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

SOFI Pipeline User Manual

VLT-MAN-ESO-19500-4284

Issue 1.2

Date 1st April 2008

	Name	Date	Signature
Prepared:	Y. Jung	1st April 2008	
Approved:	P.Ballester		
Released:	M. Peron		

No content on this page

Change record

Issue/Rev.	Date	Section/Parag. affected	Reason/Initiation/Documents/Remarks
1.0	01/04/2007	All	For SOFI 1.2.0
1.1	01/10/2007	All	For SOFI 1.3.0
1.2	01/04/2008	All	For SOFI 1.5.0

No content on this page

Contents

1 Introduction **9**

1.1 Purpose . . . 9

1.2 Acknowledgements . . . 9

1.3 Scope . . . 9

1.4 Reference and applicable documents . . . 9

2 Overview **11**

3 SOFI Instrument Description **12**

4 Quick start **14**

4.1 SOFI pipeline recipes . . . 14

4.2 An introduction to Gasgano and EsoRex . . . 14

4.2.1 Using Gasgano . . . 15

4.2.2 Using EsoRex . . . 16

5 Known Problems **20**

5.1 Wavelength calibration . . . 20

6 SOFI Data Description **21**

6.1 Imaging frames . . . 21

6.2 Spectroscopic frames . . . 23

7 Static Calibration Data **24**

7.1 Standard Star Catalogs . . . 25

7.2 Spectral lines . . . 25

8 Data Reduction **26**

8.1 Data reduction overview . . . 26

8.1.1 Bias variations (SW) . . . 26

8.1.2 Shift register glow . . . 26

8.1.3 Electrical ghosts . . . 27

8.1.4 Cross Talk effect . . . 27
8.1.5 Detector non-linearity . . . 27
8.2 Required input data . . . 27
8.3 Reduction Cascade . . . 27

9 Pipeline Recipe Interfaces **29**
9.1 sofi_img_dark . . . 29
9.1.1 Input . . . 29
9.1.2 Output . . . 29
9.1.3 Quality control . . . 29
9.1.4 Parameters . . . 30
9.2 sofi_img_domeflat . . . 30
9.2.1 Input . . . 30
9.2.2 Output . . . 30
9.2.3 Quality control . . . 30
9.2.4 Parameters . . . 30
9.3 sofi_img_detlin . . . 30
9.3.1 Input . . . 30
9.3.2 Output . . . 31
9.3.3 Quality control . . . 31
9.3.4 Parameters . . . 31
9.4 sofi_img_illum . . . 31
9.4.1 Input . . . 31
9.4.2 Output . . . 31
9.4.3 Quality control . . . 32
9.4.4 Parameters . . . 32
9.5 sofi_img_zpoint . . . 33
9.5.1 Input . . . 33
9.5.2 Output . . . 33
9.5.3 Quality control . . . 34
9.5.4 Parameters . . . 34

9.6 sofi_img_jitter . . . 35
9.6.1 Input . . . 35
9.6.2 Output . . . 35
9.6.3 Quality control . . . 35
9.6.4 Parameters . . . 36
9.7 sofi_spc_flat . . . 36
9.7.1 Input . . . 36
9.7.2 Output . . . 37
9.7.3 Quality control . . . 37
9.7.4 Parameters . . . 37
9.8 sofi_spc_arc . . . 37
9.8.1 Input . . . 37
9.8.2 Output . . . 37
9.8.3 Quality control . . . 38
9.8.4 Parameters . . . 39
9.9 sofi_spc_jitter . . . 39
9.9.1 Input . . . 39
9.9.2 Output . . . 39
9.9.3 Quality control . . . 40
9.9.4 Parameters . . . 41

10 Algorithms **42**
10.1 General Algorithms . . . 42
10.1.1 Cross-correlation . . . 42
10.1.2 Distortion computation . . . 42
10.1.3 Wavelength calibration . . . 43
10.2 Recipes Algorithms . . . 44
10.2.1 sofi_img_dark . . . 44
10.2.2 sofi_img_domeflat . . . 45
10.2.3 sofi_img_illum . . . 45
10.2.4 sofi_img_detlin . . . 46

10.2.5 sofi_img_zpoint 47
10.2.6 sofi_img_jitter 48
10.2.7 sofi_spc_flat 50
10.2.8 sofi_spc_arc 52
10.2.9 sofi_spc_jitter 53

A Installation **56**
A.1 Supported platforms 56
A.2 Building the SOFI pipeline 56
A.2.1 Requirements 56
A.2.2 Compiling and installing the SOFI pipeline 57

B Abbreviations and acronyms **59**

1 Introduction

1.1 Purpose

The SOFI pipeline is a subsystem of the *VLT Data Flow System* (DFS). It is used in La Silla 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 SOFI 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 SOFI data reduction sequence with the SOFI pipeline.

This manual is a complete description of the data reduction recipes implemented by the the SOFI pipeline, reflecting the status of the SOFI pipeline version 1.5.0.

1.2 Acknowledgements

The SOFI pipeline has been adapted in 2006 from the ISAAC pipeline. Valentin Ivanov, instrument scientist of SOFI at that time and John Pritchard have provided valuable feed-back on both the pipeline itself and its documentation, and made it possible to release it publicly.

1.3 Scope

Updated versions of the present document may be found on [13]. For general information about the current instrument pipelines status we remind the user of [3]. Quality control information are at [2].

Additional information on the Common Pipeline Library (CPL) and ESOREX can be found respectively at [8], [11]. The Gasgano tool is described in [12]. A description of the instrument is in [5]. The SOFI instrument user manual is in [6].

1.4 Reference and applicable documents

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

[2] ESO/DMO/DFO, *http://www.eso.org/observing/dfo/quality/*. *ESO-Data Flow Operation home page*. 9

[3] ESO/DMO/DFO, *http://www.eso.org/sci/software/pipelines/sofi/sofi-pipe-recipes.html*. *SOFI Pipeline Current Status*. 9

[4] ESO/INS, *http://www.eso.org/sci/facilities/lasilla/instruments/sofi/doc/Calibr_plan.html*. *SofI - Calibration Plan*.

[5] ESO/INS, *http://www.eso.org/sci/facilities/lasilla/instruments/sofi.html*. *SOFI instrument home page*. 9

[6] ESO/INS, *http://www.eso.org/sci/facilities/lasilla/instruments/sofi/doc/manual.html*. *SOFI User Manual*. 9

[7] ESO/SDD/DFS, *http://www.eso.org/observing/cpl/download.html*. *Common Pipeline Library User Manual*. VLT-MAN-ESO-19500-2720.

[8] ESO/SDD/DFS, *http://www.eso.org/cpl/*. *CPL home page*. 9

[9] ESO/SDD/DFS. *Deliverables Specification*. VLT-SPE-ESO-19000-1618 (2.0).

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

[11] ESO/SDD/DFS, *http://www.eso.org/cpl/esorex.html*. *ESOREX home page*. 9, 17, 18

[12] ESO/SDD/DFS, *http://www.eso.org/gasgano/*. *Gasgano User's Manual*. VLT-PRO-ESO-19000-1932. 9, 11, 15, 21

[13] ESO/SDD/DFS, *http://www.eso.org/pipelines*. *SOFI Pipeline Web Page*. 9

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 that 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* [12] and *EsoRex*, both included in the pipeline distribution (see Appendix A). 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 Section 4).

The SOFI instrument and the different types of SOFI 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 SOFI pipeline recipes is described and in Appendix B a list of used abbreviations and acronyms is given.

3 SOFI Instrument Description

SOFI or **S**on **OF** **I**SAAC is the infrared spectrograph and imaging camera on the NTT. In many ways, it resembles its "parent" ISAAC, and the EFOSCII on the 3.6m. Both are focal reducing instruments capable of imaging, spectroscopy and polarimetry.

SOFI offers the following observing modes.

