03	2.1395e+03	5.7650e+03	6.1230e+00	
3	1.8387e+03	0.0000e+00	2.4323e+03	- -	3.3812e+03	2.1396e+03	5.8479e+03	6.4838e+03	- -
4	1.8394e+03	0.0000e+00	2.5821e+03		3.8706e+03	2.1397e+03	5.8315e+03	6.4125e+03	
5	1.8401e+03	0.0000e+00	3.5094e+03		4.2604e+03	2.1399e+03	5.7721e+03	6.0592e+03	
6	1.8408e+03	0.0000e+00	4.3406e+03		4.4306e+03	2.1400e+03	5.6828e+03	5.9748e+03	
7	1.8415e+03	0.0000e+00	4.9509e+03		4.6560e+03	2.1401e+03	5.6803e+03	5.7643e+03	
⋮		⋮				⋮			
1022	2.5590e+03	0.0000e+00	4.1714e+02		1.7086e+02	2.2605e+03	3.2533e+03	3.3366e+03	
1023	2.5597e+03	0.0000e+00	3.6349e+02	- -	1.3115e+00	2.2606e+03	3.2440e+03	3.6316e+03	- -
1024	2.5604e+03	0.0000e+00	3.8426e+02		3.6143e+01	2.2607e+03	3.3435e+03	4.0579e+03	
1025	2.5611e+03	0.0000e+00	3.6430e+02		2.4560e+00	2.2609e+03	2.4540e+03	1.4540e+03	

The first line is the spectra global positions. A column of '0.0' for non detected spectra. Wavelength in microns.

Spectra shapes:

Sequence	LR_1	LR_2		LR_7	MR_1	MR_2		MR_6	MR_7
1	0.0000e+00	7.9940e+02		1.7119e+02	2.0372e+02	3.3092e+02		8.3181e+02	0.0000e+00
2	0.0000e+00	5.5708e-04	- -	1.4414e-02	1.0553e-02	7.5111e-03	- -	-5.9994e-03	0.0000e+00
3	0.0000e+00	1.0318e-06		-7.6569e-06	-7.9349e-06	-5.1037e-06		5.6798e-06	0.0000e+00
4	0.0000e+00	1.0046e-09		-1.0561e-09	-2.7606e-10	-6.9144e-10		-9.7254e-10	0.0000e+00

X: x position in pixel
Y: y position of th spectrum.
Line 1: a Line 3: c
Line 2: b Line 4: d

