EUROPEAN SOUTHERN OBSERVATORY

Organisation Européenne pour des Recherches Astronomiques dans l'Hémisphère Austral

Europäische Organisation für astronomische Forschung in der südlichen Hemisphäre

VERY LARGE TELESCOPE

ISAAC Pipeline User Manual

VLT-MAN-ESO-19500-3864

Issue 6.2.4

Date 2023-05-19

Prepared:	Lance Luvaul	July 21, 2025	
	Name	Date	Signature
Approved:	P. Ballester		
	Name	Date	Signature
Released:	M. Peron		
	Name	Date	Signature

This page was intentionally left blank

Change record

Issue/Rev.	Date	Section/Parag. affected	Reason/Initiation/Documents/Remarks
1.0	01/03/2006	All	For ISAAC 5.4.1
1.1	15/03/2006	All	For ISAAC 5.4.2
1.2	01/10/2006	All	For ISAAC 5.5.0
1.4	01/10/2007	All	For ISAAC 6.1.3
1.5	20/05/2021	A.2.2	For ISAAC 6.2.2
6.2.4	19/05/2023	A.2.2	For ISAAC 6.2.4

This page was intentionally left blank

Contents

1 Introduction

1.1 Purpose

The ISAAC pipeline is a subsystem of the *VLT Data Flow System* (DFS). It is used in two operational environments, for the ESO *Data Flow Operations* (DFO), and for the *Paranal Science Operations* (PSO), in the quick-look assessment of data, in the generation of master calibration data, in the reduction of scientific exposures, and in the data quality control. Additionally, the ISAAC pipeline recipes are made public to the user community, to allow a more personalised processing of the data from the instrument. The purpose of this document is to describe a typical ISAAC data reduction sequence with the ISAAC pipeline.

This manual is a complete description of the data reduction recipes implemented by the the ISAAC pipeline, reflecting the status of the ISAAC pipeline version 6.2.4.

1.2 Acknowledgements

Since the beginning of the ISAAC operations in 1998, many people have been involved in the ISAAC pipeline project. Nicolas Devillard developed the first version of the pipeline, and Thomas Rogon and Lars Lundin have brought substantial contributions to the project. Lars has in particular drastically improved the precision and the robustness of the wavelength calibration. From Paranal, the successive instrument scientists, Jean-Gabriel Cuby, Chris Lidman, Rachel Johnson, Andreas Jaunsen and Elena Mason have always contributed with valuables comments or new development ideas. In the operations team, Paola Amico at the beginning, and Wolfgang Hummel have been first choice testers, bringing stability and robustness to the different recipes.

1.3 Scope

This document describes the ISAAC pipeline used at ESO-Garching and ESO-Paranal for the purpose of data assessment and data quality control.

Updated versions of the present document may be found on [1]. For general information about the current instrument pipelines status we remind the user of [2]. Quality control information are at [3].

Additional information on the Common Pipeline Library (CPL) and ESOREX can be found respectively at [4], [5]. The Gasgano tool is described in [14]. A description of the instrument is in [6]. The ISAAC instrument user manual is in [7] while results of Science Verifications (SV) are at [8].

1.4 Reference documents

[1] ISAAC Pipeline Web Page
http://www.eso.org/projects/dfs/dfs-shared/web/vlt/vlt-instrument-pipelines.html

[2] Current pipeline status
http://www.eso.org/observing/dfo/quality/pipeline-status.html

[3] ESO-Data Flow Operation home page http://www.eso.org/observing/dfo/quality/

[4] CPL home page http://www.eso.org/cpl

[5] ESOREX home page http://www.eso.org/cpl/esorex.html

[6] ISAAC home page http://www.eso.org/instruments/isaac/

[7] VLT ISAAC User Manual VLT-MAN-ESO-14700-3517
http://www.eso.org/instruments/isaac/doc/

[8] ISAAC SV home page http://www.eso.org/science/ut1sv/

1.5 Applicable documents

[9] VLT Data Flow System Specifications for Pipeline and Quality Control
VLT-SPE-ESO-19600-1233

[10] DFS Pipeline & Quality Control – User Manual VLT-MAN-ESO-19500-1619

[11] ESO DICB – Data Interface Control Document GEN-SPE-ESO-19400-0794 (3.0)

[12] Common Pipeline Library User Manual VLT-MAN-ESO-19500-2720

[13] Gasgano User's Manual VLT-PRO-ESO-19000-1932

[14] ISAAC Calibration Plan VLT-PLA-ESO-14100-1384

[15] Deliverables Specification VLT-SPE-ESO-19000-1618 (2.0)

2 Overview

In collaboration with instrument consortia, the Data Flow Systems Department (DFS) of the Data Management and Operation Division is implementing data reduction pipelines for the most commonly used VLT/VLTI instrument modes. These data reduction pipelines have the following three main purposes:

Data quality control: pipelines are used to produce the quantitative information necessary to monitor instrument performance.

Master calibration product creation: pipelines are used to produce master calibration products (*e.g.*, combined bias frames, super-flats, wavelength dispersion solutions).

Science product creation: using pipeline-generated master calibration products, science products are produced for the supported instrument modes (*e.g.*, combined ISAAC jitter stacks; bias-corrected, flat-fielded FORS images, wavelength-calibrated UVES spectra). The accuracy of the science products is limited by the quality of the available master calibration products and by the algorithmic implementation of the pipelines themselves. In particular, adopted automatic reduction strategies may not be suitable or optimal for all scientific goals.

Instrument pipelines consist of a set of data processing modules that can be called from the command line, from the automatic data management tools available on Paranal or from Gasgano.

ESO offers two front-end applications for launching pipeline recipes, *Gasgano* [14] and *EsoRex*, both included in the pipeline distribution (see Appendix A, page 73). These applications can also be downloaded separately from `http://www.eso.org/gasgano` and `http://www.eso.org/cpl/esorex.html`. An illustrated introduction to Gasgano is provided in the "Quick Start" Section of this manual (see page 16).

The ISAAC instrument and the different types of ISAAC raw frames and auxiliary data are described in Sections 3, 6, and 7.

A brief introduction to the usage of the available reduction recipes using Gasgano or EsoRex is presented in Section 4.

In section 5 we advice the user about known data reduction problems.

An overview of the data reduction, what are the input data, and the recipes involved in the calibration cascade is provided in Section 8.

More details on what are inputs, products, quality control measured quantities, and controlling parameters of each recipe is given in Section 9.

More detailed descriptions of the data reduction algorithms used by the individual pipeline recipes can be found in Section 10.

In Appendix A the installation of the ISAAC pipeline recipes is described and in Appendix B a list of used abbreviations and acronyms is given.

3 ISAAC Instrument Description

ISAAC is an infrared (1 to 5 micrometers) imager and spectrograph that lies at the Nasmyth B focus of UT1 and is in operation since 1998. It has two arms, one equipped with a 1024 x 1024 Hawaii Rockwell array, and the other with a 1024 x 1024 Aladdin array from Santa Barbara Research Center. The Hawaii arm is used at short wavelengths (SW mode) between 1 and 2.5 micrometers. Prior to October 2002, the Aladdin arm was used exclusively at long wavelengths (LW mode) above 2.5 micrometers. From October 2002 onwards this arm is also offered for imaging in bands J, H and K.

Figure 3.0.1 shows a picture of the instrument mounted on the first unit telescope on the VLT.

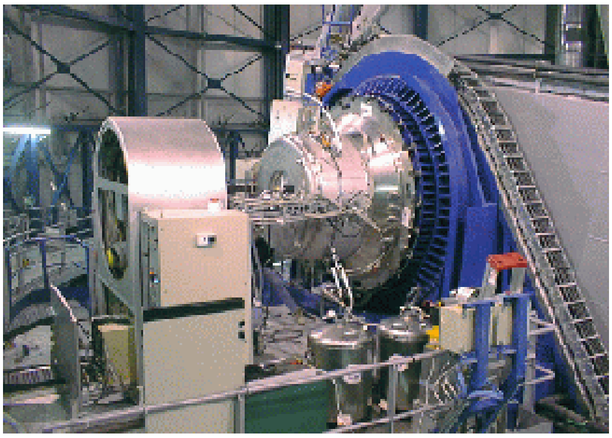

Figure 3.0.1: Picture of ISAAC

ISAAC is capable of:

- 1-2.5 microns imaging over a 2.5 arcmin x 2.5 arcmin field.
- 1-2.5 microns imaging polarimetry over a 2.5 arcmin x 2.5 arcmin field.
- 1-2.5 microns long slit low (Rs 500) and medium resolution (Rs 3000) spectroscopy
- 2.5-5 microns imaging over a 1.25 arcmin x 1.25 arcmin field.
- 2.5-5 microns long slit low (Rs 500) and medium resolution (Rs 3000) spectroscopy

Figure 3.0.2 shows the optical layout of ISAAC. At the top and bottom are two cameras which are optimized for the 1-2.5 microns and 2-5 microns ranges and used to either re-image the telescope focal plane or the intermediate spectrum produced by the grating spectrometer. The short wave camera is equipped with a Rockwell Hawaii 1024x1024 pixel Hg:Cd:Te array and the long wave camera with an SBRC Aladdin 1024x1024 pixel InSb array. The characteristics of the detectors are given in the table 3.0.0.

Given the extreme range of backgrounds, from 0.1 electrons/sec/pixel to 3,000,000 electrons/second/pixel, the LW and SW detectors are used with different readout modes.

See [7] for a more complete description of ISAAC.

Detector	Format	Pixel size	Gain	Read Noise	Q.E
SW - Rockwell	1024 x 1024	18.5 microns	4.5 electrons/ADU	11 electrons	0.65
LW - Aladdin, SBRC	1024 x 1024	27 microns	7.8 electrons/ADU (DCR-HB, UCR) 8.7 electrons/ADU (DCR-LB)	40 electrons	0.80

Table 3.0.0: ISAAC detector array characteristics

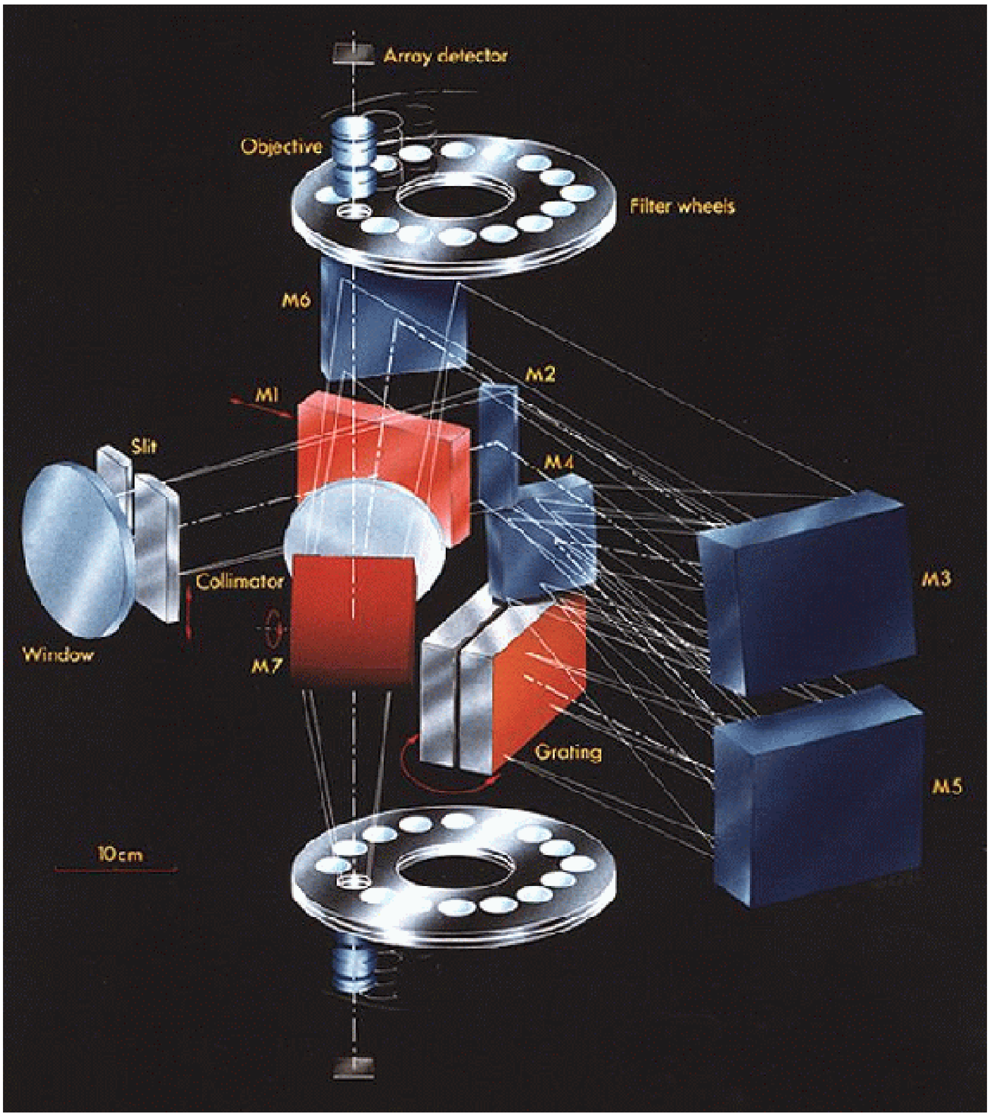

Figure 3.0.2: ISAAC optical layout

4 Quick start

This section describes the most immediate usage of the ISAAC pipeline recipes.

4.1 ISAAC pipeline recipes

The current ISAAC pipeline is based on a set of 11 stand-alone recipes involved in the data reduction cascade.

The 7 recipes in imaging mode are:

isaac_img_dark: Creates a master dark frame to calibrate the dark current.

isaac_img_twflat: Uses a twilight set of images to create a master flat field frame and a bad pixels map.

isaac_img_slitpos: Makes a precise slit analysis to help the slit positioning on the instrument.

isaac_img_detlin: Measures the non-linearity of the detector by using the calibration unit.

isaac_img_illum: Uses a scan across the field of a moderately bright star to measure large scale variations of the detector sensitivity.

isaac_img_zpoint: Zero point measurements using standard star observations.

isaac_img_jitter: Main reconstruction routine, including dark correction, flatfield calibration, bad pixels cleaning, and images correlation and recombination.

The 4 recipes in spectroscopic mode are:

isaac_spc_flat: Creates a master flat field.

isaac_spc_arc: Uses Xenon or Argon lamps to calibrate both the wavelength and the slit curvature distortion.

isaac_spc_startrace: Uses series of bright spectra taken along the slit to calibrate the stratrace distortion.

isaac_spc_jitter: Main observation recipe, that corrects the flat field, the distortion, reconstructs the combined image, detect, extract and calibrate in wavelength the brightest spectrum.

4.2 An introduction to Gasgano and EsoRex

Before being able to call pipeline recipes on a set of data, the data must be opportunely classified, and associated with the appropriate calibrations. The *Data Classification* consists of tasks such as: "What kind of data am I?", *e.g.*, BIAS, "to which group do I belong?", *e.g.*, to a particular Observation Block or template. *Data Association* is the process of selecting appropriate calibration data for the reduction of a set of raw science frames. Typically, a set of frames can be associated if they share a number of properties, such as instrument and detector configuration. As all the required information is stored in the FITS headers, data association is based on a set of keywords (called "association keywords") and is specific to each type of calibration.

The process of data classification and association is known as data organisation. The *DO Category* is the label assigned to a data type as a result of data classification.

An instrument pipeline consists of a set of data processing modules that can be called from different host applications, either from the command line with *Esorex*, from the automatic data management tools available at Paranal, or from the graphical *Gasgano* tool.