- imaging with plate scales of 0.144, 0.144, 0.273 and 0.288 arc second per pixel in the following modes: Small Field, Large Field + Focal Elongator, Spectroscopic Field and Large Field, respectively; broad and narrow band filters in the wavelength range from 0.9 to 2.5 microns are available.
- low resolution R$\sim$600 (varies across the wavelength range), 0.95-2.52 micron spectroscopy with fixed width slits of 0.6, 1 and 2 arc seconds, and slit length of 4.92 arcmin.
- medium resolution R$\sim$1500 (varies across the wavelength range) H and K-band spectroscopy, with fixed width slits of 0.6, 1 and 2 arc seconds, and slit length of 4.92 arcmin.
- 0.9-2.5 micron imaging polarimetry with the large field objective (0.288 arcsec per pixel) and the set of filters available in imaging mode.

Figure 3.0.1 shows a picture of the instrument mounted on the NTT.

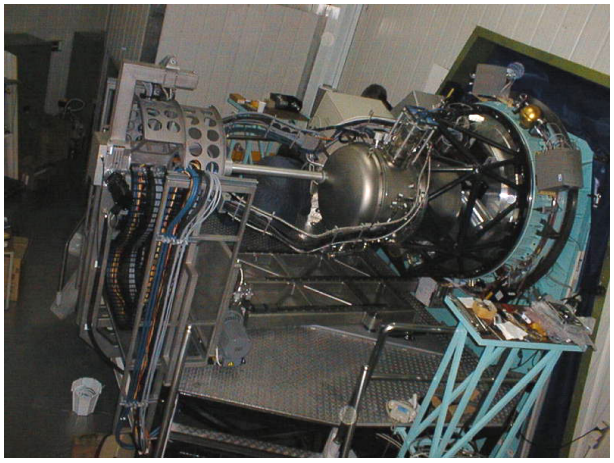

Figure 3.0.1: *Picture of SOFI.*

Figure 3.0.2 shows the optical of SOFI.

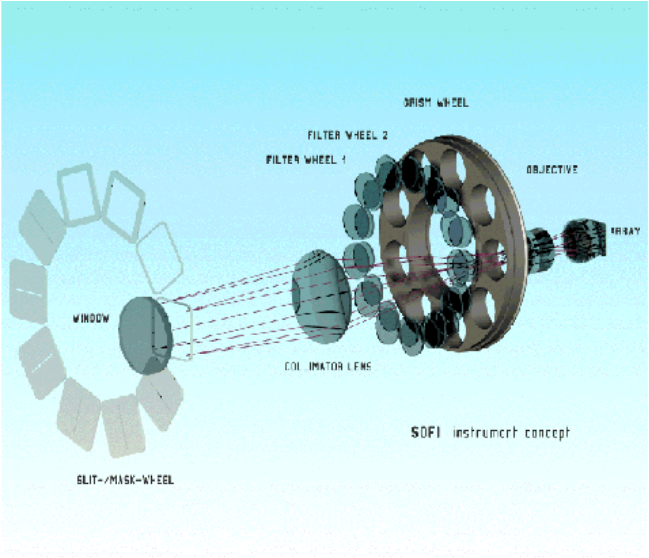

Figure 3.0.2: *Optical path of SOFI.*

4 Quick start

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

4.1 SOFI pipeline recipes

The current SOFI pipeline is based on a set of 9 stand-alone recipes involved in the data reduction cascade.

The 6 recipes in imaging mode are:

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

sofi_img_domeflat: Creates a master flat field frame and a bad pixels map.

sofi_img_detlin: Measures the non-linearity of the detector.

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

sofi_img_zpoint: Zero point measurements using standard star observations.

sofi_img_jitter: Main reconstruction routine, including cross-talk effect correction, dark correction, flatfield calibration, bad pixels cleaning, and images correlation and recombination.

The 3 recipes in spectroscopic mode are:

sofi_spc_flat: Creates a master flat field.

sofi_spc_arc: Uses Xenon or Neon lamps to calibrate both the wavelength and the slit curvature distortion.

sofi_spc_jitter: Main observation recipe, that corrects the cross-talk effect, the flat field, the distortion, reconstructs the combined image, detects, extracts and calibrates 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 the observatory, 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 SOFI 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 SOFI 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* [12] for a more complete description of the *Gasgano* interface.

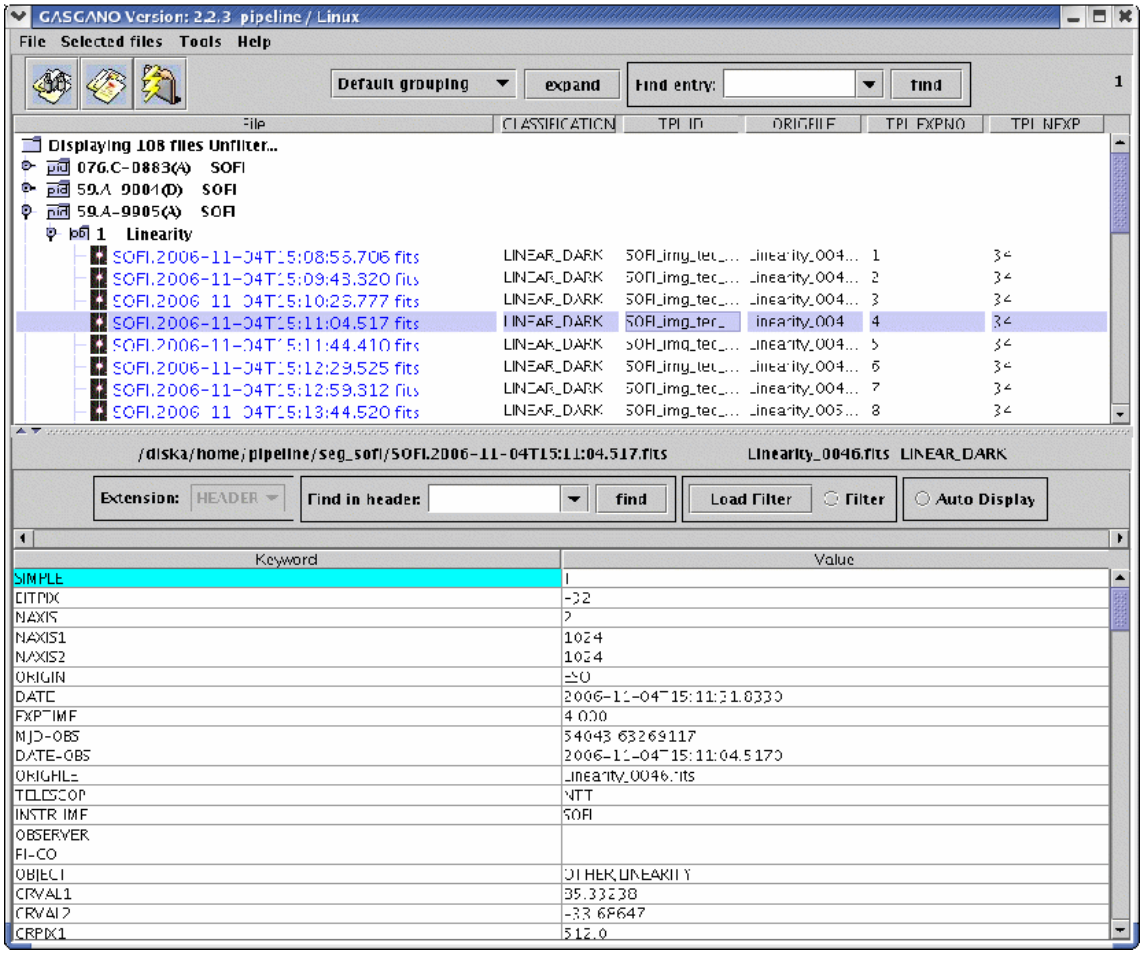

Figure 4.2.1: *The Gasgano main window.*

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). 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 *sofi_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 19).

```
/file_path/SOFI.2004-08-14T10:20:56.497.fits    IM_JITTEROBJ
/file_path/SOFI.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 SOFI 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 SOFI recipes are just the DRS component of the complete pipeline running at the observatory, 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).

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 [11]) 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 *sofi_img_jitter* has its *EsoRex* generated configuration file named `sofi_img_jitter.rc`, and is generated with the command:

esorex - -create-config sofi_img_jitter

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