$Y = a + b.X + c.X^2 + d.X^3$

Figure 6.4: Some examples of the output produced by *isaacp sw_startrace*

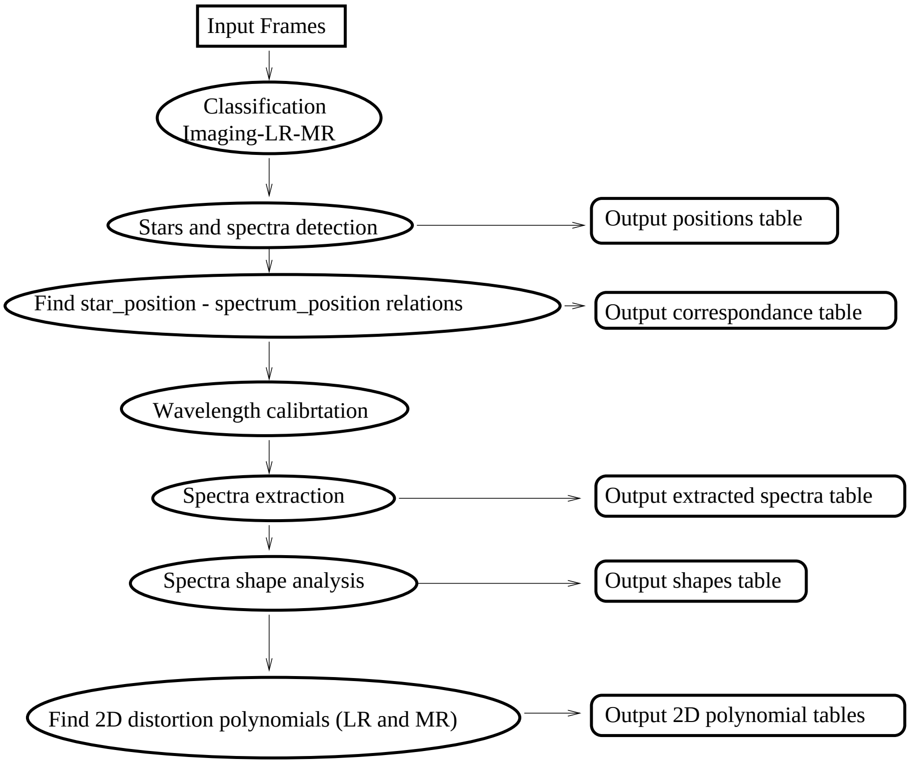

Figure 6.5: A flowchart of the steps in *isaacp sw_startrace*

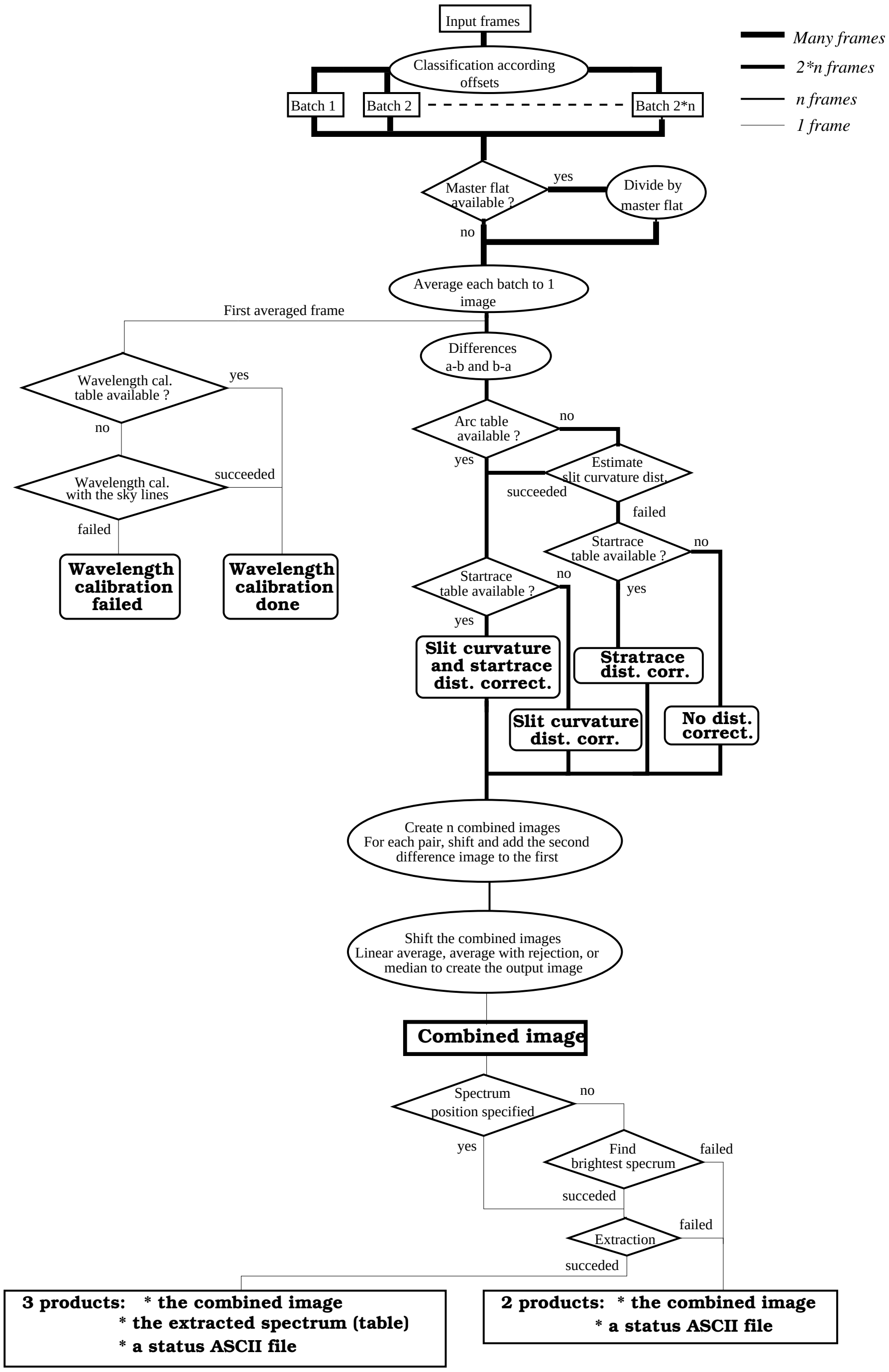

Figure 6.6: sw_spjitter algorithm

Chapter 7

An Overview of Long-wavelength Data

7.1 Introduction

The LW arm of ISAAC uses a Aladdin 1024x1024 array. When reducing data from the LW arm, you should keep in mind the following points.

- The array is simultaneously read out from 32 ports. There are 8 ports per quadrant and the readout is done along detector rows.
- There is no shutter, so the array is continuously exposed to light.
- The array is intrinsically more non-linear that the Hawaii array, and it is common to work in the regime where corrections of around 0-10% are required.

The ISAAC IRACE controller and the associated DCS software control the readout of the array. They are extremely flexible and allow many different readout schemes to be used. We offer three readout modes: UCR (Uncorrelated Read - High Bias), DCR-HB (Double Correlated Read - High Bias), DCR-LB (Double Correlated Read - Low Bias).

UCR first resets the array and then performs a single read. The minimum integration time of this mode is 0.1073 seconds. Before March 2001, the minimum integration time of this mode was 0.0733 seconds.

DCR performs two reads and the difference between these two reads is the image.

DCR-HB and DCR-LB differ in the way array voltages are set. DCR-LB has a slower readout and a smaller well depth, but far fewer hot pixels. The minimum integration time for DCR-HB is 0.2771 seconds and the minimum integration time for DCR-LB is 0.3451 seconds. Before March 2001, the minimum integration time for DCR-HB was 0.1419 seconds and DCR-LB was not offered.

The readout modes with a high bias setting are used with chopping. They have a very large number of hot pixels. The readout modes with a low bias setting are used without chopping. They have much fewer hot pixels, but they have a smaller well depth.