Gasgano is a data management tool that simplifies the data organisation process, offering automatic data classification and making the data association easier (*even if automatic association of frames is not yet provided*). *Gasgano* determines the classification of a file by applying an instrument specific rule, while users must provide this information to the recipes when they are executed manually using *Esorex* from the command line. In addition, *Gasgano* allows the user to execute directly the pipeline recipes on a set of selected files.

4.2.1 Using Gasgano

To get familiar with the ISAAC pipeline recipes and their usage, it is advisable to begin with *Gasgano*, because it provides a complete graphic interface for data browsing, classification and association, and offers several other utilities such as easy access to recipes documentation and preferred data display tools.

Gasgano can be started from the system prompt in the following way:

```
gasgano &
```

The *Gasgano* main window will appear. On Figure 4.2.1, a view on a set of ISAAC data is shown as an example. *Gasgano* can be pointed to the directories where the data to be handled are located using the navigation panels accessible via the *Add/Remove Files* entry of the *File* menu (shown on the upper left of the figure).

More information about a single frame can be obtained by clicking on its name: the corresponding FITS file header will be displayed on the bottom panel, where specific keywords can be opportunely filtered and searched. Images and tables may be easily displayed using the viewers specified in the appropriate *Preferences* fields.

Frames can be selected from the main window for being processed by the appropriate recipe. This will open a *Gasgano* recipe execution window (see Figure 4.2.2), having all the specified files listed in its *Input Frames* panel.

Help about the recipe may be obtained from the *Help* menu. Before launching the recipe, its configuration may be opportunely modified on the *Parameters* panel (on top). The window contents might be saved for later use by selecting the *Save Current Settings* entry from the *File* menu, as shown in figure.

At this point the recipe can be launched by pressing the *Execute* button. Messages from the running recipe will appear on the *Log Messages* panel at bottom, and in case of successful completion the products will be listed on the *Output Frames* panel, where they can be easily viewed and located back on the Gasgano main window.

Please refer to the *Gasgano User's Manual* [14] for a more complete description of the *Gasgano* interface.

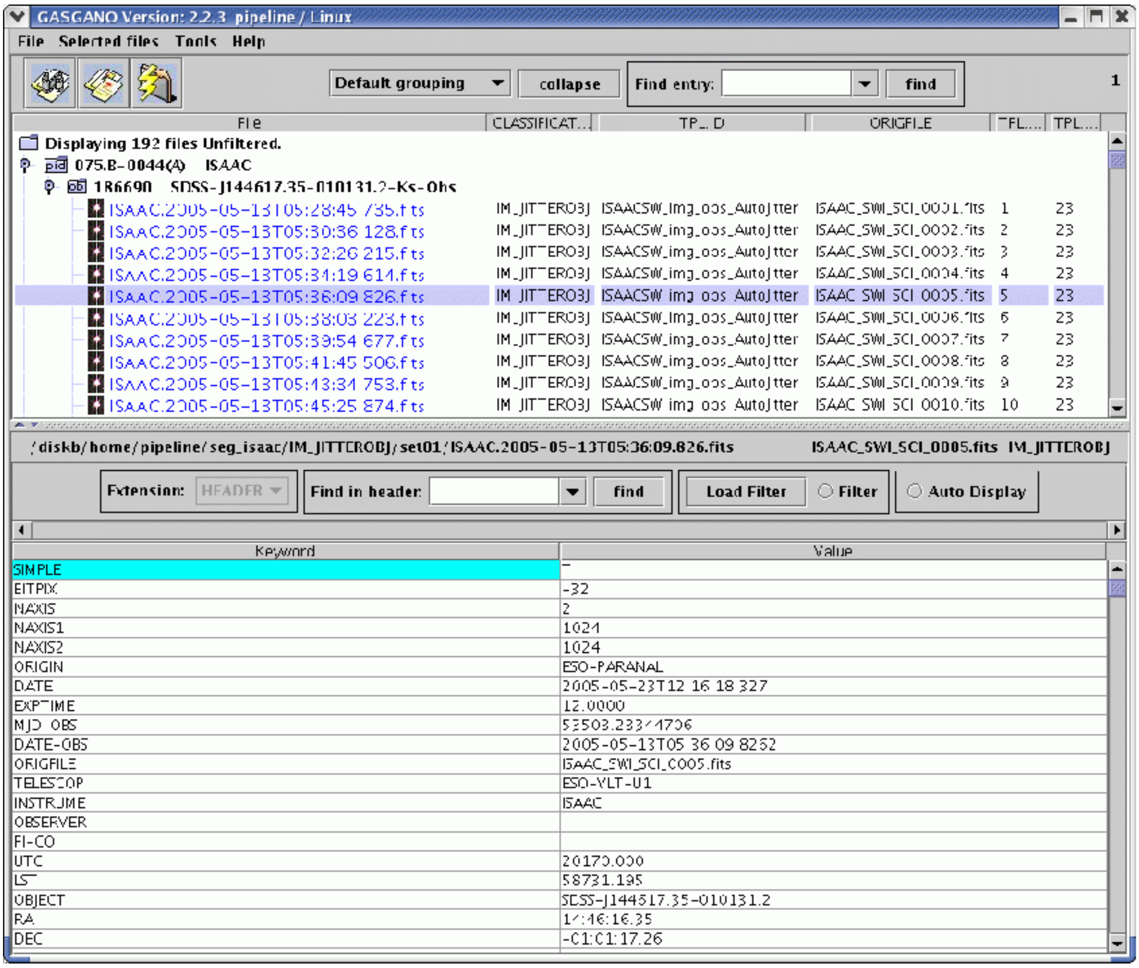

Figure 4.2.1: *The Gasgano main window*

4.2.2 Using EsoRex

EsoRex is a command line utility for running pipeline recipes. It may be embedded by users into data reduction scripts for the automation of processing tasks. On the other side, *EsoRex* doesn't offer all the facilities available with *Gasgano*, and the user must classify and associate the data using the information contained in the FITS header keywords (see Section 6, page 24). The user should also take care of defining the input set-of-frames and the appropriate configuration parameters for each recipe run:

The set-of-frames: Each pipeline recipe is run on a set of input FITS data files. When using *EsoRex* the file names must be listed together with their DO category in an ASCII file, the *set-of-frames* (SOF), that is required when launching a recipe. [1]

Here is an example of SOF, valid for the *isaac_img_jitter* recipe:

[1]The set-of-frames corresponds to the *Input Frames* panel of the *Gasgano* recipe execution window (see Figure 4.2.2, page 21).

```
/file_path/ISAAC.2004-08-14T10:20:56.497.fits   IM_JITTEROBJ
/file_path/ISAAC.2004-08-14T10:22:44.285.fits   IM_JITTEROBJ
/file_path/flat.fits                            MASTER_IMG_FLAT
/file_path/bpm.fits                             MASTER_BPM
```

It contains for each input frame the full path file name and its DO category. The pipeline recipe will access the listed files when required by the reduction algorithm.

Note that the ISAAC pipeline recipes do not verify in any way the correctness of the *DO Category* specified by the user in the SOF. The reason of this lack of control is that the ISAAC recipes are just the DRS component of the complete pipeline running on Paranal, where the task of data classification and association is carried out by separate applications. Moreover, using *Gasgano* as an interface to the pipeline recipes will always ensure a correct classification of all the data frames, assigning the appropriate DO category to each one of them (see Section 4.2.1, page 17).

A recipe handling an incorrect SOF may stop or display unclear error messages at best. In the worst cases, the recipe would apparently run without any problem, producing results that may look reasonable, but are actually flawed.

EsoRex syntax: The basic syntax to use ESOREX is the following:

esorex [esorex_options] recipe_name [recipe_options] set_of_frames

To get more information on how to customise ESOREX (see also [6]) run the command:

esorex - -help

To generate a configuration file esorex.rc in the directory $HOME/.esorex run the command:

esorex - -create-config

A list of all available recipes, each with a one-line description, can be obtained using the command:

esorex - -recipes

All recipe parameters (aliases) and their default values can be displayed by the command

esorex - -params recipe_name

To get a brief description of each parameter meaning execute the command:

esorex - -help recipe_name

To get more details about the given recipe give the command at the shell prompt:

esorex - -man-page recipe_name

Recipe configuration: Each pipeline recipe may be assigned an *EsoRex* configuration file, containing the default values of the parameters related to that recipe.[2] The configuration files are normally generated in the directory `$HOME/.esorex`, and have the same name as the recipe to which they are related, with the file name extension `.rc`. For instance, the recipe *isaac_img_jitter* has its *EsoRex* generated configuration file named `isaac_img_jitter.rc`, and is generated with the command:

esorex - -create-config isaac_img_jitter

The definition of one parameter of a recipe may look like this:

[2] The *EsoRex* recipe configuration file corresponds to the *Parameters* panel of the *Gasgano* recipe execution window (see Figure 4.2.2, page 21).

```
# --xcorr
# Cross correlation search and measure sizes.
isaac.isaac_img_jitter.xcorr=40,40,65,65
```

In this example, the parameter `isaac.isaac_img_jitter.xcorr` is set to the value `40,40,65,65`. In the configuration file generated by *EsoRex*, one or more comment lines are added containing information about the possible values of the parameter, and an alias that could be used as a command line option.

The recipes provided by the ISAAC pipeline are designed to implement a cascade of macro data reduction steps, each controlled by its own parameters. For this reason and to prevent parameter name clashes we specify as parameter prefix not only the instrument name but also the name of the step they refer to. Shorter parameter aliases are made available for use on the command line.

The command

esorex - -create-config recipe_name

generates a default configuration file **recipe_name.rc** in the directory **$HOME/.esorex**[3].

A recipe configuration file different from the default one can be specified on the command line:

esorex - -recipe-config=my_alternative_recipe_config

Recipe parameters are provided in Section 9 and their role is described in Section 10.

More than one configuration file may be maintained for the same recipe but, in order to be used, a configuration file not located under `$HOME/.esorex`, or having a name different from the recipe name, should be explicitly specified when launching a recipe.

Recipe execution: A recipe can be run by specifying its name to *EsoRex*, together with the name of a set-of-frames. For instance, the following command line would be used to run the recipe *isaac_img_jitter* for processing the files specified in the set-of-frames `isaac_img_jitter.sof`:

esorex isaac_img_jitter isaac_img_jitter.sof

The recipe parameters can be modified either by editing directly the used configuration file, or by specifying new parameter values on the command line using the command line options defined for this purpose. Such command line options should be inserted after the recipe name and before the SOF name, and they will supersede the system defaults and/or the configuration file settings. For instance, to set the *isaac_img_jitter* recipe *xcorr* parameter to `20,20,65,65`, the following should be typed:

esorex isaac_img_jitter - -xcorr="20,20,65,65" isaac_img_jitter.sof

For more information on *EsoRex*, see [6].

[3] If a number of recipe parameters are specified on the command line, the given values will be used in the created configuration file.

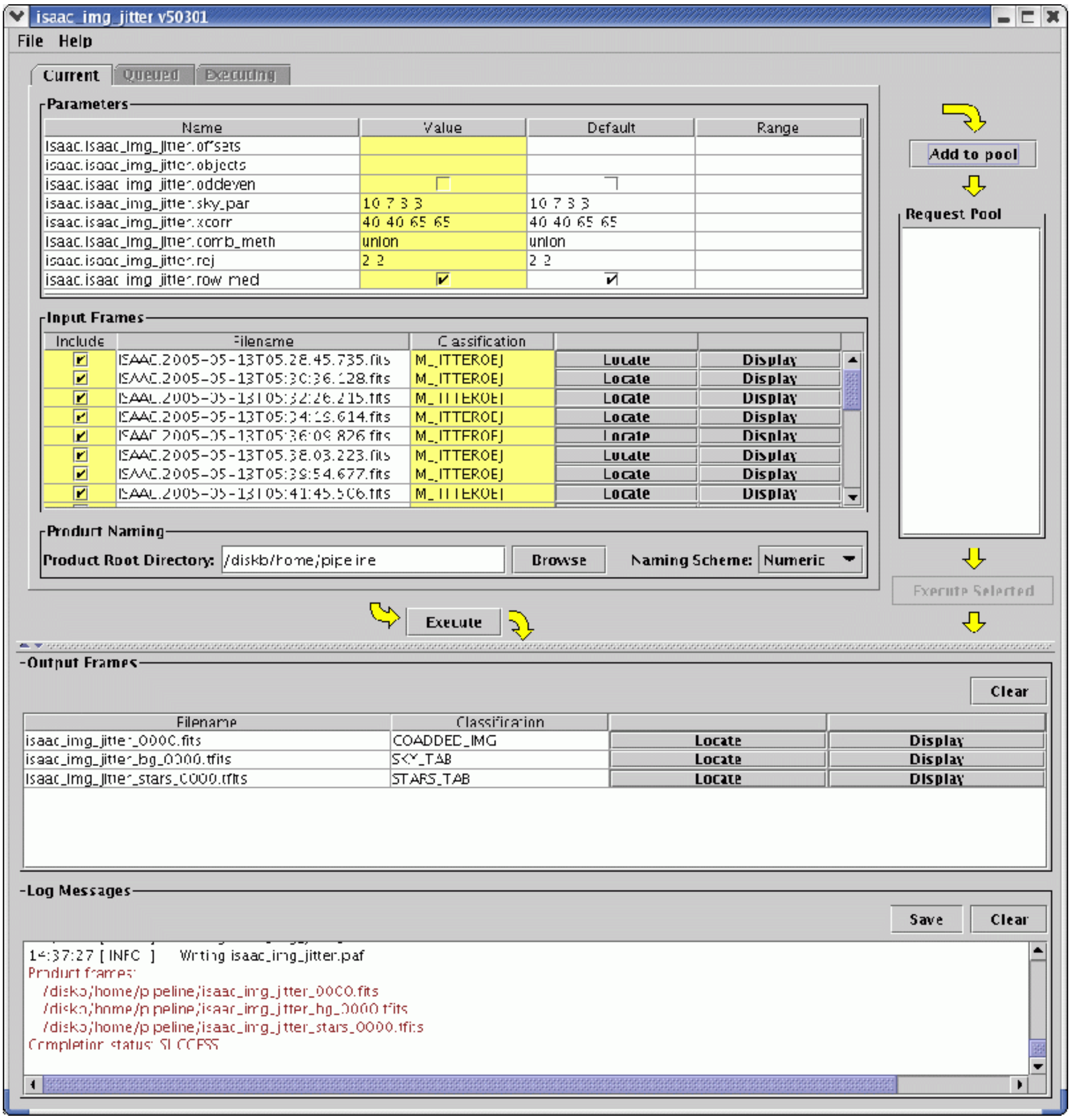

Figure 4.2.2: *The Gasgano recipe execution window*

5 Known Problems

There are some features on the ISAAC data that are challenging for the data reduction pipeline, or simply that cannot be corrected by an automatic tool. For some of these, some additional interactive data analysis may be necessary to remove them.

Some of them are described in the following sections.

5.1 Wavelength calibration

In order to perform the wavelength calibration, the observed lamp lines (Xenon or Argon) or sky lines are matched with the theoretical catalog lines. The zone of the catalog to be used is defined with a rough estimation of the wavelength range obtained (among others) from the central wavelength given in the header of the FITS file. If the solution computed gives a central wavelength that is more than 50 pixels away from the input central wavelength, an error is raised, considering that the match must be wrong. The consequence of having this safety criteria is that the wavelength calibration is systematically failing on data where the central wavelength is too far away from its real value, which may happen on some data (around 10 cases recorded during period 76).

5.2 Arc recipe

One setting (SW Medium resolution at 1.71 microns) cannot be handled by the *isaac_spc_arc* recipe, since there are only two arc lines detected to be used for the image distortion part. The recipe stops without reaching the wavelength calibration part. Science products with this setting are reduced without arc product (without optical distortion correction in x-direction) and the wavelength calibration is done using the sky lines imprinted in the science raw frames.