```
# --xcorr
# Cross correlation search and measure sizes.
sofi.sofi_img_jitter.xcorr=40,40,65,65
```

[2]The *EsoRex* recipe configuration file corresponds to the *Parameters* panel of the *Gasgano* recipe execution window (see Figure 4.2.2, page 19).

In this example, the parameter `sofi.sofi_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 SOFI 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 *sofi_img_jitter* for processing the files specified in the set-of-frames `sofi_img_jitter.sof`:

esorex sofi_img_jitter sofi_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 *sofi_img_jitter* recipe *xcorr* parameter to `20,20,65,65`, the following should be typed:

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

For more information on *EsoRex*, see [11].

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

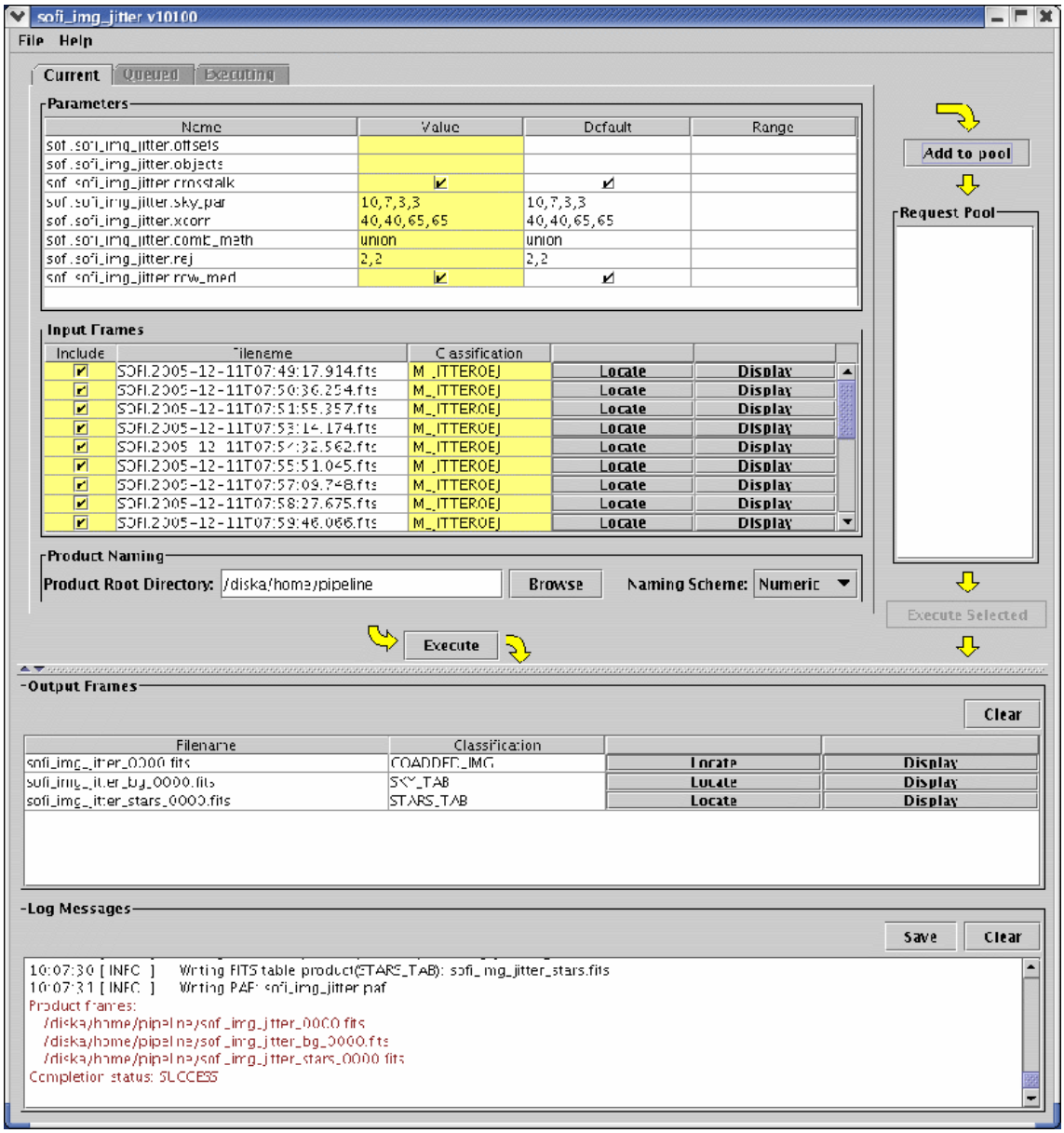

Figure 4.2.2: *The Gasgano recipe execution window.*

5 Known Problems

There are some features on the SOFI 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

The wavelength calibration in the two modes *LONG_SLIT_Z* and *LONG_SLIT_J* is still failing due to the inconsistency of the used catalogs with regard to the observed data in these wavelengths. The catalog need to be upgraded to fix this problem. The mixture of different orders is also a problem whose impact still needs to be evaluated.

6 SOFI Data Description

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

SOFI data do not contain extensions, the images are always contained in the primary FITS data unit. In spectroscopy, the slit is horizontal, and the dispersion increases from the top to the bottom of the image.

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* [12], 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 SOFI 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 SOFI 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 16).

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: `sofi_img_detlin`

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

  Association keywords:

- **Dark image for non-linearity calibration:**

  DO category: `LINEAR_DARK`
  Processed by: `sofi_img_detlin`

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

  Association keywords:

- **Image for detector illumination calibration:**

  DO category: `IM_ILLUM`
  Processed by: `sofi_img_illum`

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

Association keywords:

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

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

  Association keywords:

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

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

  Association keywords:

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

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

  Association keywords:

- **Jitter object observation:**
  DO category: `IM_JITTEROBJ`
  Processed by: `sofi_img_jitter`

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

  Association keywords:

- **Jitter sky observation:**
  DO category: `IM_JITTERSKY`
  Processed by: `sofi_img_jitter`

  Classification keywords:
  `DPR CATG = OTHER or SCIENCE`
  `DPR TYPE = OTHER or SKY`
  `DPR TECH = IMAGE`

  Association keywords:

6.2 Spectroscopic frames

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

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

  Association keywords:

- **Spectroscopic flat field:**
  DO category: `SP_FLAT`
  Processed by: `sofi_spc_flat`

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

  Association keywords:

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

  Classification keywords:
  `DPR CATG = SCIENCE`
  `DPR TYPE = OTHER`
  `DPR TECH = SPECTRUM`

  Association keywords:

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

  Classification keywords:
  `DPR CATG = OTHER or SCIENCE`
  `DPR TYPE = OTHER or OBJECT`
  `DPR TECH = SPECTRUM`

  Association keywords:

7 Static Calibration Data

For SOFI, the static calibration data are the standard stars catalogs, the OH, Xenon and Neon spectroscopic lines.

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

```
ls sofip/catalogs/*

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

sofip/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:

- sofi_util_stdstars : Standard stars catalog creation
- sofi_util_genlines : Generate spectrum calibration FITS tables

For example, the following:

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

$ esorex sofi_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 `sofi_img_zpoint` recipe to get the magnitude of the observed standard star to compute the Zero Point.

Stars are currently taken from the following catalogs:

- Arnica (41 entries)
- ESO Van der Bliek (264 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 SOFI pipeline also needs the emission lines of the atmosphere, of the Neon 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 Neon for calibration lamps).

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.

The data reduction is strongly connected with the observation strategy which in turn depends on the nature of the observed object and the science goals of the observations. The Pipeline only treats the most general cases and the user is stronly recommended to read the SofI user manual for detailed discussion of the observing strategies and their optimisations.

8.1 Data reduction overview

In 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. Jittering may be combined with offsets, to acquire a sequence of alternating images of the object and "clear" nearby sky fields.

In 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. The nodding is usually combined with small random jitters along the slit, around each nodding position.

8.1.1 Bias variations (SW)

The SOFI infrared detector bias is a function of the detector integration time (DIT) and the total detector illumination.