In addition to these modes, there are three modes that are used for a small number of calibration frames. The modes, their names, the minimum integration time, the recommended flux range and the saturation (both in ADU) are listed in the following table.

Detector Mode Name	Current MINDIT	Recommended Range	Saturation
UnCorrHighBias (UCR)	0.1073	0-28000	37000
DoubleCorrHighBias (DCR-HB)	0.2771	0-28000	37000
DoubleCorrLowBias (DCR-LB)	0.3451	0-16000	21000
UnCorrHighBiasCal	0.0733	0-28000	37000
DoubleCorrHighBiasCal	0.1419	0-28000	37000
DoubleCorrLowBiasCal	0.1419	0-16000	21000

Table 7.1: Current LW detector settings. Prior to March 2001, only the first two modes were available.

7.2 Chopping

The technique of chopping involves rapid sampling of the sky by moving the secondary in phase with the read out of the detector. In what follows, the ON beam refers to the positive image, the OFF beam refers to the negative image. In addition to chopping the telescope nods so the ON beam at position A overlaps with the OFF beam of position B. The typical nodding sequence is ABBAABBA ... etc. See the ISAAC user manual for more details.

7.3 File formats: Chopping and Non-chopping

All LW modes can be used with the chopping secondary. Only a subset of these modes, MR spectroscopy and imaging with the NB_3.21 and NB_3.28 filters, can be used without chopping.

The format of data delivered by these modes is different.

For the data that is taken without chopping, the files correspond to the average of NDIT exposures. This is similar to the format delivered by the SW detector.

For the data that is taken with chopping, one gets two sorts of frames: chopped frames and half cycle frames. Chopped frames correspond to the difference between frames taken with the secondary at the two positions (i.e. ON and OFF beams). In these frames, the sky has already been subtracted and one has both positive and negative images. Half cycle frames correspond to images taken at either of the these positions. In these frames, the sky has not been subtracted and images are always positive.