5.3 Misalignment in the dispersion direction

In August 2002 the collimator got stuck and was moved to a compromise position. As a consequence, ISAAC image quality is limited by the current fixed collimator position optimized for the broad band filters. The point spread function of SW-arm spectra of point sources can show two peaks, meaning that spectra are not longer fully focused.

Ongoing problems concerning the alignment of the LW-arm and SW-arm objective wheels had an impact on the reduction scheme of LW chopping spectra. Instead of the day-time calibration arc frames, the sky lines are used for wavelength calibration. In case the sky lines cannot be used, the physical model is used for wavelength calibration. Note that LW chopping spectra are not subject of optical distortion correction using startrace frames and arc line curvature.

Tilted telluric absorption lines can occur in spectra of point sources. They can appear for both gratings LR and MR and both arms LW and SW. The tilt is about 0.187 pixels per row, while sky emission lines are not tilted, apart from the optical distortion which is several orders smaller. Since P70 there are more and more examples where the tilt can be curved, meaning that the dispersion is no longer a constant for all of the rows. The effect is not fully understood and the pipeline does not correct this feature.

The tilted telluric features occurred again in August 2003, meaning that absorption lines and telluric features imprinted in a spectrum appear tilted with respect to the dispersion direction. This wavelength shift across the PSF of the spectrum cannot be corrected by the pipeline. The collimator is no longer at a position for minimum tilt of the telluric features in the center of the spectrum.

5.4 50 Hz noise

The 50 Hz noise effect comes and disappears in a sporadic manner in ISAAC data. It appears as horizontal stripes in the images, and can be detected with the monitoring of the RON values.

6 ISAAC Data Description

This section gives a description of the raw data produced by ISAAC.

ISAAC data do not contain extensions, the images are always contained in the primary FITS data unit. In nodding mode, it consists of one image, in chopping mode, the two Half-Cycle images are stored as a cube (NAXIS3=2).

Any raw frame can be classified on the basis of a set of keywords read from its header. Data classification is typically carried out by the DO or by *Gasgano* [14], that apply the same set of classification rules. The association of a raw frame with calibration data (*e.g.*, of a science frame with a master bias frame) can be obtained by matching the values of a different set of header keywords.

Each kind of raw frame is typically associated to a single ISAAC pipeline recipe, *i.e.*, the recipe assigned to the reduction of that specific frame type. In the pipeline environment this recipe would be launched automatically.

In the following, all ISAAC raw data frames are listed, together with the keywords used for their classification and correct association. The indicated *DO category* is a label assigned to any data type after it has been classified, which is then used to identify the frames listed in the *Set of Frames* (see Section 4.2.2, page 18).

Raw frames can be distinguished between *imaging* frames and *spectrocopy* frames. Their intended use is implicitly defined by the assigned recipe.

6.1 Imaging frames

- **Lamp image for non-linearity calibration:**
  DO category: `LINEAR_LAMP`
  Processed by: `isaac_img_detlin`

Classification keywords:	Association keywords:
`DPR CATG = CALIB`	
`DPR TYPE = LAMP`	
`DPR TYPE = LINEARITY`	
`DPR TECH = IMAGE`	

- **Dark image for non-linearity calibration:**
  DO category: `LINEAR_DARK`
  Processed by: `isaac_img_detlin`

Classification keywords:	Association keywords:
`DPR CATG = CALIB`	
`DPR TYPE = OTHER`	
`DPR TYPE = LINEARITY`	
`DPR TECH = IMAGE`	

- **Image for slit position calibration:**
  DO category: `SLIT_IMG`
  Processed by: `isaac_img_slitpos`

Classification keywords: Association keywords:
`DPR CATG = TECHNICAL`
`DPR TYPE = SLIT`
`DPR TECH = IMAGE`

- **Image for detector illumination calibration:**
  DO category: `IM_ILLUM`
  Processed by: `isaac_img_illum`

  Classification keywords: Association keywords:
  `DPR CATG = CALIB`
  `DPR TYPE = FLUX`
  `DPR TECH = IMAGE`

- **Zero point calibration image:**
  DO category: `IM_ZPOINT`
  Processed by: `isaac_img_zpoint`

  Classification keywords: Association keywords:
  `DPR CATG = CALIB`
  `DPR TYPE = STD`
  `DPR TECH = IMAGE`
  `DPR TECH = JITTER`

- **Polarimetry flat field calibration:**
  DO category: `POL_FLAT`
  Processed by: `isaac_img_twflat`

  Classification keywords: Association keywords:
  `DPR CATG = CALIB`
  `DPR TYPE = FLAT`
  `DPR TYPE = SKY`
  `DPR TECH = POLARIMETRY`

- **Imaging flat field calibration::**
  DO category: `IM_FLAT`
  Processed by: `isaac_img_twflat`

  Classification keywords: Association keywords:
  `DPR CATG = CALIB`
  `DPR TYPE = FLAT`
  `DPR TYPE = SKY`
  `DPR TECH = IMAGE`

- **Imaging dark calibration:**
  DO category: `IM_DARK`
  Processed by: `isaac_img_dark`

Classification keywords:
DPR CATG = CALIB
DPR TYPE = DARK
DPR TECH = IMAGE

Association keywords:

- **Imaging acquisition for the startrace calibration:**

  DO category: IM_STARTRACE
  Processed by: isaac_spc_startrace

  Classification keywords:
  DPR CATG = CALIB
  DPR TYPE = STARTRACE
  DPR TECH = IMAGE

  Association keywords:

- **Jitter object observation:**

  DO category: IM_JITTEROBJ
  Processed by: isaac_imag_jitter

  Classification keywords:
  DPR CATG = SCIENCE
  DPR TYPE = OBJECT
  DPR TECH = IMAGE
  DPR TECH = JITTER

  Association keywords:

- **Jitter sky observation:**

  DO category: IM_JITTERSKY
  Processed by: isaac_img_jitter

  Classification keywords:
  DPR CATG = SCIENCE
  DPR TYPE = SKY
  DPR TECH = IMAGE
  DPR TECH = JITTER

  Association keywords:

- **Chopping observation:**

  DO category: IM_CHOPPING
  Processed by: isaac_img_jitter

  Classification keywords:
  DPR CATG = SCIENCE
  DPR TYPE = OBJECT
  DPR TECH = IMAGE
  DPR TECH = CHOPPING

  Association keywords:

- **Chopping calibration:**

  DO category: IM_CHOPPING_CAL
  Processed by: isaac_img_zpoint

Classification keywords:
DPR CATG = CALIB
DPR TYPE = STD
DPR TECH = IMAGE
DPR TECH = CHOPPING

Association keywords:

6.2 Spectroscopic frames

- **Spectroscopic lamp for arcs calibration:**
  DO category: SP_ARC
  Processed by: isaac_spc_arc

  Classification keywords:
  DPR CATG = CALIB
  DPR TYPE = WAVE
  DPR TYPE = LAMP
  DPR TECH = SPECTRUM

  Association keywords:

- **Spectroscopic flat field:**
  DO category: SP_FLAT
  Processed by: isaac_spc_flat

  Classification keywords:
  DPR CATG = CALIB
  DPR TYPE = FLAT
  DPR TYPE = LAMP
  DPR TECH = SPECTRUM

  Association keywords:

- **Spectroscopic startrace calibration:**
  DO category: SP_STARTRACE
  Processed by: isaac_spc_startrace

  Classification keywords:
  DPR CATG = CALIB
  DPR TYPE = STARTRACE
  DPR TECH = SPECTRUM

  Association keywords:

- **Spectroscopic standard star in chopping:**
  DO category: SP_CHOPPING_CAL
  Processed by: isaac_spc_jitter

  Classification keywords:
  DPR CATG = CALIB
  DPR TYPE = STD
  DPR TECH = SPECTRUM
  DPR TECH = CHOPPING

  Association keywords:

- **Spectroscopic standard star in nodding:**
  DO category: `SP_NODDING_CAL`
  Processed by: `isaac_spc_jitter`

Classification keywords:	Association keywords:
`DPR CATG = CALIB`	
`DPR TYPE = STD`	
`DPR TECH = SPECTRUM`	
`DPR TECH = NODDING`	

- **Spectroscopic object in nodding:**
  DO category: `SP_NODDING_CALOBJ`
  Processed by: `isaac_spc_jitter`

Classification keywords:	Association keywords:
`DPR CATG = CALIB`	
`DPR TYPE = OBJECT`	
`DPR TECH = SPECTRUM`	
`DPR TECH = NODDING`	

- **Spectroscopic sky in nodding:**
  DO category: `SP_NODDING_CALSKY`
  Processed by: `isaac_spc_jitter`

Classification keywords:	Association keywords:
`DPR CATG = CALIB`	
`DPR TYPE = SKY`	
`DPR TECH = SPECTRUM`	
`DPR TECH = NODDING`	

- **Spectroscopic observation in chopping:**
  DO category: `SP_CHOPPING`
  Processed by: `isaac_spc_jitter`

Classification keywords:	Association keywords:
`DPR CATG = SCIENCE`	
`DPR TYPE = OBJECT`	
`DPR TECH = SPECTRUM`	
`DPR TECH = CHOPPING`	

- **Spectroscopic object observation in nodding:**
  DO category: `SP_NODDINGOBJ`
  Processed by: `isaac_spc_jitter`

Classification keywords:	Association keywords:
`DPR CATG = SCIENCE`	
`DPR TYPE = OBJECT`	
`DPR TECH = SPECTRUM`	
`DPR TECH = NODDING`	

- **Spectroscopic sky observation in nodding:**
  DO category: `SP_NODDINGSKY`
  Processed by: `isaac_spc_jitter`

  Classification keywords:
  `DPR CATG = SCIENCE`
  `DPR TYPE = SKY`
  `DPR TECH = SPECTRUM`
  `DPR TECH = NODDING`

  Association keywords:

7 Static Calibration Data

For ISAAC, the static calibration data are the standard stars catalogs, the OH, Xenon and Argon spectroscopic lines and the spectral energy distribution of the standard stars.

All these catalogs can be found in the ISAAC pipeline source distribution as ASCII files in isaacp/catalogs:

```
ls isaacp/catalogs/*

isaacp/catalogs/lines:
ne.txt  oh.txt  xe.txt

isaacp/catalogs/seds:
A0V.txt  B0V.txt  B3V.txt  B8V.txt  B9V.txt

isaacp/catalogs/stdstars:
Arnica.txt           LCO-Palomar-NICMOS-Red-Stars.txt  UKIRT-Extended.txt
Conica.txt           LCO-Palomar.txt                   UKIRT-Fundamental.txt
                     MSSSO-Photometric.txt             UKIRT-LM.txt
ESO-VanDerBliek.txt  MSSSO-Spectroscopic.txt           UKIRT-Standards.txt
Isaac.txt            SAAO-Carter.txt
```

From these ASCII catalogs, it is possible to generate the FITS calibration tables needed by the recipes by using the following utilities:

- isaac_util_seds : SEDS table creation
- isaac_util_stdstars : Standard stars catalog creation
- isaac_util_genlines : Generate spectrum calibration FITS tables

For example, the following:

```
$ more IN
/home/yjung/isaacp/catalogs/stdstars/Arnica.txt STDSTAR_CAT
/home/yjung/isaacp/catalogs/stdstars/Conica.txt STDSTAR_CAT
/home/yjung/isaacp/catalogs/stdstars/ESO-VanDerBliek.txt STDSTAR_CAT
/home/yjung/isaacp/catalogs/stdstars/Isaac.txt STDSTAR_CAT
/home/yjung/isaacp/catalogs/stdstars/LCO-Palomar-NICMOS-Red-Stars.txt STDSTAR_CAT
/home/yjung/isaacp/catalogs/stdstars/LCO-Palomar.txt STDSTAR_CAT
/home/yjung/isaacp/catalogs/stdstars/MSSSO-Photometric.txt STDSTAR_CAT
/home/yjung/isaacp/catalogs/stdstars/MSSSO-Spectroscopic.txt STDSTAR_CAT
/home/yjung/isaacp/catalogs/stdstars/SAAO-Carter.txt STDSTAR_CAT
/home/yjung/isaacp/catalogs/stdstars/UKIRT-Extended.txt STDSTAR_CAT
/home/yjung/isaacp/catalogs/stdstars/UKIRT-Fundamental.txt STDSTAR_CAT
```

```
/home/yjung/isaacp/catalogs/stdstars/UKIRT-LM.txt STDSTAR_CAT
/home/yjung/isaacp/catalogs/stdstars/UKIRT-Standards.txt STDSTAR_CAT

$ esorex isaac_util_stdstars IN
```

will create the static calibration FITS file needed by the recipes.

7.1 Standard Star Catalogs

The standard star catalogs are used by the `isaac_img_zpoint` recipe to get the magnitude of the observed standard star to compute the Zero Point, and by `isaac_spc_jitter` to get the observed standard star type in order to compute the response function of the telescope (only for standard stars observations).

Stars are currently taken from the following catalogs:

- Arnica (41 entries)
- ESO Van der Bliek (264 entries)
- Isaac specific (9238 entries)
- LCO Palomar (64 entries)
- LCO Palomar NICMOS red stars (26 entries)
- MSSSO Photometric (54 entries)
- MSSSO Spectroscopic (343 entries)
- SAAO Carter (67 entries)
- UKIRT extended (54 entries)
- UKIRT fundamental (33 entries)
- UKIRT LM (36 entries)
- UKIRT standards (89 entries)

Each entry in the catalog correspond to one standard star. They all contain the following informations:

- The name of the star
- The position (RA / DEC) of the star
- The spectral type of the star
- The different magnitudes in band J, H, K, Ks, L, M, Lprime, Mprime

7.2 Spectral lines

The ISAAC pipeline also internally contains the emission lines of the atmosphere, of the Argon and of the Xenon gas. These are used for the wavelength calibration to generate theoretical signals used to correlate with the observed ones (OH lines for real observations, Xenon and Argon for calibration lamps).

7.3 Spectral energy distribution

In order to compute the response function, a standard star of a given type is observed. The SED of this type is then used for the computation. The supported spectral types are A0V, B0V, B3V, B8V, B9V.

8 Data Reduction

In this section, after an overview of the main problems the data reduction needs to solve, we list the required data and the recipes which allow to solve them, giving the data reduction sequence necessary to reduce calibration and science data.

8.1 Data reduction overview

In SW imaging mode, the used observation technique is the jittering. Small shifts are applied between successive frames. This way, with a set of a sufficient number of frames, it is possible to make a precise estimation of the sky for all the pixels of the detector, the sky estimation being the most important and difficult part usually in IR.

In SW spectroscopy, the observation technique is the nodding. The spectrum is moved along the slit and acquired in two main positions. With successive differences, we can get rid of the sky.

In LW mode, the level of the background is too high, and it becomes necessary to apply the chopping observation technique to get rid of the sky. The chopping consists in moving the M2 miror at a high frequency to take two simultaneous images of two parts of the sky. Those two images are referred to as the Half-Cycle frames. They can then be subtracted and the sky disappears from this subtraction.

In LW imaging mode, the chopping is used together with the jitter technique. In LW spectroscopy, it is used together with the nodding.

8.1.1 Bias variations (SW)

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 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).

8.1.2 Shift register glow (SW)

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. 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.

8.1.3 Electrical ghosts (SW)

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.

8.1.4 Odd-even column effect (SW)

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 0.05% 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 than 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.

8.1.5 50Hz pickup (SW)

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. 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.

In most cases, the 50 Hz signal will not be a problem, but if you find that it is, here is a simple method to remove it:

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.

8.1.6 Fringing (SW)

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.

8.1.7 The zero level offset (bias) (LW)

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

8.1.8 Detector non-linearity (SW/LW)

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

$$f_{\rm T} = f_{\rm M} + a * f_{\rm M}^2 + b * f_{\rm M}^3 \tag{1}$$

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

For the readout modes which use the high bias voltage, equation 1 gives a relatively poor description (not better than 0.5%) of the non-linearity at low flux levels (4000 ADU).