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

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

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. It is nevertheless also visible on deep images.

8.1.4 Cross Talk effect

A bright source imaged on the array produces a "ghost" that affects all the lines where the source is and all the corresponding lines in the other half of the detector. For instance, if the bright source is on row 300, the cross-talk affect the row $300+512=812$ and vice versa, if the source is on row 750, the other affected row will be $750-512=238$. The cross talk is seen either on high S/N data or if the bright source reaches near saturation level.

Though the effect is not completely understood it is well described and can be easily corrected. The intensity of the ghost is in fact 1.4×10^{-5} times the integrated flux of the line.

8.1.5 Detector non-linearity

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

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

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

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

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.

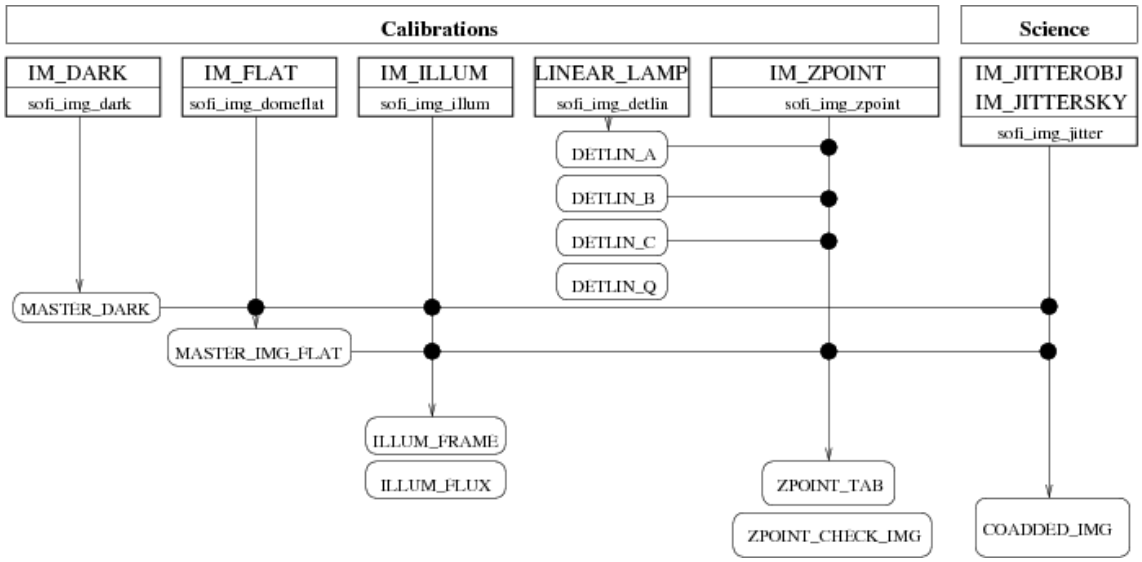

Figure 8.3.1: *SOFI Association Map in imaging mode.*

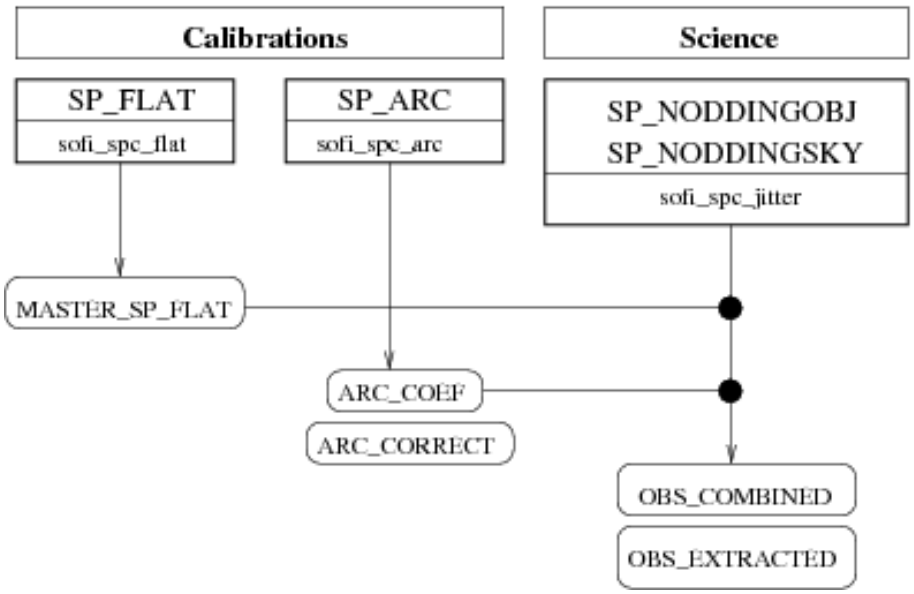

Figure 8.3.2: *SOFI Association Map in spectroscopic mode.*

9 Pipeline Recipe Interfaces

In this section we provide for each recipe examples of the required input data.

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 sofi_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 sofi_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 LL RONi: Read Out Noise value computed on the lower left quadrant on pair number i
- QC LR RONi: Read Out Noise value computed on the lower right quadrant on pair number i
- QC UL RONi: Read Out Noise value computed on the upper left quadrant on pair number i
- QC UR RONi: Read Out Noise value computed on the upper right quadrant on pair number i

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)

9.2 sofi_img_domeflat

This recipe creates a master flat field.

9.2.1 Input

This recipe expects input frames classified as IM_FLAT.

9.2.2 Output

The following product is created:

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

9.2.3 Quality control

No QC produced by this recipe.

9.2.4 Parameters

No input paramaters.

9.3 sofi_img_detlin

This recipe computes the detector non-linearity.

9.3.1 Input

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

9.3.2 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 sofi_img_detlin_A.fits, sofi_img_detlin_B.fits and sofi_img_detlin_C.fits

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

9.3.3 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

9.3.4 Parameters

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

9.4 sofi_img_illum

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

9.4.1 Input

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

9.4.2 Output

The two products are the image that gives the polynomial illumination on the whole detector named sofi_img_illum.fits (PRO CATG = ILLUM_FRAME) and a table named sofi_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 9.4.1.

```
# ----------------------------------------
# XTENSION       1
# Number of columns 3
#
      POSX|      POSY|      FLUX
       512|       530|   30961.6
       129|       146|   19073.6
       335|       148|   20640.2
       640|       148|     21213
       835|       148|   16512.2
       834|       403|     15626
       639|       403|   17055.6
       335|       403|   16164.4
       129|       402|   21223.8
       129|       657|   20937.6
       334|       657|     15579
       639|       657|   17635.6
       834|       658|   12463.2
       834|       914|   12702.2
       639|       913|   14512.6
       334|       912|   16672.6
       127|       914|   17953.8
```

Figure 9.4.1: *Table created by the illumination recipe.*

9.4.3 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

9.4.4 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)

9.5 sofi_img_zpoint

This recipe computes the Zero Point.

9.5.1 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. Those frames are tagged with IM_ZPOINT. 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).

9.5.2 Output

Two files are produced. The first is a table named sofi_img_zpoint.fits containing for each standard star the photometry computed as shown in Figure 9.5.1 (PRO CATG = ZPOINT_TAB).

```
# ------------------------------------------
# XTENSION      1
# Number of columns 8
#
     POSX|     POSY|   ZPOINT|     FLUX|     PEAK|      BGD|    FWHMX|    FWHMY
  470.195|  536.463|  24.8157|  16939.6|  380.167|  30.6667|  4.66767|  5.66409
  774.541|  841.468|  24.3906|    11451|     316|     -33|  3.25543|  4.76714
  774.577|  841.338|  24.5074|  12751.6|  294.667| -42.6666|  3.25642|  4.75382
  154.349|  841.806|  24.7252|  15584.9|  392.833|  46.3334|  3.47319|  5.11441
  154.526|  841.794|  24.7316|  15675.9|  342.833|  8.33325|  3.31606|  5.47442
  155.163|  231.566|   24.761|  16107.3|     550|      -7|  3.57897|  4.63121
  155.101|  231.454|  24.6282|  14253.1|   539.5| -9.66663|  3.33976|  4.54491
  774.399|  231.994|  24.6359|  14353.8|  378.667|  9.66663|  4.57996|  6.45542
```

Figure 9.5.1: *Example of the output produced by sofi_img_zpoint.*

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

9.5.3 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 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

9.5.4 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)

9.6 sofi_img_jitter

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

9.6.1 Input

The recipe expects frames marked with IM_JITTEROBJ and IM_JITTERSKY (whether they are objects or sky frames).