During the first year of operations, half cycle frames were produced for each chopped frame. Currently, the half cycle frames are delivered for the first chopped frame only.

Because of round off errors and because of the order in the way the half cycle and chopped frames are calculated, the chopped frame is not quite the difference between the two half cycle frames. The difference is at the level of 10e-4 and is insignificant.

7.4 General features of the ISAAC LW infrared detector

7.4.1 The zero level offset (bias)

The bias of the LW detector is much better behaved that the bias of the SW detector. The LW dark is weakly dependent on the DIT and on the amount of flux on the array.

7.4.2 Detector non-linearity

The detector non-linearity, as measured over a representative region of the array can be fitted with the function

$$f_{\mathrm{T}} = f_{\mathrm{M}} + a * f_{\mathrm{M}}^2 + b * f_{\mathrm{M}}^3 \tag{7.1}$$

where f_{M} is the the measured flux and f_{T} is the true flux.

For the readout modes which use the high bias voltage, equation 7.1 gives a relatively poor description (not better than 0.5%) of the non-linearity at low flux levels (4000 ADU). We will provide a better description of the non-linearity soon. The co-efficients for the various readout modes are given in the table .

These co-efficients have been measured over a region which is thought to be representative of the entire array. In detail, this will not be true as each pixel will behave differently.

To correct any given frame for non-linearity, one requires between one and two additional frames. For observations that do not use chopping, a dark frame

Detector Mode Name	a	b	Validity
UnCorrLowBias (UCR)	2.04e-6	0.75e-10	Before March 2001
UnCorrLowBias (UCR)	6.28e-7	1.04e-10	After March 2001
DoubleCorrHighBias (DCR-HB)	1.26e-6	1.00e-10	Before March 2001
DoubleCorrHighBias (DCR-HB)	1.05e-6	0.84e-10	After March 2001
DoubleCorrLowBias (DCR-LB)	2.78e-6	1.65e-10	-

Table 7.2: LW linearity

is required. For observations that use chopping, both the dark frame and the half cycle frames are required. These frames allow you to determine the measured flux in each pixel, which can the be corrected for non-linearity.

The procedure depends on the data set. If you have half cycle frames for every exposure, then it is simplest to correct the half cycle frames directly. If you have only the half cycle frame of the first exposure, then the correction for the non-linearity is more complex.

Correcting the half cycle frames

- Subtract the dark from each half cycle frame.
- Apply the formula listed in Eq. 7.1
- Subtract the half cycle frames from each other to create a chopped frame

Correcting the chopped frames

Correcting the chopped frame for the array non-linearity is more complicated and less accurate than correcting the half cycle frames.

- Determine the mean level of the half cycle frames in the first exposure. Call this m1
- Determine the mean level of the dark frame with the same detector set up. Call this m2
- Compute m1-m2 (an artificial background) and add it to all frames that were taken with the telescope at position A, i.e frame 1,4,5,8,9, etc.
- Apply the correction with Eq. 7.1 to these frames.
- Multiply all frames that were taken with the telescope at position B, i.e. frames 2,3,6,7,10, etc. by -1 and add m1-m2
- Apply the correction with Eq. 7.1 to these frames.

- Subtract these frames from each other to create a chopped frame

This procedure corrects the central image only. To correct the other images also, one needs to process the ON and OFF beams separately.

7.4.3 Correcting frames that were taken without chopping.

- Subtract the dark from each half cycle frame.
- Apply the formula listed in Sec. 7.4.2

7.4.4 Electronic ghosts

For bright sources, one can see electronic ghosts which are 8, 16, 24, etc. rows away from the true source. The amplitude of these ghosts depends on the brightness of the source, their position relative to the central row and how fast the array is read. The faster the array is read, the stronger these ghosts are. For very bright sources, one may see a negative ghost which is four pixels away from the source. Prior to March 2001, the array was read out more quickly than it is now, so these ghosts were significantly stronger.

7.4.5 Fringing

The Aladdin array fringes in the M band. We do not know how well the fringing can be removed with normal flats.

7.4.6 Image quality

It has taken us a long time to get acceptable image quality in the LW arm. Much of the imaging data will have a distorted PSF and most of the spectra will show strongly tilted telluric absorption lines. Although we have improved the quality of data that comes from the LW arm, we believe that there is still room for improvement.