8.1.9 Electronic ghosts (LW)

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.

8.1.10 Image quality (LW)

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.

8.2 Required input data

To be able to reduce science data one needs to use raw, product data and pipeline recipes in a given sequence which provides all the input necessary to each pipeline recipe. We call this sequence a data reduction cascade.

Calibration data products can be generated from raw data using the pipeline recipes. Alternatively the user may use calibration products obtained from the ESO archive or from the ESO Data Flow Operation department.

8.3 Reduction Cascade

The imaging and spectroscopic modes are separated, they both have their own calibration files, and their own reduction cascade. These reduction cascades are described on figure 8.3.1 in imaging and figure 8.3.2 in spectroscopy.

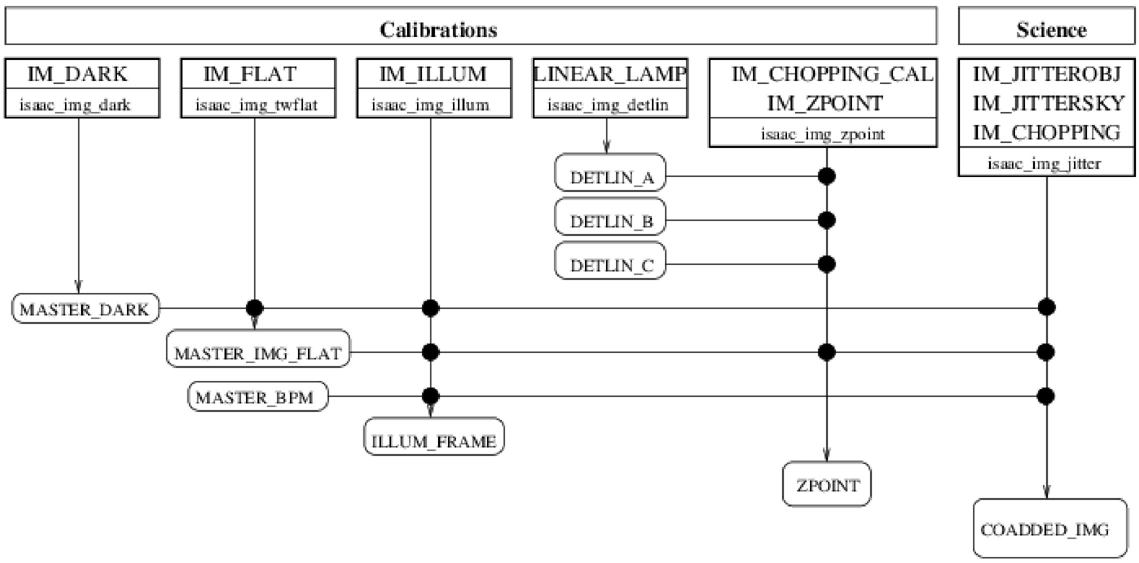

Figure 8.3.1: ISAAC Association Map in imaging mode

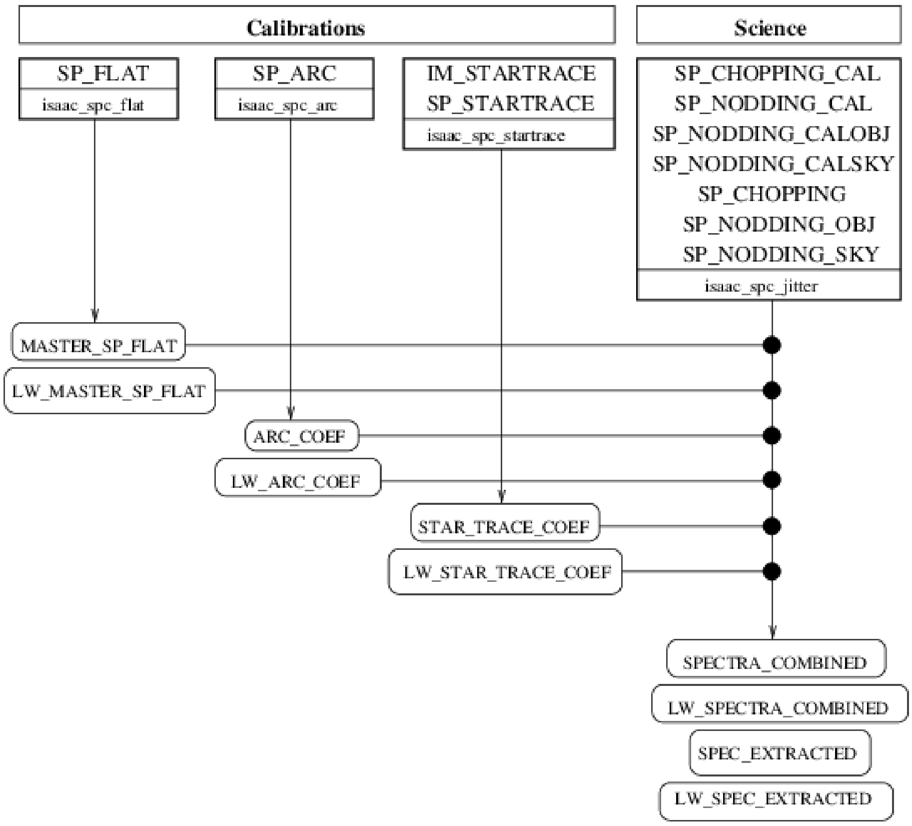

Figure 8.3.2: ISAAC Association Map in spectroscopic mode

9 Pipeline Recipe Interfaces

In this section we provide for each recipe examples of the required input data. In the following we assume that /path_file_raw/filename_raw.fits and /path_file_cdb/filename_cdb.tfits are existing FITS files.

We also provide a list of the pipeline products for each recipe, indicating their default recipe name, the value of the FITS keyword HIERARCH ESO PRO CATG (in short PRO.CATG) and a short description.

For each recipe we also list the input parameters (as they appear in the recipe configuration file), the corresponding aliases for the command line usage, and their default values. Also quality control parameters are listed. Those are stored in relevant pipeline products. More information on instrument quality control can be found on http://www.eso.org/qc

In addition to the products mentioned below, all recipes produce a PAF (VLT parameter file) which is an intermediate pipeline data file containing quality control parameter values.

9.1 isaac_img_dark

This recipe creates a master dark image, and computes the Read-Out Noise of the detector.

9.1.1 Input

This recipe expects input frames classified as IM_DARK. These frames are first classified by the recipe by different settings (DIT, NDIT, read-out mode), each setting is then reduced separately.

9.1.2 Output

For each setting, a master dark image named isaac_img_dark_setxx_avg.fits (PRO CATG = MASTER_DARK) is created where xx is the setting number (01, 02, ..., number of settings).

9.1.3 Quality control

The quality control parameters are computed for each setting. Within each setting, the successive pairs are used to compute RON values.

- QC DARKMED: Mean of the median values of the different input images
- QC DARKSTDEV: Standard deviation of the median values of the different input images
- QC RONi: Read Out Noise value computed on pair number i in LW mode
- QC LL RONi: Read Out Noise value computed on the lower left quadrant on pair number i in SW mode
- QC LR RONi: Read Out Noise value computed on the lower right quadrant on pair number i in SW mode

- QC UL RONi: Read Out Noise value computed on the upper left quadrant on pair number i in SW mode
- QC UR RONi: Read Out Noise value computed on the upper right quadrant on pair number i in SW mode

9.1.4 Parameters

- nsamples: Number of samples used to compute the RON (default is 100)
- hsize: Half-size of the boxes used to compute the RON (default is 6 in SW and 2 in LW)

9.2 isaac_img_twflat

This recipe creates a master flat field, a bad pixels map and is used to monitor the odd/even effect.

9.2.1 Input

This recipe expects input frames classified as IM_FLAT or POL_FLAT, and optionally dark frames classified as MASTER_IMG_DARK. If the dark frames are provided, their number must be either one or the same number as flat frames.

The frames are first classified by the recipe by different filters, each filter is then reduced separately.

9.2.2 Output

For each setting, the following products are created (xx identifies the setting number):

A master flat image named isaac_img_twflat_setxx.fits (PRO CATG = MASTER_IMG_FLAT).

Optionally, a bad pixels map image named isaac_img_twflat_setxx_bpm.fits (PRO CATG = MASTER_BPM).

Optionally, an image with the constant value of the fit named isaac_img_twflat_setxx_inter.fits (PRO CATG = MASTER_IMG_FLAT_INTERC).

Optionally, an image with the error bar of the fit for each pixel named isaac_img_twflat_setxx_errmap.fits (PRO CATG = MASTER_IMG_FLAT_ERRMAP).

9.2.3 Quality control

- QC FILTER OBS: The name of the filter used
- QC OBJECTIVE: The objective used
- QC TWFLAT MEDMIN: The minimum value of the medians of the input flat images
- QC TWFLAT MEDMAX: The maximum value of the medians of the input flat images
- QC TWFLAT MEDAVG: The average value of the medians of the input flat images

- QC TWFLAT MEDSTDEV: The standard deviation of the medians of the input flat images
- QC TWFLAT NBADPIX: The number of bad pixels detected
- QC ODDEVEN LLMAX: The maximum of the odd/even effect values computed in the lower left quadrants
- QC ODDEVEN LLMEAN: The average of the odd/even effect values computed in the lower left quadrants
- QC ODDEVEN LLSTDEV: The standard deviation of the odd/even effect values computed in the lower left quadrants
- QC ODDEVEN LRMAX: The maximum of the odd/even effect values computed in the lower right quadrants
- QC ODDEVEN LRMEAN: The average of the odd/even effect values computed in the lower right quadrants
- QC ODDEVEN LRSTDEV: The standard deviation of the odd/even effect values computed in the lower right quadrants
- QC ODDEVEN ULMAX: The maximum of the odd/even effect values computed in the upper left quadrants
- QC ODDEVEN ULMEAN: The average of the odd/even effect values computed in the upper left quadrants
- QC ODDEVEN ULSTDEV: The standard deviation of the odd/even effect values computed in the upper left quadrants
- QC ODDEVEN URMAX: The maximum of the odd/even effect values computed in the upper right quadrants
- QC ODDEVEN URMEAN: The average of the odd/even effect values computed in the upper right quadrants
- QC ODDEVEN URSTDEV: The standard deviation of the odd/even effect values computed in the upper right quadrants

9.2.4 Parameters

- t: Low and high thresholds for the bad pixels detection (default is 0.5,2.0)
- prop: Flag to activate the proportional fit (default is FALSE)
- bpm: Flag to activate the bad pixels map computation (default is FALSE)
- errmap: Flag to activate the error map creation (default is FALSE)
- intercept: Flag to activate the intercept map creation (default is FALSE)

9.3 isaac_img_slitpos

This recipe is used to get the precise positions of the edges of the slit.

9.3.1 Input

This recipe expects input frames classified as SLIT_IMG. The first frame may be a dark frame, the mode keyword is checked to detect that.

Each non-dark frame is an image of a vertical slit and is reduced separately.

9.3.2 Output

For each slit image, a table named isaac_img_slitpos_xx.fits (PRO CATG = SLIT_DESC) is produced. This table contains 4 columns as shown in Figure 9.3.0.

```
#
# File              isaac_img_slitpos_12.tfits
# extensions        1
# -----------------------------------------------
# XTENSION          1
# Number of columns 4
#
 SLIT_Y| SLIT_LEFT|SLIT_CENTER SLIT_RIGHT
     33|   524.204|      525.39    526.575
     34|   524.195|      525.38    526.556
     35|   524.185|     525.371    526.557
     36|   524.176|     525.362    526.547
     37|   524.167|     525.352    526.538

                          |

                          |

    841|   516.836|    517.894|   518.953
    842|   516.827|    517.885|   518.944
    843|   516.818|    517.876|   518.934
    844|   516.809|    517.867|   518.925
    845|   516.799|    517.857|   518.915
```

Figure 9.3.0: Table created by the slit detection

9.3.3 Quality control

- QC SLIT POSANG: The angle in degrees with th horizontal
- QC SLIT XPOS: The X position in pixels of the slit center
- QC SLIT YPOS: The Y position in pixels of the slit center

9.3.4 Parameters

- slit_w: The slit width in pixels (default is 20)
- prod: Flag to activate the product creation (default is TRUE)

9.4 isaac_img_detlin

This recipe computes the detector non-linearity.

9.4.1 Input

This recipe expects input frames classified as LINEAR_LAMP or LINEAR_DARK. There must be the same number of frames with both tags.

9.4.2 Output

For each pixel, the values are fitted with the increase of the lamp intensity by a 2nd degree polynomial: $F(x) = a + b.x + c.x^2$ where F(x) is the pixel intensity and x is the illumination.

The three images containing the a, b and c coefficients are produced with the PRO CATG = DETLIN_A, DETLIN_B and DETLIN_C and are named isaac_img_detlin_A.fits, isaac_img_detlin_B.fits and isaac_img_detlin_C.fits

A fourth image containing the error on the fit is saved with PRO CATG = DETLIN_Q and named isaac_img_detlin_Q.fits.

9.4.3 Quality control

- QC DETLIN MEDA: The median value of the a coefficients
- QC DETLIN MEDB: The median value of the b coefficients
- QC DETLIN MEDC: The median value of the c coefficients
- QC DETLIN MEDQ: The median value of the errors
- QC DETLIN LAMP: The lamp stability

9.4.4 Parameters

- force: Flag to compute the results even if the lamp is not considered stable

9.5 isaac_img_illum

This recipe computes the response of the detector from its illumination on a grid of standard stars.

9.5.1 Input

This recipe expects input frames classified as IM_ILLUM for the illumination frames. It also accepts bad pixels map (MASTER_BPM), a flat field (MASTER_IMG_FLAT) and/or a dark (MASTER_DARK).

9.5.2 Output

The two products are the image that gives the polynomial illumination on the whole detector named isaac_img_illum.fits (PRO CATG = ILLUM_FRAME) and a table named isaac_img_illum_flux.fits containing the flux and the position of the standard starts in the input frames (PRO CATG = ILLUM_FLUX) as shown in Figure 9.5.0.

```
#
# file           isaac_img_illum_flux.fits
# extensions     1
# ----------------------------------------
# XTENSION       1
# Number of columns 3
#
      POSX        POSY|      FLUX
       512         530|   30961.6
       129         146|   19073.6
       385         148|   20640.2
       640         148|     21213
       895         148|   16512.2
       894         403|     15626
       639         403|   17055.6
       385         403|   16164.4
       129         402|   21225.8
       129         657|   20937.6
       304         657|     15579
       639         657|   17635.6
       894         658|   12463.2
       894         914|   12702.2
       639         913|   14512.6
       384         912|   16672.6
       127         914|   17953.8
```

Figure 9.5.0: Table created by the illumination recipe

9.5.3 Quality control

If I is the illumination of the detector at position (x,y) in pixels: $I(x, y) = a + b.x + c.y + d.x.y + e.x^2 + f.y^2$

- QC ILLUM1: a
- QC ILLUMX: b
- QC ILLUMY: c
- QC ILLUMXY: d

- QC ILLUMXX: e
- QC ILLUMYY: f

9.5.4 Parameters

- star_r: The star radius in pixels (default is 10.0)
- bg_r1: The internal radius in pixels of the ring used for the background computation (default is 12.0)
- bg_r2: The external radius in pixels of the ring used for the background computation (default is 30.0)
- s_hx: X half size in pixels of the box used for the star detection (default is 50).
- s_hy: Y half size in pixels of the box used for the star detection (default is 50).
- pos_x: X position of the star in the first image (default is -1 for unknown)
- pos_y: Y position of the star in the first image (default is -1 for unknown)

9.6 isaac_img_zpoint

This recipe computes the Zero Point.