It also accepts flat field (MASTER_IMG_FLAT) and/or dark frames (MASTER_DARK).

9.6.2 Output

The produced image is the combined image named sofi_img_jitter.fits (PRO CATG = COADDED_IMG). Besides, a table named sofi_img_jitter_bg.fits with the background values (PRO CATG = SKY_TAB) and a table named sofi_img_jitter_stars.fits (PRO CATG = STARS_TAB) with the detected stars and there photometry are produced (see Figure 9.6.1).

```
# ------------------------------------------
# XTENSION       1
# Number of columns 1
#
    SKY_BG
   5877.27
   5918.32
   5921.37
   5935.77
   5991.52
    6027.7
   6037.29
   6009.39
   5968.26
   5973.37
```

```
# ------------------------------------------
# XTENSION       1
# Number of columns 5
#
     POS_X|     POS_Y|    FWHM_X|    FWHM_Y|      FLUX
   966.229|   29.3925|   4.15001|   4.34206|   7034E.9
   895.691|   19.3489|   2.16832|   2.53739|   1423.17
   701.366|   28.5227|   5.74418|   7.50237|   733.248
   951.035|   32.0196|   1.99187|   1.84194|   229.722
   418.532|   149.388|   2.02365|   2.52913|   2139.38
   853.567|   180.459|   2.16345|   2.15424|   1176.63
   1131.46|    181.21|   2.06202|   4.27282|   895.869
    899.77|   197.164|   2.39389|   2.69425|   1526.55
   908.008|   198.968|   2.27176|   2.44846|   342.383
    824.77|   248.541|   2.60779|   2.27524|   1749.04
   261.036|   253.924|   4.12744|   2.70624|   267.574
   827.847|   275.022|   3.53761|   2.94732|   731.066
   482.626|   281.944|   3.57243|   3.17498|   384.599
      1143|   293.251|   2.54608|   2.25019|   532.816
    352.84|   326.215|   2.41201|   2.41321|   2534.67
   395.368|   342.226|   4.53454|   3.80684|   10062.4
   297.972|   334.979|   5.06531|    3.2535|   218.463
   462.586|   337.361|   2.34536|   3.04701|   1444.01
   872.084|   383.485|   3.53319|   2.06865|   811.721
   636.038|   392.954|    2.5139|   2.35479|    302.91
   641.174|   397.042|   2.86443|   2.57792|   314.505
   150.542|   425.336|   3.57191|   5.39918|   398.571
   112.939|   429.107|   3.40157|   2.36389|   603.473
```

Figure 9.6.1: *Products of sofi_img_jitter.*

9.6.3 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
- 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

9.6.4 Parameters

- off: The offsets text file (default is NULL)
- objs: The correlation object(s) position(s) (default is NULL)
- crosstalk: Flag to activate the crosstalk effect correction (default is TRUE)
- 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)

9.7 sofi_spc_flat

This recipe generates the spectroscopic flat field.

9.7.1 Input

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

9.7.2 Output

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

9.7.3 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

9.7.4 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: Offset to apply before the fitting (default is 40)

9.8 sofi_spc_arc

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

9.8.1 Input

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

9.8.2 Output

For each setting, the produced tables are named sofi_spc_arc_setxx_frameyy.fits. It contains both the distortion polynomial and the dispersion relation as shown in Figure 9.8.1 (PRO CATG = ARC_COEF).

Additionally, the image with the distortion corrected arcs is produced under the name sofi_spc_arc_setxx_frameyy_corr.fits (PRO CATG = ARC_CORRECT).

```
# ------------------------------------------
# XTENSION       1
# Number of columns 4
#
Degree_of_x|Degree_of_y|poly2d_coef|WL_coefficients
         0|         0|    -6.82641|        14082,9
         1|         0|   0,995758|        4,78237
         0|         1| 0.002397261|   -4,60157e-06
         1|         1|-9.70806e-05|   -7,51516e-09
         2|         0|2.25572e-05|              0
         0|         2| 3.3765e-06|              0
```

Figure 9.8.1: *Example of the output produced by sofi_spc_arc.*

9.8.3 Quality control

The dispersion relation is the following: $Wavelength(pix) = a + b.pix + c.pix^2 + d.pix^3 + e.pix^4$

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 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 DISPCO5: The factor e 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

9.8.4 Parameters

- rej: Left and Right zones of the image to reject (default is 100,100)
- subdark: Flag to apply an automatic dark subtraction (default is FALSE)
- arc_max_w: Arc maximum width in pixels (default is 33)
- out_corr: Flag to produce distortion corrected images (default is FALSE)
- display: Flag to activate plotting facility (only works if gnuplot is installed) (default is FALSE)
- degree: Requested degree for the wavelength calibration polynomial (default is 2)
- wl_nsamples: Number of samples for the best wavelength dispersion polynomial search (default is 100)
- wl_err: Wavelength search size in Angstroms for the best wavelength dispersion polynomial search (default is 1000.0)
- ppm: Flag to activate the Point Pattern Matching (default is FALSE)

9.9 sofi_spc_jitter

This recipe is the science data reduction recipe. It produces the combined image and extracts the spectrum.

9.9.1 Input

The expected frames must be tagged with SP_NODDINGOBJ and SP_NODDINGSKY. The recipe also needs the OH lines FITS catalogs (CALPRO_OH_CATALOG).

The recipe also accepts calibration files like an arc file (ARC_COEF) and/or a flat field (MASTER_SP_FLAT).

9.9.2 Output

The produced image is the combined image. It is named sofi_spc_jitter_combined.fits (PRO CATG = OBS_COMBINED).

The extracted spectrum is stored in a table named sofi_spc_jitter_extracted.fits as shown in Figure 9.9.1 (PRO CATG = OBS_EXTRACTED).

```
# ------------------------------------------
# XTENSION       1
# Number of columns 3
#
X_coordinate|Extracted_spectrum_value|Sky_spectrum
     18428.9|                      0|          0
       18436|                      0|          0
     18443.2|                34.9542|   0.121982
     18450.3|                24.5074|   0.262553
     18457.4|                27.6124|  -0.347931
     18464.6|                14.5368|   0.176988
     18471.7|                19.3864|  -0.431524
     18478.8|                10.1018|  -0.149821
       18486|                  11.36|   0.404973
     18493.1|                17.3553|  -0.0757238
     18500.3|                22.8754|  -0.163189
     18507.4|                16.7424|   0.182963
     18514.5|                23.9685|  -0.105734
     18521.7|                20.6796|   0.051249
     18528.8|                16.9328|  0.0112508
       18536|                 12.964|   0.201385
     18543.1|                23.6014|   -0.32384
     18550.2|                16.3745|   0.258388
        |                      |          |
        |                      |          |
        |                      |          |
        |                      |          |
```

Figure 9.9.1: *Table produced by sofi_spc_jitter.*

9.9.3 Quality control

The computed dispersion relation is the following: $Wavelength(pix) = a + b.pix + c.pix^2 + d.pix^3 + e.pix^4$

- QC FILTER OBS: The filter used
- 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 DISPCO5: The e 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

9.9.4 Parameters

- crosstalk: Flag to activate the crosstalk effect correction (default is TRUE)
- wavecal: Wavelength calibration method (default is sky)
- wavecal_ppm: Flag to activate the Point Pattern Matching (default is FALSE)
- wavecal_degree: Requested degree for the wavelength calibration polynomial (default is 2)
- wavecal_nsamples: Number of samples for the best wavelength dispersion polynomial search (default is 100)
- wavecal_err: Wavelength search size in Angstroms for the best wavelength dispersion polynomial search (default is 1000.0)
- 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_le_width: Sky zone width left to the spectrum (default is 10)
- sky_ri_width: Sky zone width right to the spectrum (default is 10)
- sky_le_dist: Sky zone distance to the spectrum (default is -1)
- sky_ri_dist: Sky zone distance to the spectrum (default is -1)
- display: Flag to activate the plotting mode (default is FALSE)

10 Algorithms

In this section the data reduction procedures applied by the 9 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.