7.4.7 Dead region

The top left hand corner of the array is completely insensitive to flux.

Chapter 8

LW imaging

8.1 Introduction

We first give an outline on how to reduce LW imaging data and then we discuss the steps in detail.

LW imaging data can be taken with two different techniques. The first uses the chopping secondary in phase with the detector readout to remove the sky, and the second uses techniques that are identical to those used in the SW arm..

8.2 Reducing Science Data

For frames taken without chopping, the basic steps are similar to those used for data taken in the SW arm:

- Dark subtraction
- Ghost removal *
- Non-linearity correction *
- Flat Fielding
- Sky subtraction
- Registration and stacking

For frames taken with chopping, the basic steps are:

- Ghost removal *
- Non-linearity correction *

- Flat Fielding
- A-B Subtraction
- Registration and stacking

Steps that are marked with an asterisk have not been thoroughly tested.

8.2.1 Dark subtraction

The dark frame and the frame from which it is subtracted should have the same DIT and should have been taken with the same readout mode. In comparison to the SW arm, the zero level offset of the array is almost independent of DIT and flux.

The dark frame should not be subtracted from chopped frames.

8.2.2 Non-linearity correction

See section 7.4.2.

8.2.3 Flat fielding

At long wavelengths, the sky is relatively insensitive to changes in the twilight sky, so one cannot use the method used for SW data. Instead, one observes a region of the sky with the telescope pointing to the Zenith, a region to the South with an airmass of 2 and a region that is even further South (an airmass of 2.4). The flat can then be created by subtracting the images that were taken at one of the higher airmasses from the image taken at Zenith. The resulting image is then normalised to 1. Five images are taken at each position. Since the position of these images is slightly different, stars can be removed before the frames are averaged.

In principle, one should apply the non-linearity correction to the flat field frames; however, in practice, correcting for the non-linearity in flat field frames will make very little difference to the final result as the pixel-to-pixel sensitivity variation in the Aladdin array is quite low, about 1%. As a comparison, the pixel-to-pixel sensitivity variation in the Hawaii array is much higher, about 7%. If a non-linearity corection is applied, a dark frame with the appropriate read out mode and DIT should be subtracted from all frames before the non-linearity correction is made.

If one divides one flat with another the pixel-to-pixel RMS is around 0.2%, so the S/N of the flats is high. However, there are other systematic effects which cause much larger differences. For flats that are taken close in time, these differences are caused by pupil ghosts (at the level of 2%), imperfections in the objective (at the level of 1-5%), and hot pixels (at the level 10-20%). For flats that are taken during different nights, the differences caused by imperfections in the objective are larger, and one starts to see differences which can only be described as smudges with a size of around 30 pixels.

Since the RMS of the flat is much lower than the typical deviation caused by hot pixels, one should replace these hot pixels with the local average.

8.2.4 Sky subtraction

For templates that do not use chopping. please refer to the description given for SW data as the steps are identical.

For templates that use chopping, the typical observing sequence is AB-BAABBA... etc. One first groups the frames into AB pairs and subtracts one image from another. This will result in an frame with one positive image and two negative images. The positive image will have a flux which corresponds to 2*DIT seconds.

8.2.5 Registration and stacking

For templates that do not use chopping, please refer to the description given for SW data as the technique is the same.

For templates that use chopping, one registers the positive images. Since there are many hot pixels, it is very helpful if there are several images to register and if these images are offset from one another.

The final step is optional and depends on how crowded the field is. One can use the two negative images that are either side of the positive image to increase the S/N ratio of the central image. The end result will be a central positive image with an exposure scaling of 4*DIT seconds, two negative images with an exposure scaling of 3*DIT seconds and two additional positive images with an exposure scaling of DIT seconds. This step should not be applied to images for which the non-linearity correction was done on the chopped frames, unless the negative beams have been treated correctly.

8.3 Ghost with the NB_4.07 filter.

This filter has a ghost around (x,y)=(530,330). It is probably caused by imperfections with the 4.07 micron NB filter, as it is not seen in other long wavelength filters. In most cases, the ghost is removed by nodding, but some residuals can be seen when paralactic angle is changing quickly. This occurs when the telescope crosses the meridian near Zenith. Flats taken with the 4.07 micron NB filter will contain this ghost, so it may be better to flat field with one of the other NB filters.

