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

VISIR Pipeline User Manual

VLT-MAN-ESO-19500-3852

Issue 1.0

Date 1st February 2006

Prepared:	L. K. Lundin	1st February 2006	
	Name	Date	Signature
Approved:	P.Ballester, M. Peron		
	Name	Date	Signature
Released:	P. Quinn		
	Name	Date	Signature

Change record

Issue/Rev.	Date	Section/Parag. affected	Reason/Initiation/Documents/Remarks
1.0	1st February 2006	All	First version

Contents

1 Introduction

1.1 Purpose

The VISIR pipeline is a subsystem of the *VLT Data Flow System* (DFS). It is used in two operational environtments, 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 VISIR 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 VISIR data reduction sequence with the VISIR pipeline.

This manual is a complete description of the data reduction recipes used by the VISIR pipeline, reflecting the status of the VISIR pipeline as of 1st February 2006 (version 1.3.7).

1.2 Acknowledgements

We want to thank Eric Pantin, CEA and Ralf Siebenmorgen for providing valuable ideas for improving the pipeline recipes. We thank also Yves Jung, who played a major role in the development of the first version of the pipeline. The feedback we received in numerous discussions with our “beta-testers”, VISIR Instrument Scientists Alain Smette and Stephane Brillant, and VISIR Quality Control Scientist Danuta Dobrzycka was very much appreciated.

1.3 Scope

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

1.4 Applicable documents

[1]	VLT Data Flow System Specifications for Pipeline and Quality Control	VLT-SPE-ESO-19600-1233
[2]	Data Flow for VLT instruments Requirement Specification	VLT-SPE-ESO-19000-1618/2.0
[3]	DFS Pipeline & Quality Control – User Manual	VLT-MAN-ESO-19500-1619
[4]	ESO DICB – Data Interface Control Document	GEN-SPE-ESO-19400-0794
[5]	Common Pipeline Library User Manual	VLT-MAN-ESO-19500-2720
[6]	Gasgano User’s Manual	VLT-PRO-ESO-19000-1932
[7]	VISIR Calibration Plan	VLT-PLA-VIS-14300-0009

1.5 Reference documents

[8]	VISIR User Manual	VLT-MAN-ESO-14300-3514

[9] Parameters for setting the VISIR Spectrometer VLT-TRE-VIS-14321-5046
[10] Rio Y. et al. , VISIR: A mid infrared imager and spectrometer for the VLT, SPIE Vol. 2475, pp. 286-295, Orlando, April 1995
[11] Lagage, P.-O. et al. , Result of the phase A study for the VLT Mid-infrared instrument: VISIR, The ESO Messenger, No. 80, pp. 13-16, June 1995
[12] Lagage, P.-O. et al. , VISIR at PDR, The (ESO) Messenger, No. 91, pp. 17-21, March 1998
[13] Rio Y. et al. , VISIR: The mid infrared imager and spectrometer for the VLT, SPIE Vol. 3354, pp. 615-626, Kona, Hawaii, March 1998
[14] Lagage, P.-O., The final design of VISIR, the mid-infrared imager and spectrometer for the VLT, SPIE Vol. 4008, pp. 1120 - 1131, March 2000
[15] Common Pipeline Library Reference Manual VLT-ESO-MAN-19500-2721

1.6 Abbreviations and acronyms

CPL	Common Pipeline Library
DFS	Data Flow System
DMD	Data Management Division
DO	Data Organiser
DRS	Data Reduction System
FITS	Flexible Image Transport System
HR	High Resolution
LR	Low Resolution
MR	Medium Resolution
QC	Quality Control
RB	Reduction Block
RBS	Reduction Block Scheduler
SOF	Set Of Frames
VISIR	VLT Imager and Spectrometer for the InfraRed
VLT	Very Large Telescope
WCS	World Coordinate System

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 every scientific goal.

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* [6] and *EsoRex*, both included in the pipeline distribution (see Appendix A on page 60). 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 4 on page 14.

The VISIR instrument and the different types of VISIR raw frames and auxiliary data are described in sections 3 on the following page, 6 on page 24, and 7 on page 28.

A brief introduction to the usage of the available reduction recipes using Gasgano or EsoRex is presented in section 4 on page 14. In section 5 on page 23 we advice the user about known data reduction problems providing also possible solutions.