The cross-correlation is used in imaging mode to find the exact shift between two images containing the same stars field.

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 squared differences 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

The slit curvature (horizontal lines) is 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:

$Px(X, Y) = Y$

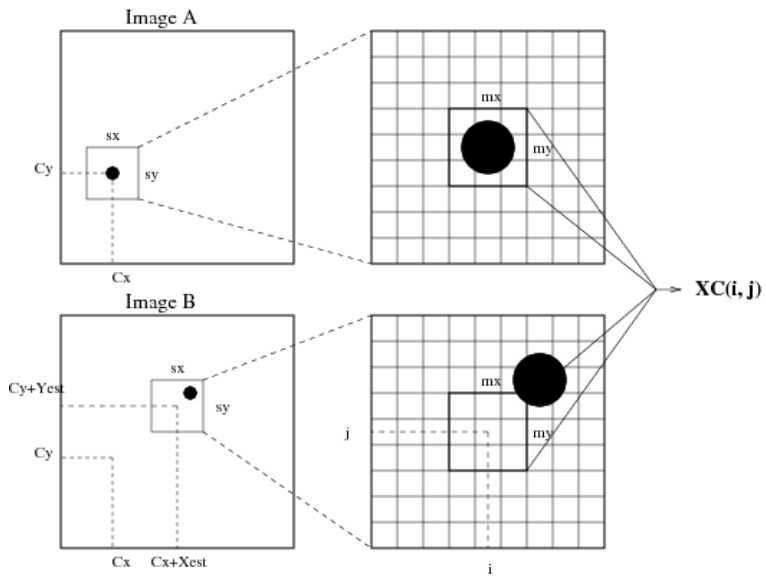

Figure 10.1.1: *Cross-Correlation algorithm.*

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

with a=6.9587, b=-0.035183, c=0.9993589213, d=1.50739833e-06, e=3.335438357e-05, f=-2.512360788e-06.

10.1.3 Wavelength calibration

The wavelength calibration is either computed using some calibration data obtained with Neon 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.
For each setting (identified from the header), we have a first guess of the first and the last pixels wavelengths (first degree polynomial). Around this first guess, we consider $nsamples$ points and use them to define a number ($nsamples^{ndegree+1}$) of candidate polynomials as shown Figure 10.1.3.

For each of these candidate polynomials, the extracted spectrum is converted from pixels to wavelength and directly compared to the catalog. A likelihood factor is computed and associated to the candidate polynomial (Figure 10.1.4 gives an example of computed likelihood factors).

The maximum defines the winner, the polynomial that is the closest to the real solution.

On the computation efficiency point of view, specifying a big number of samples together with a high degree for the polynomial explodes the number of candidate polynomials to evaluate. On the other hand if the first guess is not very precise, it is necessary to use a big wavelength error. Therefore, lots of samples are needed to avoid the real solution to be far away from the best polynomial.

For those resons, we proceed iteratively. A first pass is used with a big wavelength error, a big number of samples (100), but a first degree polynomial to refine the linear first guess (10000 evaluations).

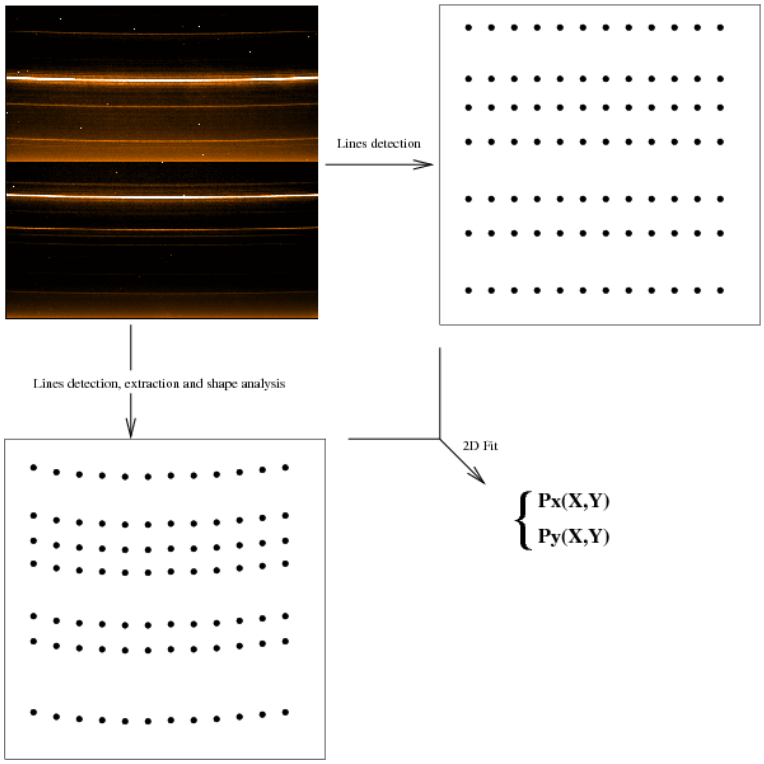

Figure 10.1.2: *Distortion estimation algorithm.*

The result is used as first guess in the second pass, where the number of samples is divided by 10 and the requested degree is 2 (1000 evaluations).

Again, the result is used as first guess in the third pass where the wavelength error is divided by 10 but the degree of the requested polynomial (2) and the number of samples (10) remain the same.

This way, we iteratively converge to the searched solution.

10.2 Recipes Algorithms

10.2.1 sofi_img_dark

Dark frames are exposures without detector illumination. The dark current of the SOFI detector is small, so the dominant feature in these frames is the detector bias.

The darks are reduced with the `sofi_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

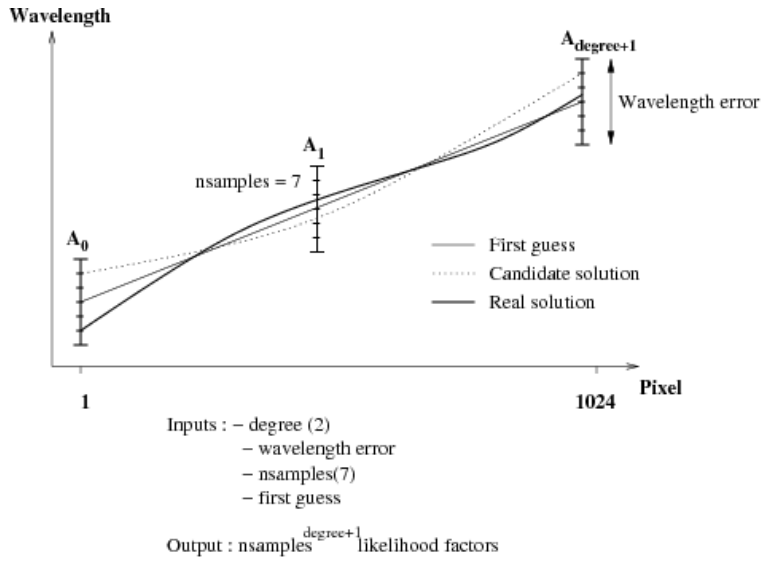

Figure 10.1.3: *Wavelength calibration algorithm.*

of the detector (4 measurements).

In each setting, for each pair, the second frame is subtracted from the first. The following measurement is applied to the 4 quadrants:

- Generate 100 13x13 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 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 sofi_img_domeflat

The flatfield in imaging is computed using a set of 4 'on' and 4 'off' images. Those 8 images are recombined following the algorithm described in Figure 10.2.1.

10.2.3 sofi_img_illum

The illumination correction takes into account low-frequency differences between the true flat and the dome flat. These variations have an RMS of 2%. If this is accurate enough, then an illumination correction is not needed.