8.4 Photometric Accuracy

The information we have on the photometric accuracy of LW imaging data is limited, but we do see systematic errors at the 0.03 magnitude level. The reasons for this large error are not clear, but with additional data, we hope to get a better understanding.

8.5 Pipeline reduction

The eclipse command *lwjitter* can be used as a way to quickly reduce the data. The command uses an initialisation file, which contains parameters that define how the recipe runs. To generate one, type *lwjitter -g lwjitter.ini* at the prompt and this will create an initialisation file called lwjitter.ini. This file can be edited and modified by the user to setup some parameters.

The recipe reduces data by:

- classifying the input frames into those taken with the secondary at either side of the nod, i.e. positions A and B.
- computing (A-B)/2. Each frame shows one positive and two negative objects.
- computing the offsets between frames. Initial estimates are taken from the FITS headers and further refined through cross-correlation.
- registering the frames and stack into a single image.

Chapter 9

LW spectroscopy

9.1 Introduction

We first give an outline on how to reduce LW spectroscopic data and then we discuss the steps in detail.

LW spectroscopic data can be taken with two different techniques. The first uses the chopping secondary and the second does not.

9.2 Reducing Science Data

If chopping was not used, then the most basic way of taking IR spectra is to observe the target along two slit positions. The sky is then removed by a process which is sometimes called double sky subtraction. The basic steps of how to reduce these type of data are:

- Dark subtraction
- Ghost removal *
- Non-linearity correction *
- Flat Fielding
- First sky subtraction
- Distortion Correction and wavelength calibration
- Registration and stacking
- Extraction
- Telluric line correction

- Flux calibration

For frames taken with chopping, the basic steps are:

- Ghost removal *
- Non-linearity correction *
- Flat Fielding
- A-B Subtraction
- Slit Curvature Correction and wavelength calibration
- Registration and stacking
- Extraction
- Telluric line correction
- Flux calibration

Steps that are marked with an asterisk have not been thoroughly tested.

9.2.1 Dark subtraction

The dark frame and the frame from which it is subtracted should have the same DIT and should have been taken with the same readout mode. In comparison to the SW arm, the zero level offset of the array is almost independent of DIT and flux.

The dark frame should not be subtracted from chopped frames.

9.2.2 Non-linearity correction

See section 7.4.2.

9.2.3 Flat fielding

The technique at long wavelengths is similar to that used at short wavelengths. Please refer to the SW documentation.

9.2.4 First Sky subtraction - non chopping only

When the chopper is not used, there are two techniques one uses in taking IR spectra. There is the classical sequence where one observes the object at two slit positions. i.e. ABBAABBA... etc., and there is the more complex case where one has observed the target along several slit positions.

In the classical case, one simply subtracts frames taken at different slit positions. So one needs to form A-B, B-A, etc. This simple step removes the sky and results in an image with two spectra, one positive and one negative.

In more complex cases, one could build a sky frame from several spectra, as one does when building the sky frame in imaging. This results in an image with only one positive spectrum.

9.2.5 A-B Subtraction - chopping only

For templates that use chopping, the typical observing sequence is ABBAA... etc. One first groups the frames into AB pairs and subtracts one frame from another. This will result in an frame with one positive spectrum and two negative spectra. The positive spectrum will have a flux which corresponds to 2*DIT seconds.

9.2.6 Slit curvature correction and wavelength calibration

ISAAC spectra are strongly curved and tilted. Before the 2d spectra are shifted and combined, users should think about straightening them. It is useful to do the wavelength calibration at the same time, so that the horizontal axis is in wavelength units.

Wavelength calibration in the LW is more complicated. The arc lines are fewer and they have to compete with the bright thermal background. In the M band, there are no arc lines at all, so the wavelength calibration with arcs can only be done with the grating in third order.