An overview of the data reduction, what are the input data, and the recipes is provided in section 8 on page 30.

More details on what are inputs, products, quality control measured quantities, and controlling parameters of each recipe is given in section 9 on page 32.

More detailed descriptions of the data reduction algorithms used by the individual pipeline recipes can be found in section 10 on page 48.

In appendix A on page 60 the installation of the VISIR pipeline recipes is given.

3 VISIR Instrument Description

This section provides a brief description of the VISIR instrument.

A more complete documentation can be found in the VISIR User Manual, downloadable from `http://www.eso.org/instruments/visir/doc`

VISIR has been developed under ESO contract by CEA/DAPNIA/SAP and NFRA/ASTRON. The instrument has been made available to the community and started operations in Paranal in April 2005.

3.1 Instrument overview

The VISIR instrument is located at the Cassegrain focus of UT3 of the VLT at Paranal. It provides diffraction-limited imaging in the two mid infrared (MIR) atmospheric windows: the N band between 8 to 13μm and the Q band between 16.5 and 24.5μm, respectively.

It also features a spectrometer offering long-slit spectroscopy at low resolution (down to 150) in the N band, medium resolution in the N and Q band and high resolution (up to 30000) for a limited set of wavelengths, as well as cross-dispersed high resolution spectroscopy over most of the N and Q band.

Because of the very high background from the ambient atmosphere and telescope, the sensitivity of ground–based MIR instruments cannot compete with that of space-borne ones. However, ground based instruments mounted on large telescopes offer superior spatial resolution. For example VISIR at the VLT provides diffraction limited images at $\sim 0.3''$ (FWHM) in the N band. This is an order of magnitude better than what can be reached by the Spitzer Space Telescope (SST).

The VISIR imager and spectrometer are each equipped with a DRS (former Boing) 256x256 BIB detector. The quantum efficiency of the detectors reaches close to 70% in the N-band and has a sharp absorption feature at 8.8μm. See also 7.3 on page 28.

For a description of the VISIR instrument, please see [8]. *Parameters for setting the VISIR Spectrometer* (VLT-TRE-VIS-14321-5046) contains more information about the VISIR spectrometer.