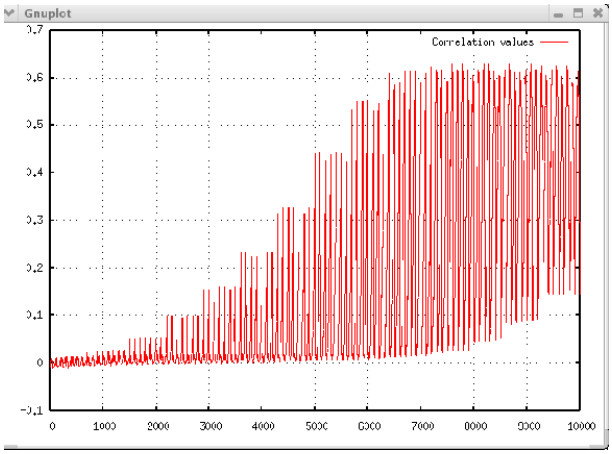

Figure 10.1.4: *Likelihood factors of the candidates* ($nsamples = 10$ *and* $degree = 3$*).*

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) can be passed as well.

The implemented algorithm is the following:

- The calibrations (flat field, dark) 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 sofi_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.

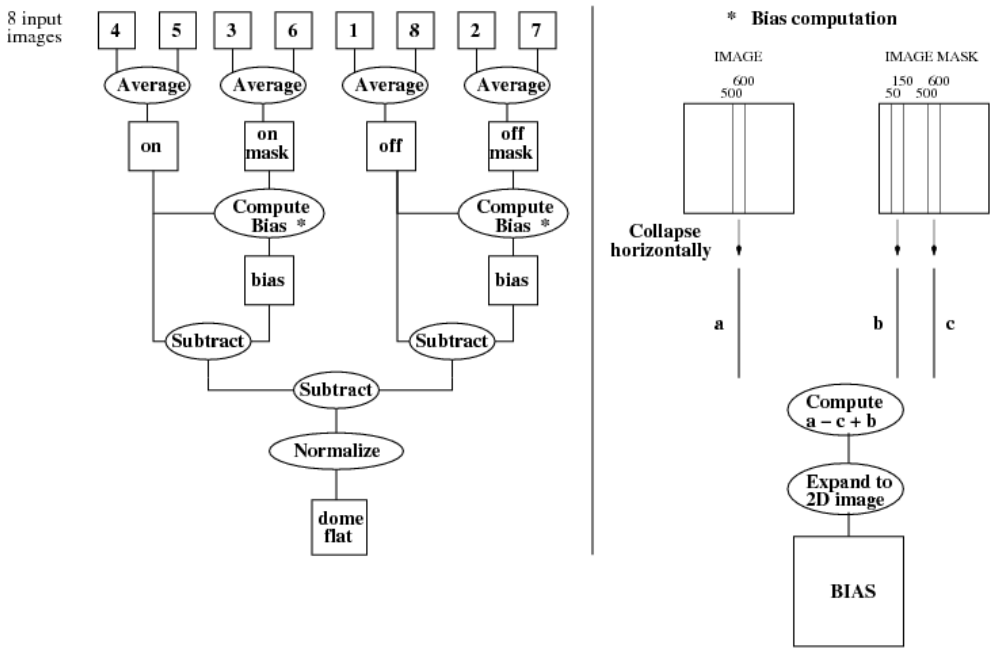

Figure 10.2.1: *Dome flat algorithm.*

- 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 sofi_img_zpoint

Standard stars are observed every night in different filters.

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 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: 1-2, 2-1, 2-3, 3-2, 3-4, 4-3, 4-5, 5-4).

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`.

10.2.6 sofi_img_jitter

The basic steps in reducing imaging data are:

1. Removal of the crosstalk effect

   The crosstalk effect implies a difference of level between the lower part and the upper part of the detector. This difference is removed by the crosstalk correction algorithm described Figure 10.2.2.

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 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.

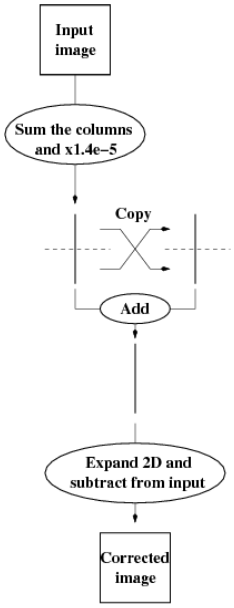

Figure 10.2.2: *Crosstalk effect correction.*

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.

The 'non- flatness' of the array is also due to the variations of the detector efficiency. In fact, the circular structures in the SofI images are real detector features (note that they also have a few dead pixels at their centers).

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 too few object frames (the limit can be specified by the user) 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. The recipe 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 related to the cross-correlation search size specified by the user. The default corresponds to an offset error around 5 pixels.

   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. On the other hand, the linear interpolation has an important advantage that it conserves the flux. It may be preferable in case when the main objective is accurate photometry.

   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 10.2.3 shows an example of a raw image and the result obtained with the *sofi_img_jitter* recipe.

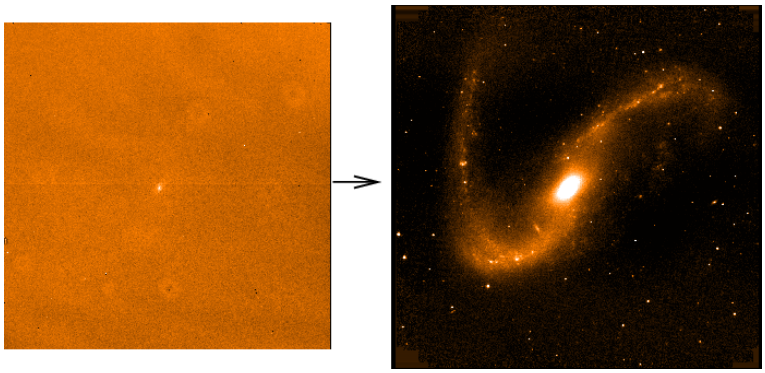

Figure 10.2.3: *Jitter results in imaging. Raw file (left) and combined result (right).*

10.2.7 sofi_spc_flat

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

In each of the settings, 4 FITS files are expected. The list consists of a series of 'off'/'on'/'on'/'off' images.

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 2 'on'-'off' pairs.

The algorithm is illustrated in figure 10.2.4.

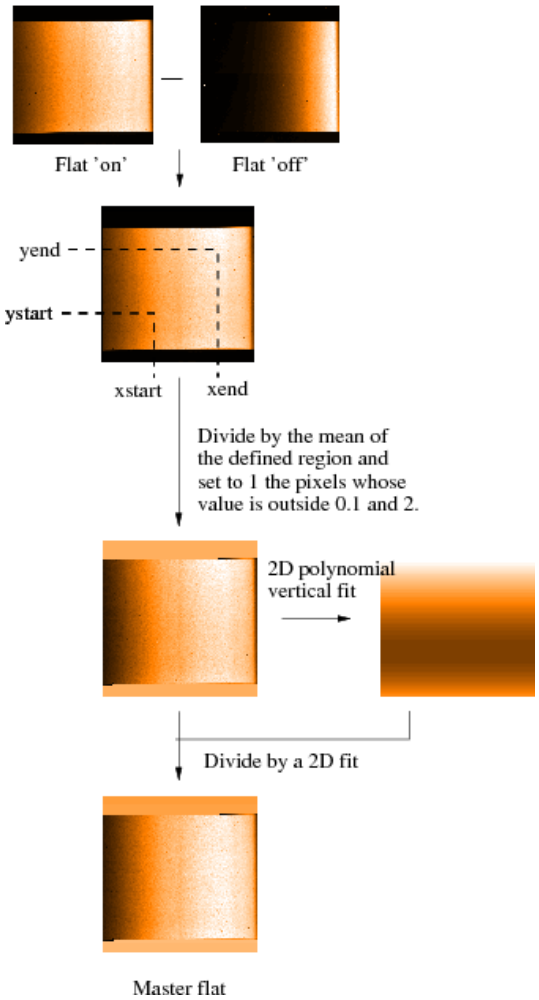

Figure 10.2.4: *Flat field reduction.*

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

10.2.8 sofi_spc_arc

SOFI 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 vertical 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

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 Neon. 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:

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

The dispersion relation is computed by matching a Xenon and/or Neon 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 10.2.5 shows a raw image and the corresponding corrected one.