For L band MR spectroscopy we provide arcs with the grating in the first and third order. Users can choose which arc they wish to use. For reasons which are not clear, the measured wavelengths of the 3rd order lines are slightly different to what one gets by multiplying the wavelengths by 3. The difference has been measured in the L band, since it is possible to see both 3rd order and 1st order lines. For the medium resolution grating the difference is described by the formula.

$$\mathrm{CorrectWavelength} = 1.005 * (\mathrm{CalibratedWavelength} - 0.005 * \mathrm{GratingSetting})$$

The formula for the low resolution grating has still to be determined.

Alternatively, for bright targets, one can use the ubiquitous telluric absorption lines. This has been tested on 1-d spectra only and has proven to be as accurate as using the arcs. For faint targets or for the purpose of correcting the 2d slit curvature, one can use the telluric emission features.

It is perfectly reasonable to use the arc lines to correct for the slit curvature and the telluric features to do the wavelength calibration of the extracted spectra.

For both arcs and telluric lines a 3rd order Legendre (4 terms) gives a good description of dispersion.

9.2.7 Combining 2d spectra - non chopping

For the classical ABBA technique, one multiplies each image by -1 and adds it back to itself after a suitable shift. This method of combining data is often called double sky subtraction as it effectively moves any residual sky that remains after the first subtraction. It results in an image that has one positive spectrum and two negative spectra either side of the positive spectrum.

In the more complex cases, one combines the individual spectra after suitable shifts have been applied.

9.2.8 Combining 2d spectra - chopping

For templates that use chopping, one registers the positive spectra. Since there are many hot pixels, it is very helpful if there are several spectra to register and if these spectra are offset from one another.

The final step is optional and depends on how crowded the field is. One can use the two negative spectra that are either side of the positive spectrum to increase the S/N ratio of the observations. The end result will be a central positive spectrum with an exposure scaling of 4*DIT seconds, two negative spectra with an exposure scaling of 3*DIT seconds and two additional positive spectra with an exposure scaling of DIT seconds. This step should not be applied to images for which the non-linearity correction was done on the chopped frames, unless the negative beams have been treated correctly.

9.2.9 Extraction

For data in which nodding or the classical ABBA technique was used, one should extract the spectrum without fitting the sky. Fitting the sky only adds noise. For more complex cases, a low order fit to the sky may be required.

9.2.10 Removing telluric lines

The technique at long wavelengths is similar to that used at short wavelengths. Please refer to the SW documentation.

Recipes to produce calibrations and to do quality control	
issacp lw_spflat	Creates spectroscopic flat fields
isaacp lw_arc	Models the slit curvature
isaacp slitpos	Measures slit positions
isaacp lw_startrace	Models the trace of a star as a function of y
isaacp lw_respfunc	Measures the spectroscopic throughput
Recipes to reduce the data	
isaacp lw_spjitter	Reduces ISAAC spectroscopic data

9.2.11 Flux calibration

The technique at long wavelengths is similar to that used at short wavelengths. Please refer to the SW documentation. The only difference is that it is not possible take slitless spectra in the LW arm.

9.3 Pipeline Recipes

This section describes the **eclipse** recipes that are used in the ISAAC pipeline in the long wavelength mode. The recipes have a variety of purposes, from recipes that take quality control data to recipes that produce reduced data. Not all recipes will be of direct interest to people wishing to reduce ISAAC data. Franes taken without chopping are not yet reduced by the pipeline.

9.3.1 Arc recipe

The recipe name is *isaacp lw_arc*. It is very similar to the SW equivalent, see Sec. 6.3.2. The slit curvature and the wavelength calibration are computed in the same way. The main difference lies in the way the frames are classified.

- The frames are classified by instrument setting.
- For each setting, there should be an even number of frames. In some cases, there will be one pair of frames (an arc with the grating at a single order) and in other cases, there will be two pairs of frames (arcs with the grating in the first and then third order).
- In each pair, there should be a dark (a frame with the lamp off) and a frame with the arc lamp on. The lamp frame is dark corrected and then reduced, and a table containing the results is written to disk.

The procedure produces two output files: a FITS table, which contains the fit to the slit curvature and the linear dispersion relation; and a PAF file, which contains the same information as the FITS file as well as additional quality control information.

With the -c option, the recipe will produce the spectrum corrected for slit curvature.

9.3.2 Slit position recipe

On a regular basis, the observatory takes images of every slit through every filter. These images are processed with the eclipse recipe *isaacp slitpos* which measures the slit position.