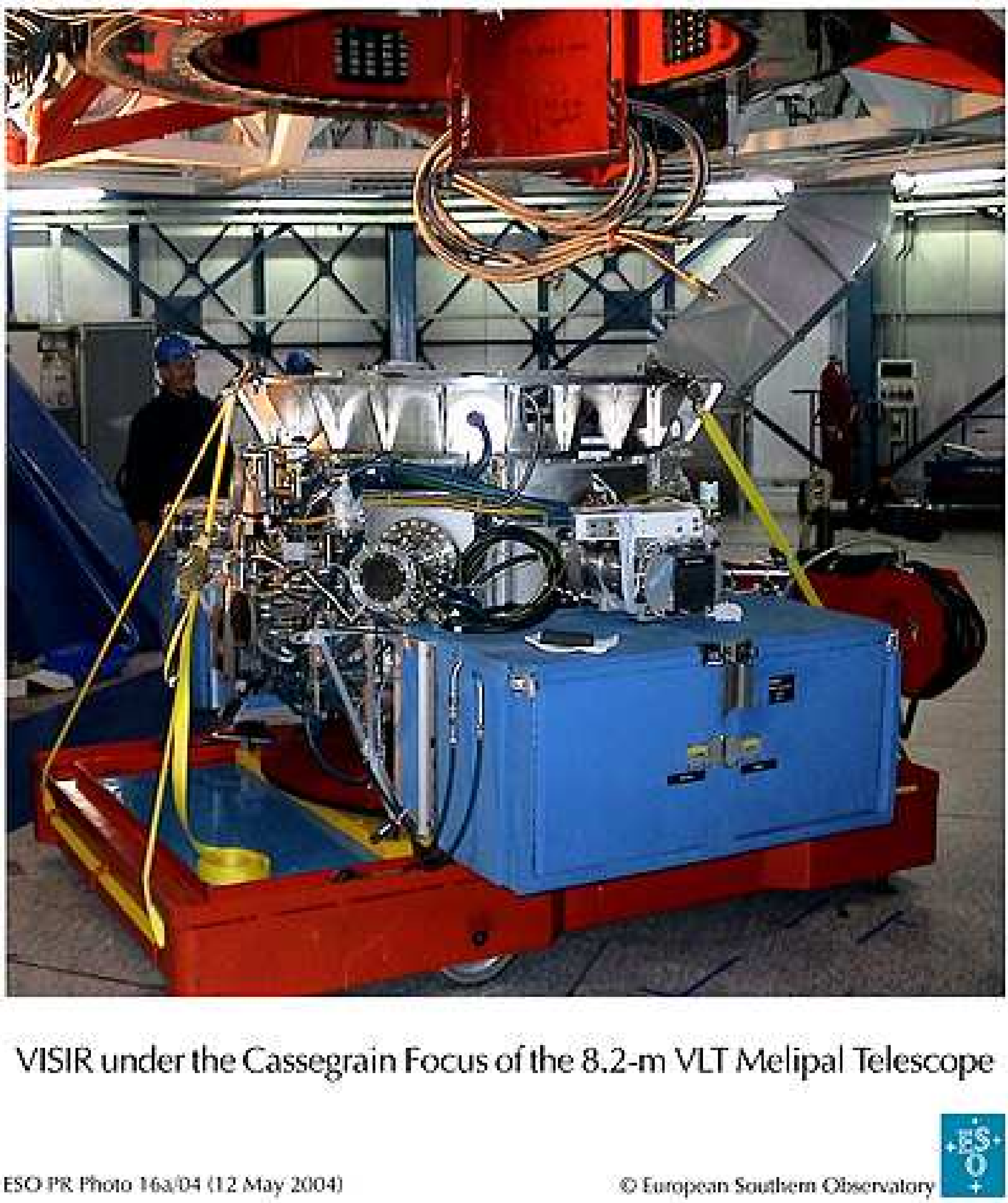

Figure 3.1.1: *The VISIR Instrument.*

4 Quick start

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

4.1 VISIR pipeline recipes

For reduction of science and calibration data, the current VISIR pipeline offers a set of 9 stand-alone recipes:

visir_img_ff creates a flat field and a bad pixel image from a sequence of flat fields at different exposure levels in both imaging and spectroscopy.

visir_img_combine combines a stack of jittered science exposures into a single image and an additional contribution map.

visir_img_phot uses images of a photometric standard star in a given filter and corresponding flux listed in a catalogue to determine the conversion factor, i.e. the ratio between the number of integrated detector counts per second from the star and the astrophysical source flux. The photometric sensitivity is determined as well.

visir_spc_wcal estimates the dispersion relation using the atmospheric spectrum in a long-slit spectroscopy half-cycle frame.

visir_spc_obs wavelength calibration as **visir_spc_wcal** followed by spectrum extraction from a combined image.

visir_spc_phot uses images of a spectrophotometric standard star in a given spectroscopic setting and corresponding flux listed in a catalogue to determine the conversion factor per step in wavelength. The spectrophotometric sensitivity is determined as well.

visir_spc_wcal_ech same as **visir_spc_wcal**, but intended for the spectrometer echelle instead of the long slit.

visir_spc_obs_ech same as **visir_spc_obs**, but intended for the spectrometer echelle instead of the long slit.

visir_spc_phot_ech same as **visir_spc_phot**, but intended for the spectrometer echelle instead of the long slit.

4.2 An introduction to Gasgano and EsoRex

Before being able to call pipeline recipes on a set of data, the data must be properly classified, and associated with the appropriate calibrations. The *Data Classification* consists of tasks such as: "What kind of data am I?", *e.g.*, FLAT, "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[1].

[1] The data association is based on the value of the triplet of FITS keys DPR.CATG, DPR.TYPE and DPR.TECH

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 VISIR 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 Command Line Interface in the following way:

```
gasgano &
```

Figure 4.2.1 shows the *Gasgano* main window with the 4 VISIR calibration files automatically loaded, which is the configuration of the publicly available version 1.3.7 of the VISIR pipeline.