9.6.1 Input

The recipe expects a set of 5 frames with a standard star exposure for each. The first frame must have the standard star around the center, for the other frames, the star appears usually on the 4 quadrants of the detector. Thos frames are tagged with IM_ZPOINT in non-chopping mode, or IM_CHOPPING_CAL in chopping mode. The recipe also expects the FITS calibration file (STDSTARS_CATS) containing the list of known standard stars with their positions and magnitudes in the different bands.

The recipe also accepts detector linearity coefficients (tagged with DETLIN_A, DETLIN_B and DETLIN_C) or flat field (tagged with MASTER_IMG_FLAT).

9.6.2 Output

Two files are produced. The first is a table named isaac_img_zpoint.fits containing for each standard star the photometry computed as shown in Figure 9.6.1 (PRO CATG = ZP_TAB).

The second one is an image named isaac_img_zpoint_check.fits with the extracted stars used to verify that the proper stars have been used for the computation (PRO CATG = ZPOINT).

```
#
# file            isaac_img_zpoint.fits
# extensions      1
# --------------------------------------------
# XTENSION        1
# Number of columns 0
#
      POSX        POSY|    ZPOINT|      FLUX|      PEAK|       BGD|     FWHMX|     FWHMY
   470.195     536.453|   24.8157|   16939.6|   380.167|   30.6667|   4.66767|   5.66409
   774.541     841.458|   24.3306|     11451|       316|       -33|   3.29543|   4.76714
   774.677     841.338|   24.5074|   12751.6|   294.667|  -42.6666|   3.29642|   4.75382
   164.349     841.806|   24.7252|   15534.9|   392.833|   46.3334|   3.47319|   5.11441
   164.525     841.794|   24.7316|   15675.9|   342.833|   8.33325|   3.31606|   5.47442
   165.165     231.566|    24.761|   16107.3|       560|        -7|   3.57857|   4.63121
   165.101     231.454|   24.6282|   14253.1|     539.5|  -9.66663|   3.33976|   4.54491
   774.399     231.934|   24.6359|   14353.0|   373.667|   9.66663|   4.57956|   6.45542
```

Figure 9.6.1: Example of the output produced by *isaac_img_zpoint*

9.6.3 Quality control

- QC FILTER OBS: The name of the filter used in the header
- QC FILTER REF: The name of the filter actually used to get the informations from the catalog
- QC AMBI RHUM AVG: The humidity average
- QC ZPOINT: The computed zero point
- QC ZPOINT ATX0: The computed zero point with the extinction correction
- QC ZPOINTRMS: The error on the zero point
- QC FLUX MED: The median of the computed flux values
- QC STDNAME: The name of the standard star
- QC SPECTYPE: The spectral type of the standard star
- QC STARMAG: The magnitude of the standard star
- QC CATNAME: The catalog name where the star has been found
- QC GRADX: The flux gradient in X
- QC GRADY: The flux gradient in Y
- QC GRADDX: The error on the flux gradient in X
- QC GRADDY: The error on the flux gradient in Y
- QC FWHM MEAN: The average of the FWHM

9.6.4 Parameters

- star_r: Star radius in pixels (default is 30)
- bg_r1: Background ring internal radius in pixels (default is 40)
- bg_r2: Background ring external radius in pixels (default is 60)
- ra: RA position (default is 999.0 for unknown)
- dec: DEC position (default is 999.0 for unknown)
- mag: star magnitude (default is 99.0 for unknown)
- sx: Search size in pixels in the x direction (default is 10)
- sy: Search size in pixels in the y direction (default is 10)
- check_im: Flag to activate the 'check' image creation (default is FALSE)

9.7 isaac_img_jitter

This recipe is the science recipe in imaging. It reduces the data to create a clean combined image.

9.7.1 Input

The recipe expects frames marked with IM_JITTEROBJ and IM_JITTERSKY (whether they are objects or sky frames) in non-chopping mode, or frames marked as IM_CHOPPING in chopping mode.

It also accepts bad pixels map (MASTER_BPM), flat field (MASTER_IMG_FLAT) and/or dark frames (MASTER_DARK).

9.7.2 Output

The produced image is the combined image named isaac_img_jitter.fits (PRO CATG = COADDED_IMG). Besides, a table named isaac_img_jitter_bg.fits with the background values (PROI CATG = SKY_TAB) and a table named isaac_img_jitter_stars.fits with the detected stars and there photometry are produced (see Figure 9.7.1).

9.7.3 Quality control

- QC BACKGD MEAN: The mean of the background values
- QC BACKGD STDEV: The standard deviation of the background values
- QC BACKGD INSTMAG: The background magnitude
- QC NBOBJS: The number of detected objects

```
#
# file           isaac_img_jitter_bg.fits
# extensions     1
# --------------------------------------------
# XTENSION       1
# Number of columns 1
#
   SKY_BG
  5877.27
  5918.32
  5921.87
  5955.77
  5991.52
   6027.7
  6037.29
  6009.99
  5968.26
  5973.07
```

```
#
# file           isaac_img_jitter_stars.fits
# extensions     1
# --------------------------------------------
# XTENSION       1
# Number of columns 5
#
     POS_X       POS_Y|      FWHM_X|      FWHM_Y|        FLUX
   966.229    29.3925|    4.15001|    4.34206|     70346.9
   895.691    19.3439|    2.16332|    2.53739|     1423.17
   701.365    20.5227|    5.74410|    7.50237|     733.240
   951.035    32.0136|    1.99187|    1.84194|     229.722
   418.532    149.338|    2.02365|    2.52913|     2139.38
   853.567    180.459|    2.16345|    2.15424|     1176.63
   1131.45    181.211|    2.06202|    4.27282|     895.869
    899.77    197.154|    2.39389|    2.69425|     1526.55
   908.003    198.958|    2.27176|    2.41846|     342.383
    824.77    248.541|    2.60779|    2.27524|     1749.04
   261.065    253.924|    4.12744|    2.70624|     2677.574
   827.847    275.022|    3.53761|    2.94732|     731.066
   402.625    201.944|    3.57243|    3.17490|     304.599
      1143    293.251|    2.54508|    2.25019|     532.816
    352.84    326.215|    2.41201|    2.41321|     2534.67
   395.363    342.226|    4.53454|    3.80684|     10062.4
   297.972    334.979|    5.06531|     3.2535|     213.463
   462.585    337.361|    2.34536|    3.04701|     1444.01
   872.081    383.435|    3.53319|    2.06865|     811.721
   636.033    392.954|     2.5139|    2.35479|      302.91
   641.174    397.042|    2.86443|    2.57792|     314.505
   150.542    425.336|    3.57191|    5.33918|     393.571
   112.933    429.107|    3.40157|    2.35309|     603.473
```

Figure 9.7.1: Products of *isaac_img_jitter*

- QC IQ: The Image Quality
- QC FWHM PIX: The average FWHM in pixels
- QC FWHM ARCSEC: The average FWHM in arc seconds
- QC FWHM MODE: The FWHM mode
- QC NB_OBJ_F: The number of object frames
- QC NB_SKY_F: The number of sky frames
- QC NB_REJ_F: The number of rejected frames

9.7.4 Parameters

- off: The offsets text file (default is NULL)
- objs: The correlation object(s) position(s) (default is NULL)
- oddeven: Flag to activate the odd/even effect correction
- sky_par: The sky filtering parameters (minimum number of frames, half size of the window for the running filter, number of low rejections, number of high rejections) (default is 10,7,3,3)
- xcorr: Cross correlation search and measure sizes in x and y (default is 40,40,65,65)

- comb_meth: Combination method for the stacking (union/inter/first) (default is union)
- rej: High and Low rejections for the stacking (default is 2,2)
- row_med: Flag to subtract the median of each row (default is TRUE)

9.8 isaac_spc_flat

This recipe generates the spectroscopic flat field.

9.8.1 Input

The input frames expected by the recipe are tagged with SP_FLAT. They are first classified by settings (slit, resolution, central wavelength), and every setting is reduced separately.

9.8.2 Output

For each setting (xx being the setting number), the master flat named isaac_spc_flat_setxx.fits (PRO CATG = MASTER_SP_FLAT or LW_MASTER_SP_FLAT) is created.

9.8.3 Quality control

- QC FILTER OBS: The filter used
- QC SPECFLAT NCOUNTS: The average of the median values of the intermediate flatfields
- QC SPECFLAT STDEV: The standard deviation of the median values of the intermediate flatfields

9.8.4 Parameters

- thresholds: Low and high thresholds (default is 0.01,3.0)
- fit_order: Order of the fit (default is 3)
- fit_size: Size of the central window where the fit is applied (default is 200)
- zone: Zone to consider for normalisation (default is 256,256,768,768)
- offset: Zone in pixels to discard at top and bottom for the fit (default is 40)

9.9 isaac_spc_arc

This recipe both computes the wavelength calibration and the slit curvature distortion using vertical lines.

9.9.1 Input

The expected frames must be tagged with SP_ARC. They are either frames obtained with a Xenon lamp, an Argon lamp, with both, or none (dark). The recipe also needs the Xenon and Argon lines FITS catalogs (CALPRO_XE_CATALOG and CALPRO_AR_CATALOG).

9.9.2 Output

For each setting, the produced table is named isaac_spc_arc_setxx.fits. It contains both the distortion polynomial and the dispersion relation as shown in Figure 9.9.1 (PRO CATG = ARC_COEF or LW_ARC_COEF).

Additionally, on request, the image with the distortion corrected arcs is produced under the name isaac_spc_arc_setxx_corr.f
(PRO CATG = ARC_CORRECT or LW_ARC_CORRECT).

```
#
# File              isaac_spc_arc_set1_pair1.fits
# extensions        1
# --------------------------------------------
# XTENSION          1
# Number of columns 4
#
Degree_of_x|Degree_of_y poly2d_coef|WL_coefficients
          0|          0     -6.8264|        14062.9
          1|          0    0.995758|        4.76237
          0|          1  0.00239726|   -4.30157e-06
          1|          1 -9.70806e-06|   -7.51516e-09
          2|          0 2.25572e-05|              0
          0|          2  3.3765e-06|              0
```

Figure 9.9.1: Example of the output produced by *isaac_spc_arc*

9.9.3 Quality control

The dispersion relation is the following: $Wavelength(pix) = a + b.pix + c.pix^2 + d.pix^3$

The distortion relation is the following: $X(x, y) = A + B.x + C.Y + D.X.Y + E.X^2 + F.Y^2$

- QC FILTER OBS: The filter used
- QC LAMP: The lamp used
- QC DISP XCORR: The cross-correlation factor
- QC DISP NUMCAT: The number of lines in the catalog
- QC DISP NUMMATCH: The number of matched lines in the catalog
- QC DISP STDEV: The error on the dispersion relation
- QC DISPCO1: The factor a of the dispersion relation
- QC DISPCO2: The factor b of the dispersion relation

- QC DISPCO3: The factor c of the dispersion relation
- QC DISPCO4: The factor d of the dispersion relation
- QC DIST1: The factor A of the distortion relation
- QC DISTX: The factor B of the distortion relation
- QC DISTY: The factor C of the distortion relation
- QC DISTXY: The factor D of the distortion relation
- QC DISTXX: The factor E of the distortion relation
- QC DISTYY: The factor F of the distortion relation
- QC SATUR NBPIX: The number of saturated pixels
- QC WLEN: The central wavelength
- QC ARCS NUM: The number of detected arcs
- QC ARCSi XPOS: The positions of the arcs
- QC ARCSi FWHM: The FWHM of the arcs
- QC ARCSi FLUX: The flux of the arcs
- QC ARCS NUMGOOD: The number of valid arcs
- QC FWHM MED: The median of the FWHMs

9.9.4 Parameters

- rej: Zones of the image to reject (default is -1,-1,100,100)
- subdark: Flag to apply an automatic dark subtraction (default is FALSE)
- arc_max_w: Arc maximum width in pixels (default is 33)
- arc_kappa: kappa for the threshold used for arcs detection (default is 0.33)
- out_corr: Flag to produce distortion corrected images (default is FALSE)

9.10 isaac_spc_startrace

This recipe computes the spectra curvature.

9.10.1 Input

The recipe expects in input n imaging frames with a star moving along the slit (tagged IM_STARTRACE), n spectra in low resolution and n spectra in medium resolution also moving along the slit (tagged SP_STARTRACE).

9.10.2 Output

A series of tables are produced. isaac_spc_startrace_2dpoly_LR.fits and isaac_spc_startrace_2dpoly_MR.fits contain the 2D distortion polynomials for the LR and MR modes (PRO CATG = STAR_TRACE_COEF or LW_STAR_TRACE_COEF).

isaac_spc_startrace_positions_LR.fits and isaac_spc_startrace_positions_MR.fits both contain the positions of the stars and the spectra in the LR and MR modes (PRO CATG = STAR_TRACE_POSI or LW_STAR_TRACE_POSI).

isaac_spc_startrace_shapes_LR.fits and isaac_spc_startrace_shapes_MR.fits contain the polynomial shapes of the different spectra in LR and MR (PRO CATG = STAR_TRACE_SHAPE or LW_STAR_TRACE_SHAPE) .

Those tables are shown in Figure 9.10.1.

Positions table in LR

```
#
# file            isaac_spc_startrace_positions_LR.fits
# extensions      1
# ---------------------------------------------
# XTENSION        1
# Number of columns 2
#
Star_positions|Spec_positions
      870.248|       886.93
      795.642|      813.434
      721.536|      739.749
      647.417|      665.812
      573.433|      591.642
      499.339|      516.855
      425.438|      442.269
      351.552|      367.301
      277.317|      291.991
      203.233|      215.956
      128.632|      139.663
```

Shapes table in LR

```
#
# file            isaac_spc_startrace_shapes_LR.fits
# extensions      1
# ---------------------------------------------
# XTENSION        1
# Number of columns 11
#
    Spec_1      Spec_2|     Spec_3|     Spec_4|     Spec_5|     Spec_6|     Spec_7|     Spec_8|     Spec_9|    Spec_10|    Spec_11
   883.953     809.625|    735.301|    660.616|    585.087|    511.064|    435.463|    359.778|    283.839|    207.125|    129.773
0.00710583 0.00522213|0.00705518|0.00921685| 0.0103134| 0.0100641| 0.0145915| 0.0160697| 0.0177427| 0.0206254| 0.0239382
6.07747e-06|4.68267e-06|3.44141e-06|6.25681e-07|8.52176e-07|4.46881e-06|-3.1277e-06|-2.98782e-06|-4.09343e-06|-7.56037e-06|-1.09371e-05
-9.82438e-10|-1.30526e-09|-1.35689e-09|-2.89696e-10|-1.44325e-09|-4.74018e-09|-5.94103e-10 -1.85377e-09|-2.07343e-09|-9.69782e-10|-3.1763e-10
```

Distortion polynomial in LR

```
#
# file            isaac_spc_startrace_2dpoly_LR.fits
# extensions      1
# ---------------------------------------------
# XTENSION        1
# Number of columns 3
#
Degree_of_x|Degree_of_y poly2d_coef
          0|          0      -8.337
          1|          0   0.0158603
          0|          1     1.00406
          1|          1 -4.42009e-06
          2|          0 -3.14188e-06
          0|          2 -1.81933e-07
```

Figure 9.10.1: Tables produced by *isaac_spc_startrace*

9.10.3 Quality control

The relation between the spectrum position and the star position is: $Spec(star) = a + b.star + c.star^2$

The spectrum distortion is given by: $Y(x, y) = A + B.x + C.y + D.x.y + E.x^2 + F.y^2$

- QC CORR_IS1: The a coefficient
- QC CORR_IS2: The b coefficient
- QC CORR_IS3: The c coefficient
- QC FITMSE: The error on the fit
- QC DIST1: The A coefficient
- QC DISTX: The B coefficient
- QC DISTY: The C coefficient
- QC DISTXY: The D coefficient
- QC DISTXX: The E coefficient
- QC DISTYY: The F coefficient

9.10.4 Parameters

- deg: The degree of the polynomial to fit (default is 3)
- spec_width: The spectrum width in pixels (default is 40)
- reject_left: The number of pixels to reject on the left (default is -1 for unknown)
- reject_right: The number of pixels to reject on the right (default is -1 for unknown)
- display: Flag to activate the plotting mode

9.11 isaac_spc_jitter

This recipe is the science data reduction recipe. It produces the combined image and extracts the spectrum. It can be used to reduce observations as well as standard star observations.