The input file can be either a FITS file or an ASCII file containing a list of FITS files. If the input is a FITS file, it is assumed to be a slit image. In the latter case, the first file may be a *dark*, which would then be subtracted from all subsequent images.

The output consists of two files: a FITS table which records the location of the slit in great detail, and a PAF file that contains the slit angle and slit position. Figure 6.3 shows the how the FITS table is formatted. The recipe works for both the LW and SW arms.

9.3.3 Star trace recipe

On a regular basis, the observatory takes data that can be used to determine the tilt of spectra as a function of position along the slit and the relationship between the position of an object along the slit and the corresponding location of the spectrum on the array.

The data consists of n images of a bright star stepped along the slit and spectra of the star in the MR and LR gratings at each of these positions. In total, n images and $2 * n$ spectra are obtained.

The eclipse recipe that processes this data is called *isaacp lw_startrace*. The input file consists of an ASCII file containing the data described above. The input frames can be corrected for slit curvature at the very beginning via a command line option. In this case, the user has to provide the ARC table containing the model of the slit curvature.

The recipe produces several files.

1. Positions table. The location of the stellar images and spectra are found and stored in a table named [outname]_positions.tfits.

2. Positions correspondence table. The relationship between the position of the star on the slit and the location of the corresponding spectra are fitted with a polynomial. There is one polynomial for LR and one for MR and the co-efficients of both are stored in the table called [outname]_corresp.tfits.

3. Extracted spectra. The spectra are traced, extracted, wavelength calibrated and stored in a table named [outname]_extracted.tfits. The spectra are traced with a 3rd order polynomial and the results are stored in [outname]_shapes.tfits.

4. A table with 2d polynomial modelling the slit tilt. For each grating, the spectral tilt is modelled with a 2d polynomial, which is written to a table named [outname]_poly2D_XR.tfits. This table is the one used by *spjitter* to correct the spectral tilt.

An example of what the SW version of the recipe produces is given in figure 6.4 and a flowchart of what the SW version of the recipe does is given in figure 6.5. The LW version is similar.

9.3.4 Flatfield recipe

.

The flat is the normalised difference of two images, one with the lamp on and another with the lamp off. In general, three pairs of images are taken.

The LW recipe name is *isaacp lw_spflat*, and it expects a list of frames, which are first classified by setting. For each setting, there should be an even number of frames, consisting of frames with the lamp alternatively on and off. The frames with the lamp off are subtracted from frames with the lamp on. If there is more than one pair, the resulting frames are averaged. For each setting, a fit to the vertical gradient is fitted and removed.

The recipe is very similar to that used to produce SW flats. See figure 6.1.

9.3.5 Observation recipe - chopping

The pipeline recipe that reduces LW spectroscopic data is called *isaacp lw_spjitter* and it uses an initialisation file, which contains parameters that define how the recipe runs. To generate one, type *isaacp lw_spjitter -g lw_spjitter.ini* at the prompt and this will create an initialisation file called lw_spjitter.ini. This file can be edited and modified by the user to setup some parameters.

The initialisation file contains parameters which indicate

- which table to use to correct for slit distortion
- which table to use for wavelength calibration
- which table to use to correct for spectrum tilt (the one produced by the startrace recipe) and
- which flat field to use

The recipe first classifies the frames into the three frame types: int, hcycle1 and hcycle2. It then classifies the int frames according to offsets that are recorded in the header, i.e. frames of type A and type B, and subtracts frames of type B from type A. This removes the residual background very well. Images are then flat fielded. If the flat field is missing, the recipe skips the flat fielding step.

The list of combined images are corrected for slit curvature, wavelength calibrated, shifted and recombined together to form the final combined 2d spectrum. If possible, the wavelength calibration is done with the sky lines in the half-cycle frames. Otherwise, it uses a physical model.

Then, at the end, the spectrum is detected and extracted.

9.3.6 Response function recipe

On a regular basis, the observatory observes a standard star that has a reasonably well characterised spectral energy distribution. The purpose of the *lw_respfunc* recipe is to use this data to calculate the instrument throughput. It is used for quality control purposes and observers will have little use for this routine.