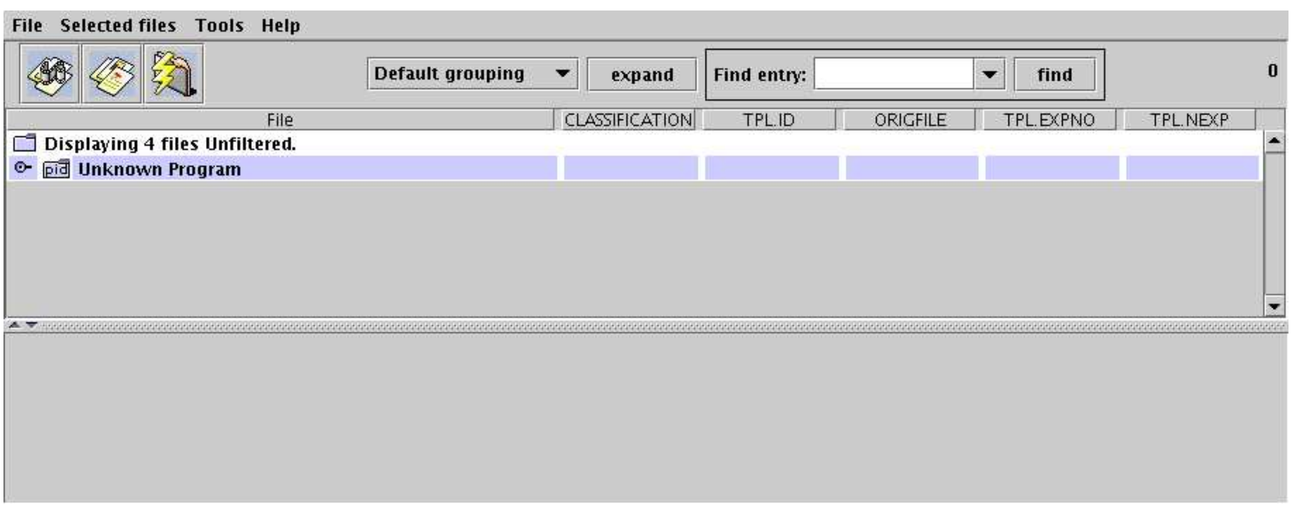

Figure 4.2.1: *The Gasgano main window with the VISIR calibration files.*

The VISIR calibration files become visible with their DO categories, when the file browser is expanded as shown in figure 4.2.2 on the next page. With the pull-down-menu *File->Add/Remove Files* directories containing

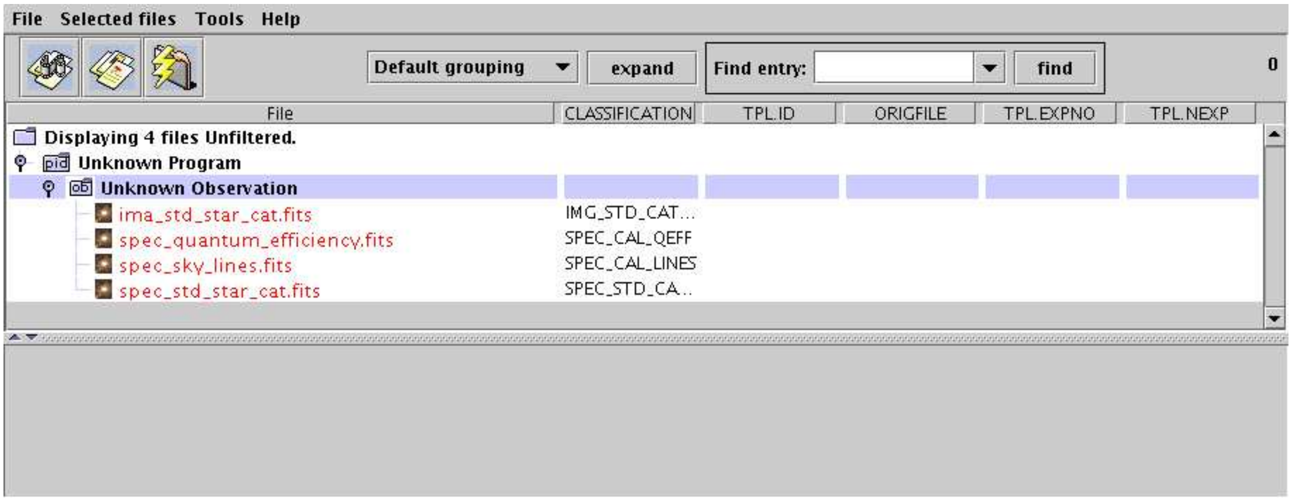

Figure 4.2.2: *The Gasgano main window with expanded file view of the initially loaded calibration files.*

VISIR data can be added, as shown in figure 4.2.3 on the facing page. Figure 4.2.4 on page 18 shows the example data set distributed with the public release of the VISIR pipeline.