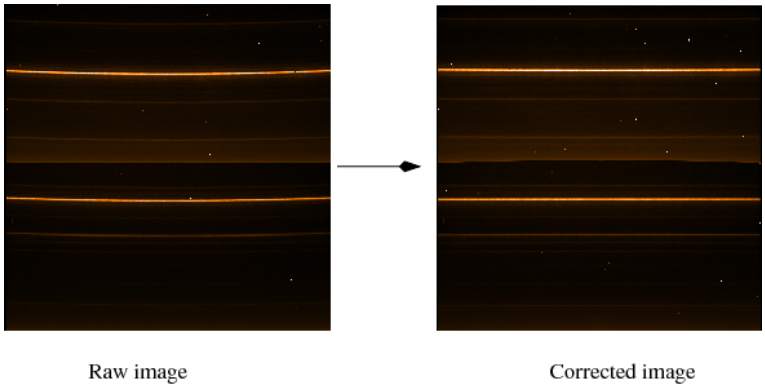

Figure 10.2.5: *Slit curvature correction.*

10.2.9 sofi_spc_jitter

The recipe accepts in input different calibration files:

- The FITS catalog with the OH emission lines
- A table to use to correct for slit curvature
- A table to use for wavelength calibration
- 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.

The frames are calibrated (flat-field, distortion) and the cross-talk effect is corrected.

If a table containing a wavelength solution is provided, this one is used. Otherwise, the wavelength calibration is computed by the recipe. In order to compute the wavelength calibration, 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.

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.

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

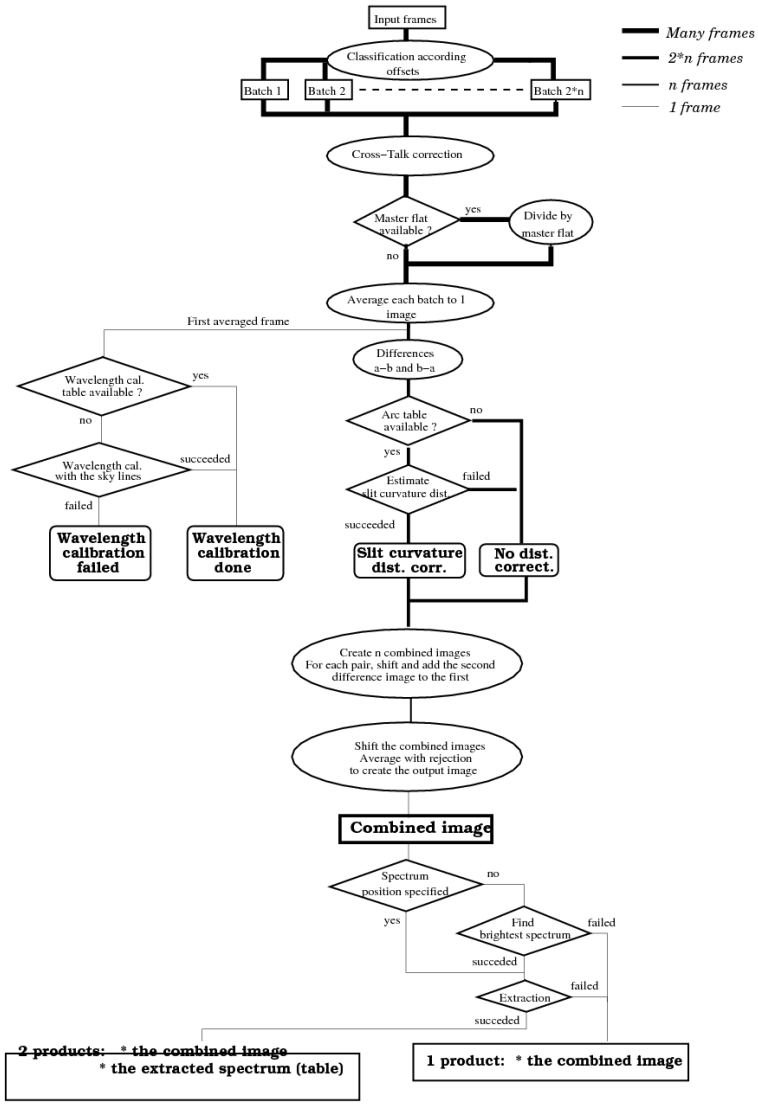

Figure 10.2.6: *A flowchart of the steps in sofi_spc_jitter.*

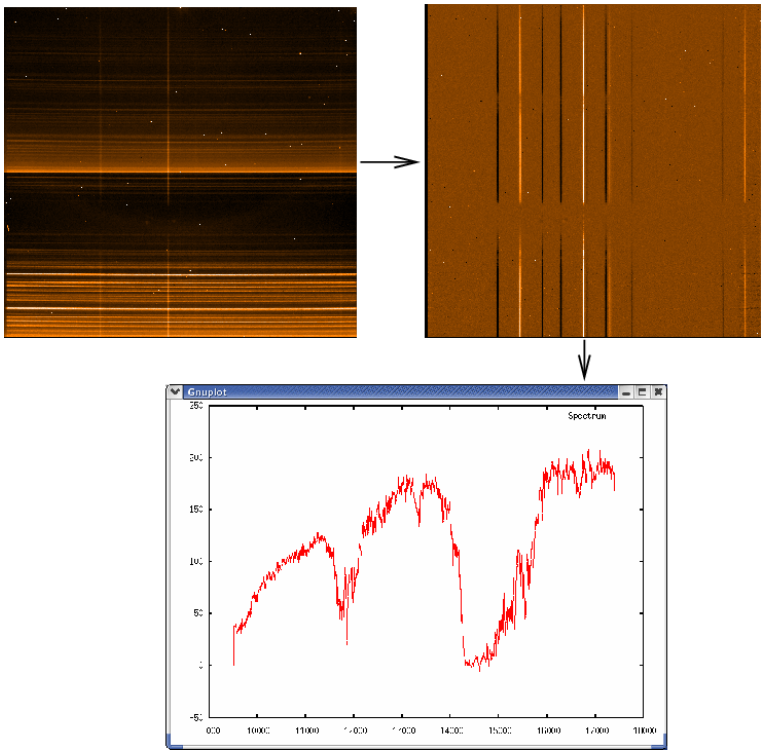

Figure 10.2.7: *Jitter results in spectroscopy.*

A Installation

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

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

A.2.1 Requirements

To compile and install the SOFI 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 SOFI 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 14). 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 SOFI pipeline

The SOFI pipeline distribution kit 1.5.0contains:

sofi-pipeline-manual-1.2.pdf	The SOFI pipeline manual
install_pipeline	Install script
cfitsio2510.tar.gz	CFITSIO 2510
cpl-4.1.0.tar.gz	CPL 4.1.0
esorex-3.6.8.tar.gz	esorex 3.6.8
gasgano-2.2.7.tar.gz	GASGANO 2.2.7
sofi-1.5.0.tar.gz	SOFI 1.5.0
sofi-calib-1.5.0.tar.gz	SOFI calibration files 1.5.0

Here is a description of the installation procedure:

1. Change directory to where you want to retrieve the SOFI 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 SOFI pipeline distribution.

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

4. Unpack using the following command:

   gunzip sofi-kit-1.5.0.tar.gz

   tar -xvf sofi-kit-1.5.0.tar

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

   cd sofi-kit-1.5.0

   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 SOFI 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*).

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
ISAAC	Infrared Spectrometer And Array Camera
OB	Observation Block
QC	Quality Control
RON	Read Out Noise
SOF	Set Of Frames
SOFI	Son OF Isaac
UT	Unit Telescope
VLT	Very Large Telescope