9.11.1 Input

The expected frames must be tagged with SP_NODDINGOBJ and SP_NODDINGSKY in non chopping mode, and with SP_CHOPPING in chopping mode.

If the observed object is a standard star, the frames must be tagged with SP_NODDING_CAL, SP_NODDING_CALOBJ, SP_NODDING_CALSKY or SP_NODDING_FLUX in non-chopping mode, and with SP_CHOPPING_CAL in chopping mode.

The recipe also needs the OH lines FITS catalogs (CALPRO_OH_CATALOG) for wavelength calibration and spectral energy distributions (SEDS) and standard stars catalogs (STDSTARS_CATS) for the response function computation.

The recipe also accepts calibration files like an arc file (ARC_COEF or LW_ARC_COEF), a star trace file (STAR_TRACE_COEF or LW_STAR_TRACE_COEF) or a flat field (MASTER_SP_FLAT or LW_MASTER_SP_FLAT).

9.11.2 Output

The produced image is the combined image. It is named isaac_spc_jitter_combined.fits (PRO CATG = OBS_COMBINED or SPECTRA_COMBINED for standard stars).

The extracted spectrum is stored in a table named isaac_spc_jitter_extracted.fits as shown in Figure 9.11.1 (PRO CATG = OBS_EXTRACTED or SPEC_EXTRACTED for standard stars).

```
#
# file           isaac_spc_jitter_extracted.fits
# extensions     1
# -------------------------------------------
# XTENSION       1
# Number of columns 3
#
X_coordinate|Extracted_spectrum_value|Sky_spectrum
     13428.9|                       0           0
       13436|                       0           0
     13443.2|                 34.9542    0.121982
     13450.3|                 24.5074    0.262553
     13457.4|                 27.6124   -0.347931
     13464.6|                 14.5363    0.175988
     13471.7|                 15.3864   -0.431524
     13478.8|                 10.1013   -0.149821
       13486|                   11.35    0.404973
     13493.1|                 17.3553   0.0757230
     13500.2|                 22.8754   -0.163189
     13507.4|                 16.7424    0.182963
     13514.5|                 23.9685   -0.105734
     13521.7|                 20.6795    0.051249
     13528.8|                 16.9323   0.0112508
       13536|                  12.961    0.201395
     13543.1|                 23.6014    -0.32384
           :                       :           :
```

Figure 9.11.1: Table produced by *isaac_spc_jitter*

9.11.3 Quality control

The computed dispersion relation is the following: $Wavelength(pix) = a + b.pix + c.pix^2 + d.pix^3$

- QC FILTER OBS: The filter used for observation
- QC FILTER REF: The filter used to retrieve the magnitude
- QC DISPCO1: The a coefficient in the dispersion relation
- QC DISPCO2: The b coefficient in the dispersion relation
- QC DISPCO3: The c coefficient in the dispersion relation

- QC DISPCO4: The d coefficient in the dispersion relation
- QC WLEN: The central wavelength
- QC DISP XCORR: The cross correlation factor
- QC WLMETHOD: The wavelength calibration method used
- QC SPEC INTENSi: The spectra intensities
- QC STDNAME: The standard star name
- QC SPECTYPE: The standard star spectral type
- QC STARMAG: The standard star magnitude retrieved from the catalog

9.11.4 Parameters

- oddeven: Flag to activate the odd/even correction (default is FALSE)
- wavecal: Wavelength calibration method (default is sky)
- wc_rej: Wavelength calibration rejected zones (default is -1,-1,50,50)
- saa_refine: Flag to refine the spectra positions before the shift-and-add (default is TRUE)
- saa_rej: Low and high rejections in percent for the shift-and-add (default is 0.1,0.1)
- spec_pos: Spectrum position (default is -1 for unknown)
- spec_width: Spectrum width in pixels (default is 10)
- sky_hi_width: Sky zone width above the spectrum (default is 10)
- sky_lo_width: Sky zone width under the spectrum (default is 10)
- sky_hi_dist: Sky zone distance to the spectrum (default is -1)
- sky_lo_dist: Sky zone distance to the spectrum (default is -1)
- display: Flag to activate the plotting mode (default is FALSE)
- std: Flag to activate the standard star mode (default is FALSE)
- mag: Standard star magnitude (default is -1 as unknown)

10 Algorithms

In this section the data reduction procedures applied by the 11 pipeline recipes are described in some detail.

10.1 General Algorithms

10.1.1 Cross-correlation

The cross-correlation is used both in spectroscopy mode on 1D signals and in imaging mode on images.

In spectroscopy, it is used in the wavelength calibration to find the best match between the observed spectrum and the theoretical one.

In imaging, it is used to find the exact shift between two images containing the same stars field. This part is describing the 2D method, the 1D method is based on the same concepts.

We suppose that we have

- two images A and B that represent about the same field, but that are shifted by (X,Y) pixels
- a rough estimate of the shift (Xest, Yest)
- a search size (sx, sy) and a measure size (mx, my)
- a correlation point in the first image (Cx, Cy)

Each pixel in a box of size (sx, sy) around the position (Cx+Xest, Cy+Yest) in image B is candidate for being the one that looks like the pixel at position (Cx, Cy) in image A. For each of those pixels (i, j), a cross-correlation factor XC(i, j) is computed. The pixel with the lowest factor 'wins'.

The cross-correlation factor is the normalised sum of the square difference of the pixels of the two images in a box of size (mx, my). The more the two windows look alike, the lower the factor will be.

The figure 10.1.1 gives an illustration of the procedure.

To have a sub-pixel precision a fit of the cross-correlation values is computed around the minimum to get the 'real' minimum, and its position that corresponds to the precise sub-pixel shift we are looking for.

10.1.2 Distortion computation

Both the slit curvature (vertical lines) and the star trace (horizontal lines) distortions are estimated using a set of bright (curved) lines spread on the whole detector. These bright lines are detected, and the positions of regularly spaced points on these lines are associated to the positions that these points would have on a non-distorted image. These two grids are fitted to obtain the 2D distortion polynomial. Figure 10.1.2 illustrates the algorithm in the slit curvature case.

In this case, the obtained 2D polynomial looks like:

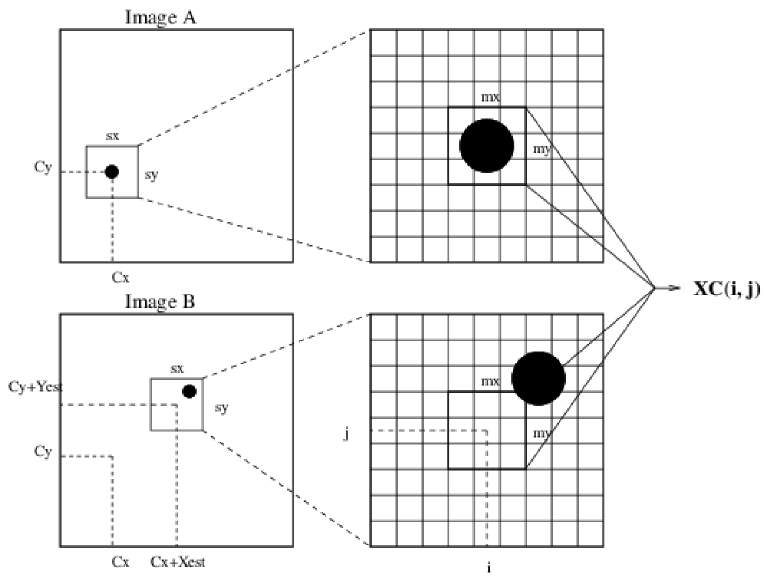

Figure 10.1.1: Cross-Correlation algorithm

$$Px(X,Y) = a + b.X + c.Y + d.XY + e.X^2 + f.Y^2$$

$$Py(X,Y) = Y$$

with a=-6.8264, b=0.995758, c=0.00239726, d=-9.70806e-06, e=2.25572e-05, f=3.3765e-06.

10.1.3 Wavelength calibration

The wavelength calibration is either computed using some calibration data obtained with Argon or Xenon lamps or using directly the sky emission lines obtained from the observation. These lines are then compared to a theoretical signal extracted from catalogs of lines to obtain the wavelength calibration. The advantage of using the sky lines is that the conditions are exactly the same for the observation. This assures a better accuracy.

The solution is a 3rd degree polynomial obtained by applying the following steps:

- Image Preparation

  Both arc images in the thermal regime (defined as wavelengths longer than 2 microns) and OH images may have noticeable low-frequency features, which may cause misalignment of lines, typically in the upper part of the spectrum. A low-pass filter is applied and the result is subtracted of that from the image. Experiments show that a filter with window size of 8 times the slit width (in pixels) applied to the collapsed spectroscopy image yield the best results. Since some information is lost in this filtering it is only applied when a low-frequency feature is expected, i.e. when calibrating with arcs in the thermal regime and with OH sky lines.

  A dark-corrected image may have negative intensities. Since they are considered meaningless they are removed (set to zero).

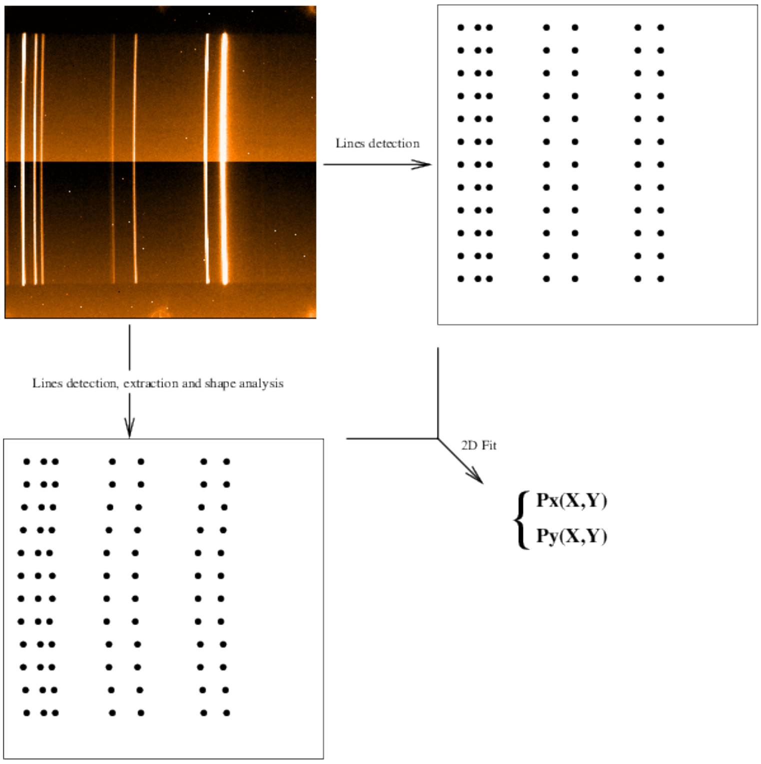

Figure 10.1.2: Distortion estimation algorithm

A few other cosmetic corrections are applied to the image to prepare it for the next steps like the removal of the 10 pixels on the right and on the left of the spectrum.

- Signal Generation From the Catalog

  An initial guess of the wavelength range is obtained from the physical model of the instrument. This physical model is the same as the one used for the ETC. This first guess gives the part of the catalog that has to be used to generate the artificial spectrum.

  The generated spectrum is convolved with a Gaussian function (with a sigma equal to a quarter of the slit width) to keep the same line width as in the observation.

- Polynomial Solution

  To find the best 3rd degree dispersion relation that makes the observation look like the generated spectrum the most, the spectra for all possible solutions in the 4 dimensions space (4 coefficients of the polynomial) are generated and correlated with the artificial one. We are using optimisation techniques to avoid to cover the whole space for obvious efficiency reasons. Each correlation comes with a cross-correlation factor that indicate how similar the two signal are. The cross-correlation factor is a good indicator on the quality of the result.

- Quality of the Solution

  To estimate the quality of the method, it has been applied to a number of ISAAC spectroscopic images, namely 2601. About half of the images are OH sky lines mainly from 2001 while the rest are Xenon and/or Argon arcs mainly from 2002 and 2003.

34 of the images either caused the calibration to fail or to return a very small cross-correlation, $\chi < 0.15$. The images were inspected manually and found not to contain any spectroscopic information. The following analysis ignores these images. For the remaining, valid spectroscopic images χ has been observed to be in the range $0.35 < \chi < 0.96$. An approximate dispersion relation with χ closer to 0.15 than 0.35 should thus be regarded with skepticism. The wavelength calibration will however not discard an approximate dispersion relation on grounds of an insufficient cross-correlation.

The images have been grouped together such that images in the same group share instrument settings that give them similar accuracy in the approximate dispersion relation. The images are grouped together after the following criteria: Whether they are arcs or sky lines, whether they are produced with the short or long wavelength detector, whether they are in medium or low resolution. They are also grouped after the slit width, and in some cases the filter.

Table 10.1.0 summarises some key numbers for each group of images. When no grouping is done according to the filter, this field simply shows which filters are present in the group. The standard deviation is in pixels. C is the number of catalog lines, D is the number of catalog lines detected in the image. The ratio p_1/q_1 is formed between the first degree coefficients of the maximizing polynomial P and the initial estimate Q. Each number is an average of the values in the group, except $\chi_{\min}$ (the minimum χ among the files in the group) and the standard deviation, with the number of detected lines D instead being the total number of detected lines in the group.

Some comments on the information in Figure 10.1.0:

- The slit width affects the accuracy of the calibration: With a wider slit the line misalignment tends to increase. This is explained by the fact that a certain misalignment (in pixels) is a relatively smaller misalignment for wider slits. If the line profiles to be cross-correlated were perfect top-hats with identical intensity, then a misalignment of 1 pixel would decrease χ by 50% when the width of the line profile is 2 pixels (0.3"), while for a line profile of width 13.5 pixels (2"), the decrease is only 7.5%.
- In some cases the chosen filter effects the accuracy of the calibration. For arc images this is the case with a narrow slit and low resolution. Here a change from SH to SL filter increases the misalignment by about a factor three. This is most likely because the wavelength range of the SL filter includes a strong thermal background in which some accuracy is lost. For sky images each of the filters J, SH and SK result in calibrations with different accuracy, which may be explained by the varying levels of the background at the different wavelengths.
- The required number of iterations does not vary much with the instrument setting.
- The CPU time is measured in seconds on a 2.6 GHz Xeon running Linux.

The average line misalignment for all the calibrations is 0.66 pixels. The catalog of arc lines have lines with no known intensity. For detected lines of this kind the average line misalignment is about 1.5 times higher than the overall average. The same ratio is observed when comparing the average misalignment of lines with known and unknown intensity for each calibration. An increase in the misalignment of lines with unknown intensity is expected since the cross-correlation can do nothing to reduce such a misalignment. It is therefore difficult to say whether this increased misalignment also indicates that the approximate dispersion relation has a too low degree.

Files	Type	Light	Slit	Filter	χ	$\chi_{\min}$	StDev	D	C	p_1/q_1	N_{ite}	CPU
138	SWMR	lamp	0.3 to 0.8	J,SH,SK,SZ	0.77	0.55	0.21	8	11	0.997	2.24	5.9
118	SWMR	lamp	1	J,SH,SK,SZ	0.86	0.68	0.34	10	12	0.998	2.14	11.2
104	SWMR	lamp	2	J,SH,SK,SZ	0.87	0.65	0.98	9	11	0.998	2.21	18.2
72	LWMR	lamp	0.3	J+Block,SH,SL	0.78	0.70	0.53	6	14	1.029	2.44	8.1
169	LWMR	lamp	0.6 to 1.5	J+Block,SH,SK,SL	0.86	0.63	0.84	10	14	1.028	2.52	18.6
95	LWMR	lamp	2	J+Block,SH,SL	0.85	0.54	0.93	8	14	1.028	2.47	27.4
33	SWLR	lamp	0.3 or 0.6	J,J+Block,SH,SK	0.83	0.67	0.65	13	39	1.013	2.61	16.1
2	SWLR	lamp	0.3 or 0.6	SZ,Z	0.45	0.38	0.74	7	43	1.012	2.50	12.6
14	SWLR	lamp	0.8	SH,SK	0.85	0.76	0.96	13	34	1.013	2.64	24.3
114	SWLR	lamp	1	J,J+Block,SH,SK	0.83	0.56	1.13	13	36	1.012	2.46	24.4
3	SWLR	lamp	1	SZ	0.51	0.35	1.60	9	34	1.008	3.00	32.7
66	SWLR	lamp	1.5 or 2	J,SH,SK	0.86	0.70	1.51	9	35	1.011	2.62	39.3
5	SWLR	lamp	1.5 or 2	SZ	0.50	0.38	3.29	8	34	1.002	2.80	43.3
27	LWLR	lamp	0.3	SL	0.69	0.55	0.66	3	62	1.016	2.74	8.3
7	LWLR	lamp	0.6 or 1	SH	0.65	0.63	0.25	9	71	1.014	2.43	13.6
40	LWLR	lamp	0.6 or 1	SL	0.81	0.75	0.79	9	62	1.016	2.80	17.5
20	LWLR	lamp	2	SH,SL	0.62	0.56	2.62	5	64	1.015	2.85	39.7
23	SWMR	sky	0.6	SH	0.84	0.81	0.27	14	159	0.999	2.04	12.8
36	SWMR	sky	0.6	SK	0.75	0.69	0.55	4	64	1.000	2.92	13.5
1	SWMR	sky	0.6	J	0.83	0.83	1.12	8	147	0.999	2.00	10.4
46	SWLR	sky	0.6	J,SZ	0.53	0.48	1.00	2	1043	1.013	2.20	16.9
16	SWMR	sky	1	SK	0.87	0.87	0.16	5	97	1.002	3.00	22.3
32	SWMR	sky	1	J	0.69	0.56	2.00	13	127	0.998	2.00	13.6
579	SWLR	sky	1	SK	0.61	0.56	0.49	7	580	1.009	2.44	13.3
551	SWLR	sky	1	SH	0.68	0.67	0.82	13	1007	1.013	2.63	29.7
256	SWLR	sky	1	J	0.50	0.46	2.47	3	968	1.011	2.66	28.3

Table 10.1.0: Key numbers for various ISAAC wavelength calibrations.

10.2 Recipes Algorithms

10.2.1 isaac_img_dark

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.

The darks are reduced with the `isaac_img_dark` recipe. This recipe first separates the input list of frames by settings. A setting is defined by the DET DIT keyword (integration time), the DET NDIT and the read-out mode. All input frames with those matching keywords are in the same setting.

For each setting, the recipe produces one master dark that is nothing more than an average of the input files. The Read-Out Noise (RON) is also measured and written as QC parameter. The RON is computed for each quadrant of the detector in SW mode (4 measurements) and for the whole detector in LW mode.

In each setting, for each pair, the second frame is subtracted from the first. The following measurement is applied to the 4 quadrants in SW mode or to the image in LW mode:

- Generate 100 13x13 in SW or 100 5x5 in LW 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 multiplied by $\sqrt{\frac{NDIT}{2}}$.

The results are written as QC parameters. For example, an input data set containing 9 frames of 3 different settings in SW will generate 3 master darks (1 per setting) and 8 (2 pairs x 4 quadrants) RON measurements per setting (24 in total).

10.2.2 isaac_img_twflat

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.

The implemented algorithm is the following:

- The data are classified by filter, and each filter is separately reduced. The following steps apply for each filter.
- If provided, the dark frames are subtracted to the flat frames.
- The odd/even columns effect is monitored on the dark corrected frames.
- 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. This image is the master flat produced.

A master flat is produced for each subset of frames with the same filter.

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 too far from 1 as bad.

The odd-even effect is monitored by this recipe as QC parameters.

10.2.3 isaac_img_illum

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. Usually, the procedure is run once every six months and after major instrument interventions.

The recipe expects in input a series of standard star observations moved across the detector over a regularly spaced grid (around 20 positions). The recipe expects the first star around the center, if it is not, its position can be passed in parameters. Some calibration data (flat field, dark and bad pixels map) can be passed as well.

The implemented algorithm is the following:

- The calibrations (flat field, dark, bad pixels map) are used to correct the input raw frames.
- The precise positions of the star on the grid are derived from the images with the use of the header offsets and of the expectation that the first image has its star around the center.
- The photometry of each star is computed using a radius of 10 pixels for the flux and a ring of radii between 12 and 30 pixels for the background computation.
- A 2nd degree, two dimensions polynomial is fitted to the flux as a function of position on the array. The polynomial image is generated, normalised to a mean of unity and created as a product. An other product is the table containing the star positions and flux.

10.2.4 isaac_img_detlin

To estimate the non-linearity of the detector, series of images are taken with a lamp where the integration time is increased. Dark frames (lamp off) are taken at the same time with the same integration times as for the 'lamp on' frames. During the process of increasing the integration time, some images are taken at a regular period (typically every 5 frames) with a reference short integration time to verify the stability of the lamp during the whole process.

The recipe algorithm is the following:

- The dark frames are associated (and subtracted) to the 'lamp on' frames by using the DIT.

- The frames with the same DIT as the first frame are used to check the lamp stability over the whole observation: the average of the images should not vary more than 1 percent.
- The rest of the frames are used to compute the non-linearity: The evolution of the level of every pixel is fitted against the DIT. A second degree polynomial is fitted for every pixel. 4 images are produced: One for each polynomial coefficient and one for the error of the fit.

10.2.5 isaac_img_slitpos

On a regular basis, the observatory takes images of every slit through every filter. These images are processed with the `isaac_img_slitpos` recipe which measures the slit position.

The input file can contain a dark frame or not. If yes, this one is subtracted to the slit images (the other frames passed in input).

For every slit image, the slit is detected, and precisely analysed to get its precise position. The slit borders positions are stored in a FITS table.

To isolate the slit from other features in the image, a threshold is first applied. Then, successive morphological erosions (with a vertical kernel) are performed until there is one object left. The slit is then reconstructed with dilations (see Figure 10.2.1).

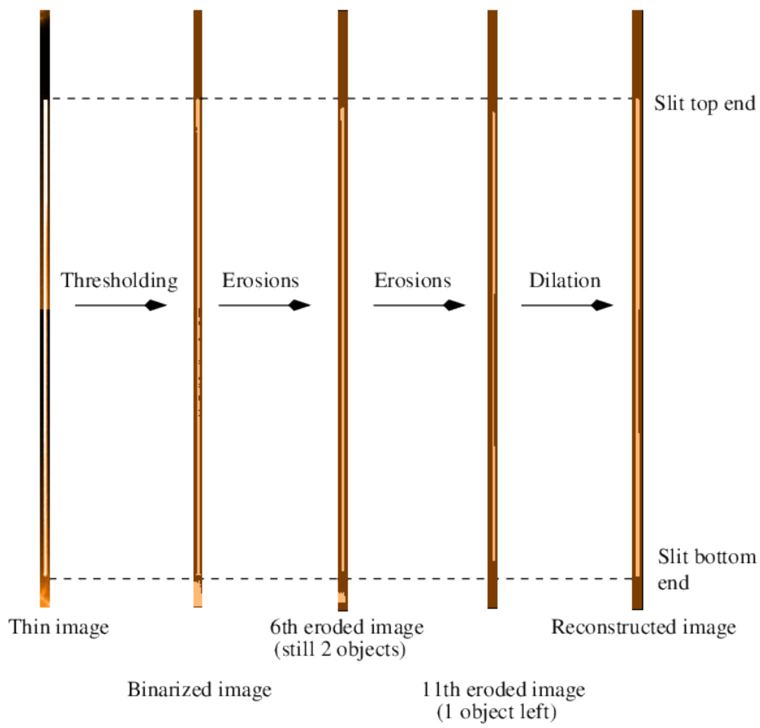

Figure 10.2.1: Slit detection

Once the slit is precisely isolated, the edges can easily be detected with a simple thresholding. For each pixel along the slit, the profile is analysed to get the left and the right positions on the slit.

Assuming that the slit is not curved, the left (resp. right) positions are linear-fitted and the fitted positions are written in a table.

10.2.6 isaac_img_zpoint

Standard stars are observed every night in the J, Js, H and Ks filters. For the Narrow Band 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 Zero Point.

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 implemented recipe is the following:

1. The detector non-linearity is corrected if the coefficients images are passed as calibration files to the recipe.
2. Compute the difference of the successive images (8 differences for 5 input images).
3. In each difference, locate the star around the expected pixel position.
4. Compute the background around the star, and the star flux.
5. Store the flux result in an output table.

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 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`.

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

10.2.7 isaac_img_jitter

The basic steps in reducing imaging data are:

1. Removal of the 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. By default, nothing is done by the recipe, but a command line option can be used to switch on the odd-even correction on the input frames.

2. Dark subtraction

   If provided, the master dark is subtracted 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.

3. Flat Fielding

   If provided, the master flat is divided to all images. 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 we look at the flat fielded images, we can see that they are far from flat. There is a jump between the two halves of the array, and structures 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.

4. Flagging bad pixels, removing vignetted regions

   The recipe can replace bad pixels by an average of their valid neighbors if a master bad pixel map is provided.

5. 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.

   For each input object image, a sky frame is created and subtracted to the object frame. If in the input set of frame, there are some sky frames provided, the sky estimation is simply the median of those sky frames. This median is then used for all the object frames.

   In most cases, there are no sky frames taken, and the sky estimation has to be computed from the object frames. In this case, we take advantage of the fact that we observe in jitter mode, and that a bright object does not stay at the same position. This means that a given pixel only falls on an object for a minority of the input object frames. To use this, for each pixel of the detector, the sky value is the average (with rejection of outliers) of the same pixel values before and after the current frame.

   If there are to few object frames in input to use this method, a simple median of the object frames is used.

6. Registration and stacking

   To register the sky-subtracted images to a common reference, it is necessary to precisely estimate the offsets between them. `isaac_img_jitter` applies a 2D cross-correlation routine (see section 10.1.1) to determine the offsets to an accuracy of 1/10th of a pixel. An initial estimate of the offsets between frames can be found in the FITS headers. The recipe assumes that the offsets found in the input FITS headers have a given accuracy.

   Registering the images is done by resampling them with sub-pixel 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.

   Stacking the resulting images is done by averaging them with a rejection of the outliers.

7. Removal of residual bias variations

   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.

Figure 10.2.2 shows an example of a raw image and the result obtained with the *isaac_img_jitter* recipe.

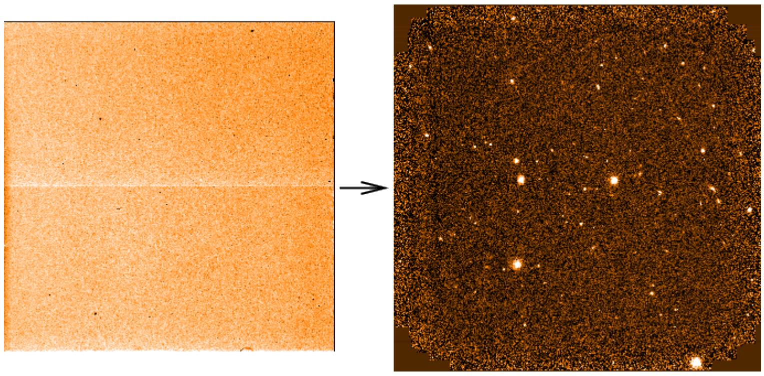

Figure 10.2.2: Jitter results in imaging

10.2.8 isaac_spc_flat

The flat is the normalised difference of two images, one with the lamp on and another with the lamp off.

This recipe first separates the input list of frames by settings. A setting is defined by the slit used, the central wavelength and the resolution. All input frames with those matching keywords are in the same setting.

In each of the settings, an even number of FITS files is expected. 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.

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 10.2.3.

The output consists of one master flat for each instrument setting.

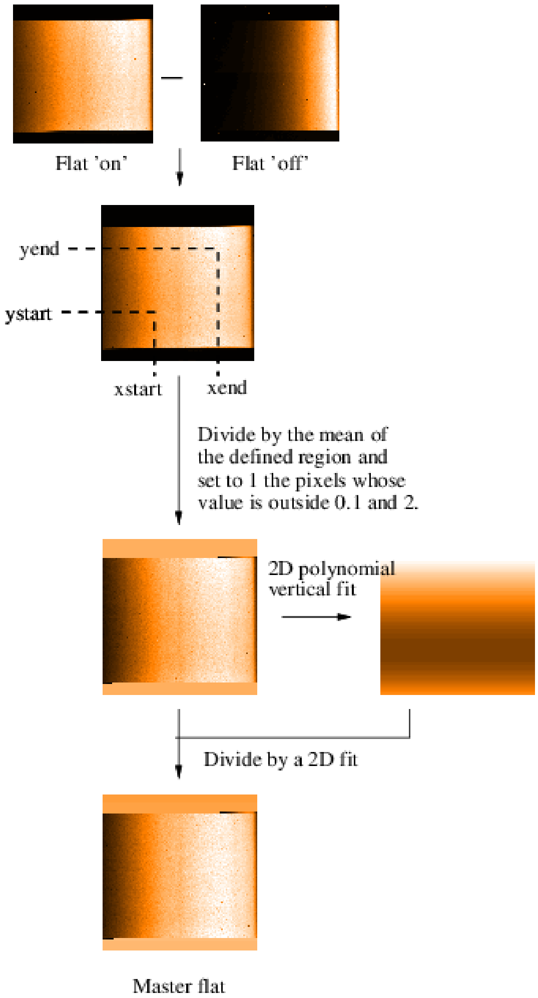

Figure 10.2.3: Flat field reduction

10.2.9 isaac_spc_arc

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 5 microns 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 polynomial (4 terms) gives a good description of the dispersion.

For a given instrument setting (defined by the slit used, the resolution and the central wavelength), 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 recipe starts by classifying images based on instrument setting.

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 is computed by matching a Xenon and/or Argon atlas with the corrected spectra.

The procedure produces a FITS table, which contains the fit to the slit curvature and the linear dispersion relation.

Figure 10.2.4 shows a raw image and the corresponding corrected one.

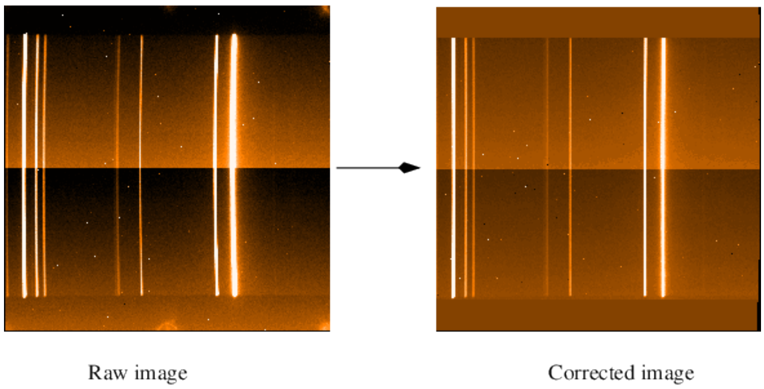

Figure 10.2.4: Slit curvature correction

10.2.10 isaac_spc_startrace

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 recipe produces several files.

1. Positions table. The location of the stellar images and spectra are found and stored in a table for LR and MR resolutions.

2. Shapes of spectra. The spectra shapes are traced with a 3rd order polynomial and the results are stored in tables for LR and MR resolutions

3. 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 tables. These tables are the ones used by *isac_spc_jitter* to correct the spectral tilt.

A flowchart of what the recipe does is given in figure 10.2.5.

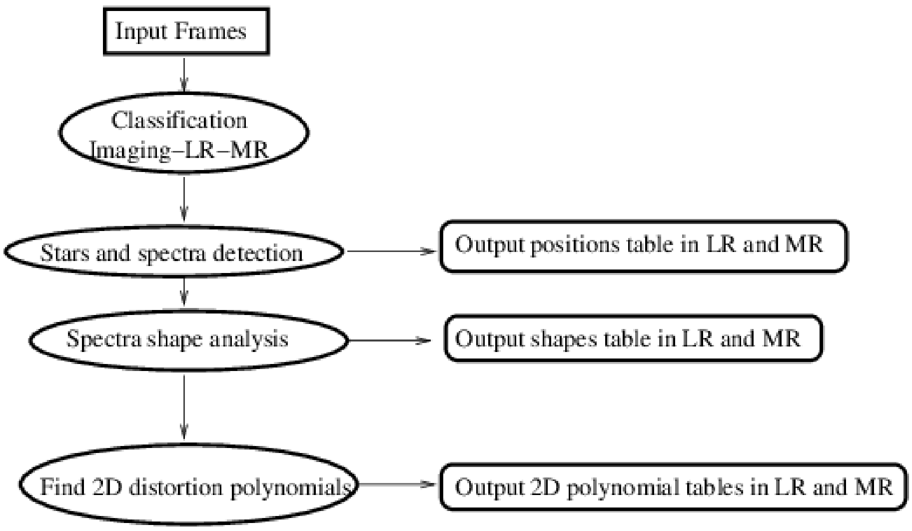

Figure 10.2.5: A flowchart of the steps in *isaac_spc_startrace*

10.2.11 isaac_spc_jitter

The recipe accepts in input different calibration files:

- A table to use to correct for slit curvature
- A table to use for wavelength calibration
- A table to use to correct for spectral tilt
- A master flat field

The recipe starts by classifying the input images according to the cumulative offsets in the headers. The frames are classified according to their nodding position along the slit.

The sequence AAA BBB BBB AAA (A and B represent the two positions along the slit) is then transformed in ABBA by averaging the As and Bs. The pairs are then subtracted to give A-B, B-A, B-A and A-B.

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.

The subtracted frames contain positive and negative spectra. The two spectra are combined. 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 10.2.6 shows the individual steps of the recipe.

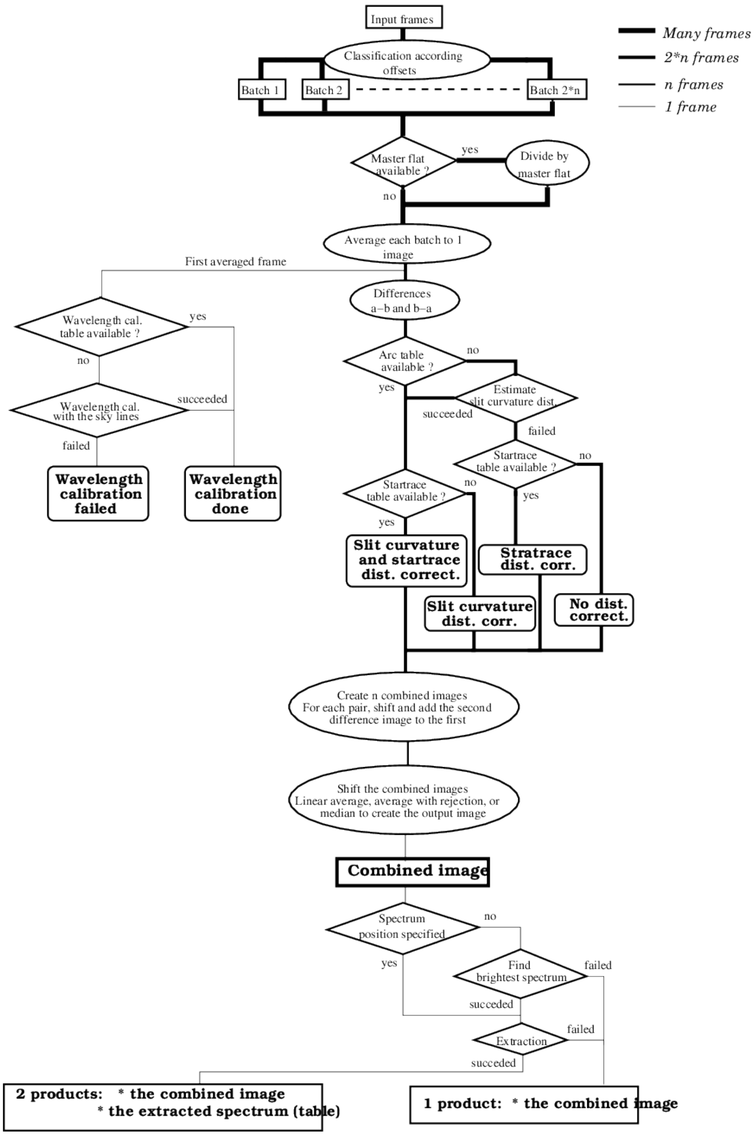

Figure 10.2.6: A flowchart of the steps in *isaac_spc_jitter*

The output consists of the reduced FITS image and the extracted spectrum, as illustrated in Figure 10.2.7.

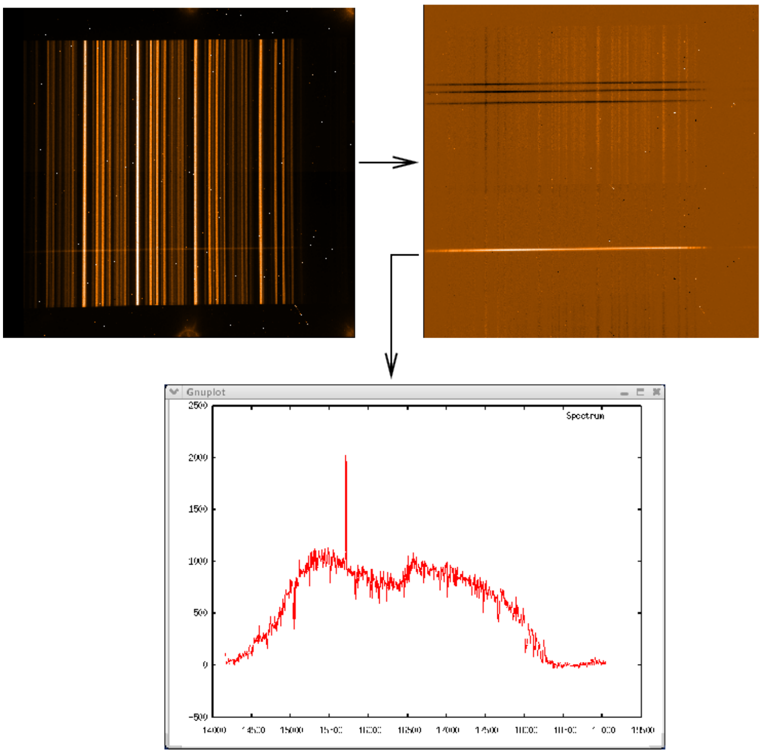

Figure 10.2.7: Jitter results in spectroscopy

In case the recipe is executed on standard star observations, the efficiency curve and the conversion factor are computed. In those cases, the spectral energy distribution and the standard stars catalog need to be provided.

The conversion is computed like this:

$$conversion(\lambda) = \frac{spec(\lambda) * gain * 10^{\frac{mag}{2.5}} * h * c}{dit * S * \delta\lambda * \lambda} \tag{2}$$

where

- λ is the wavelength in angstroms
- $spec$ is the extracted spectrum
- $gain$ is the instrument gain (4.5 for ISAAC)
- mag is the magnitude in the observed band
- $h = 6.62e - 27$
- $c = 3e18$

- dit is the integration time
- S is the primary miror surface (3.1415 * 400 * 400)
- $\delta\lambda$ is the dispersion

The efficiency is computed from the conversion:

$$efficiency(\lambda) = \frac{conversion(\lambda)}{mag_0(\lambda)} \tag{3}$$

where

mag_0 is the normalized spectral energy distribution.

The Efficiency and conversion columns are added to the produced table as shown in Figure 10.2.8.

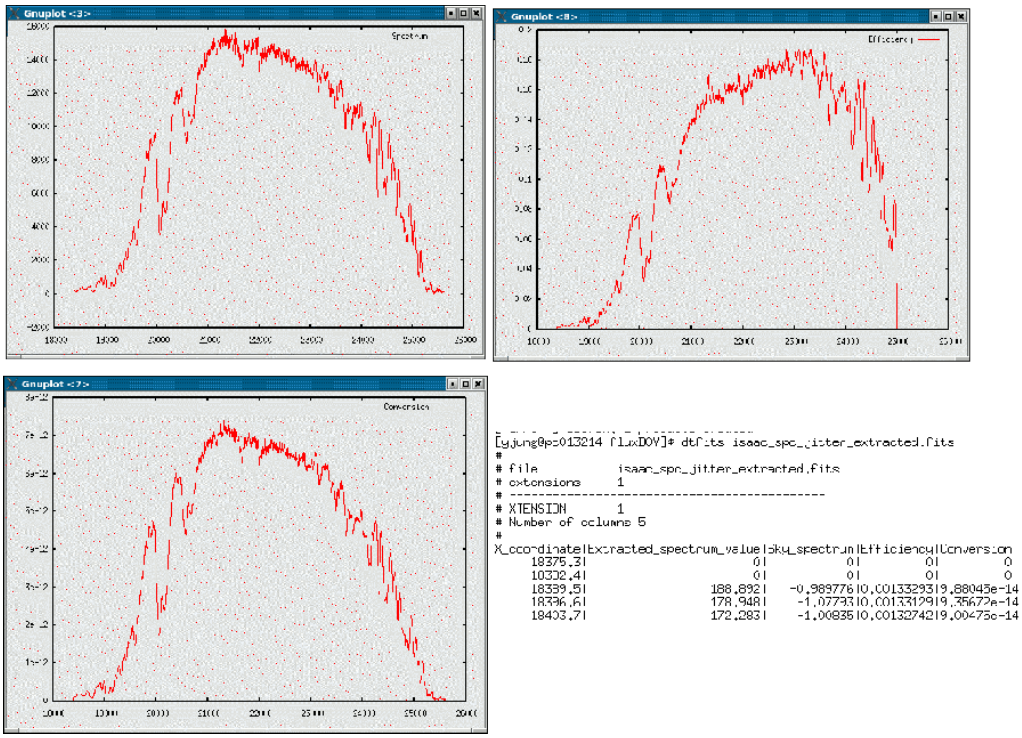

Figure 10.2.8: Example of the products in case of standard stars observations

A Installation

This chapter gives generic instructions on how to obtain, build and install the ISAAC pipeline.

The supported platforms are listed in Section A.1. It is recommended reading through Section A.2.2 before starting the installation.

A bundled version of the ISAAC pipeline with all the required tools and an installer script is available from `http://www.eso.org/pipelines/`, for users who are not familiar with the installation of software packages.

A.1 Supported platforms

The utilisation of the GNU build tools should allow to build and install the ISAAC pipeline on a variety of UNIX platforms, but it has only been verified on the VLT target platforms:

- Linux (glibc 2.1 or later),
- Sun Solaris 2.8 or later,

using the GNU C compiler (version 3.2 or newer).

A.2 Building the ISAAC pipeline

This section shows how to obtain, build and install the ISAAC pipeline from the official source distribution.

A.2.1 Requirements

To compile and install the ISAAC pipeline one needs:

- the GNU C compiler (version 3.2 or later),
- the GNU `gzip` data compression program,
- a version of the `tar` file-archiving program, and,
- the GNU `make` utility.

An installation of the Common Pipeline library (CPL) must also be available on the system. The CPL distribution can be obtained from `http://www.eso.org/cpl`.

Please note that CPL itself depends on an existing cfitsio installation.

In order to run the ISAAC pipeline recipes a front-end application is also required. Currently there are two such applications available, a command-line tool called *EsoRex* and the Java based data file organizer, *Gasgano*, which provides an intuitive graphical user interface (see Section 4.2, page 16). At least one of them must be installed. The *EsoRex* and *Gasgano* packages are available at `http://www.eso.org/cpl/esorex.html` and `http://www.eso.org/gasgano` respectively.

For installation instructions of any of the additional packages mentioned before please refer to the documentation of these packages.

A.2.2 Compiling and installing the ISAAC pipeline

The ISAAC pipeline distribution kit 6.2.4contains:

isaac-pipeline-manual-6.2.4.pdf	The ISAAC pipeline manual
install_pipeline	Install script
cfitsio3490.tar.gz	CFITSIO 3490
cpl-7.3.tar.gz	CPL 7.3
esorex-3.13.6.tar.gz	esorex 3.13.6
gasgano-2.4.8.tar.gz	GASGANO 2.4.8
isaac-6.2.4.tar.gz	ISAAC 6.2.4
isaac-calib-6.2.4.tar.gz	ISAAC calibration files 6.2.4

Here is a description of the installation procedure:

1. Change directory to where you want to retrieve the ISAAC pipeline recipes package. It can be any directory of your choice but not:

   ```
   $HOME/gasgano
   $HOME/.esorex
   ```

2. Download from the ESO ftp server, `http://www.eso.org/pipelines/`, the latest release of the ISAAC pipeline distribution.

3. Verify the checksum value of the tar file with the cksum command.

4. Unpack using the following command:

   gunzip isaac-kit-6.2.4.tar.gz

   tar -xvf isaac-kit-6.2.4.tar

5. Install: after moving to the top installation directory,

   cd isaac-kit-6.2.4

   it is possible to perform a simple installation using the available installer script (*recommended*):

   ```
   ./install_pipeline
   ```

(beware: the execution may take a few minutes on Linux and several minutes on SunOS).

By default the script will install the ISAAC recipes, *Gasgano*, *EsoRex*, all the necessary libraries, and the static calibration tables, into a directory tree rooted at `$HOME`. A different path may be specified as soon as the script is run.

The only exception to all this is the *Gasgano* tool, that will always be installed under the directory `$HOME/gasgano`. Note that the installer will move an existing `$HOME/gasgano` directory to `$HOME/gasgano` before the new *Gasgano* version is installed.

Important: the installation script would ensure that any existing *Gasgano* and *EsoRex* setup would be inherited into the newly installed configuration files (avoiding in this way any conflict with other installed instrument pipelines).

Alternatively, it is possible to perform a manual installation (*experienced users only*): the `README` file located in the top installation directory contains more detailed information about a step-by-step installation.

B Abbreviations and acronyms

ANSI	American National Standards Institute
ASCII	American Standard Code for Information Interchange
CalibDB	Calibration Database
CPL	Common Pipeline Library
DFO	Data Flow Operations department
DFS	Data Flow System department
DMD	Data Management and Operations Division
DRS	Data Reduction System
ESO	European Southern Observatory
ESOREX	ESO-Recipe Execution tool
FITS	Flexible Image Transport System
FOV	Field Of View
FPN	Fixed Patter Noise
GUI	Graphical User Interface
LW	Long Wavelength
OB	Observation Block
PSO	Paranal Science Operations
QC	Quality Control
RON	Read Out Noise
SOF	Set Of Frames
SW	Short Wavelength
UT	Unit Telescope
VLT	Very Large Telescope