The data are hierarchically organised as preferred by the user. After each file name are shown the classification, the template id, the original filename, the template exposure number and the number of exposures in the template.

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 with a <CTRL>-left-click for processing by the appropriate recipe: on Figure 4.2.5 on page 19, the example spectroscopic FITS-files and the three spectroscopic calibration files have been selected and the pull-down-menu with the VISIR recipes is shown.

Selecting the appropriate recipe, *visir_spc_phot_ech* , will open a *Gasgano* recipe execution window (see Figure 4.2.6 on page 21), 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 parameters may be 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. The succesful processing of the example data can be seen in figure 4.2.7 on page 22.

Please refer to the *Gasgano User's Manual* [6] for a more complete description of the *Gasgano* interface.

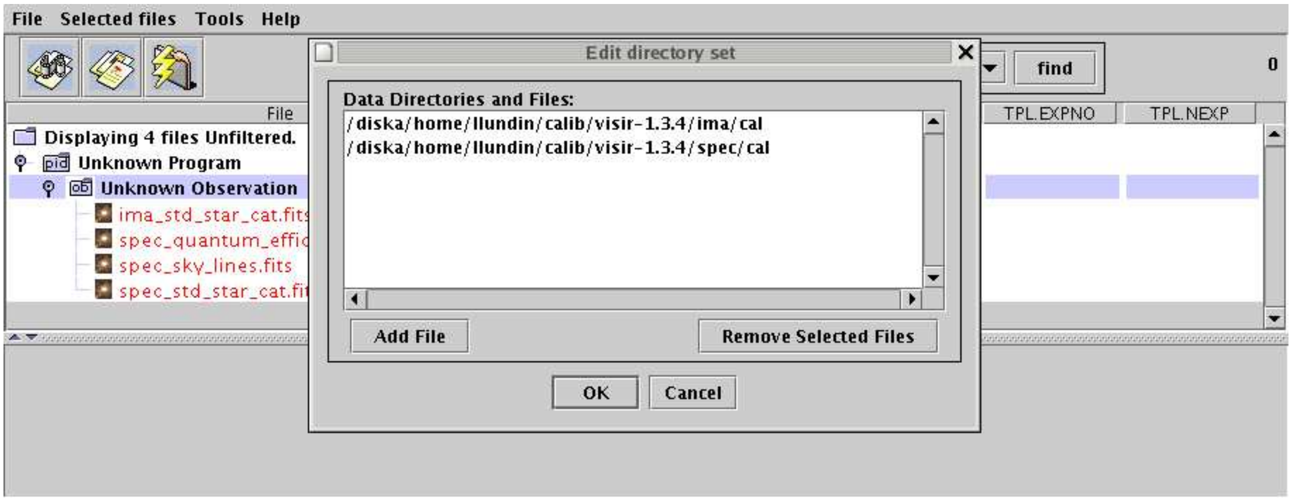

Figure 4.2.3: *The Gasgano main window with the file loader window on top.*

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 6.2 on page 25). 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 filenames must be listed together with their DO category in an ASCII file, the *set-of-frames* (SOF), that is required when launching a recipe. [2]

Here is an example of an SOF suitable for the *visir_spc_obs* recipe:

```
VISIR.2004-09-03T23:23:11.260.fits SPEC_OBS_LMR
VISIR.2004-09-03T23:24:05.263.fits SPEC_OBS_LMR
VISIR.2004-09-03T23:24:52.508.fits SPEC_OBS_LMR
VISIR.2004-09-03T23:25:45.026.fits SPEC_OBS_LMR
spec/cal/spec_sky_lines.fits SPEC_CAL_LINES
spec/cal/spec_quantum_efficiency.fits SPEC_CAL_QEFF
```

Note that the VISIR pipeline recipes do not verify the correctness of the DO category specified by the user in the SOF. The reason of this lack of control is that VISIR recipes are just one 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 on page 15).

[2] The set-of-frames corresponds to the *Input Frames* panel of the *Gasgano* recipe execution window (see figure 4.2.6 on page 21).

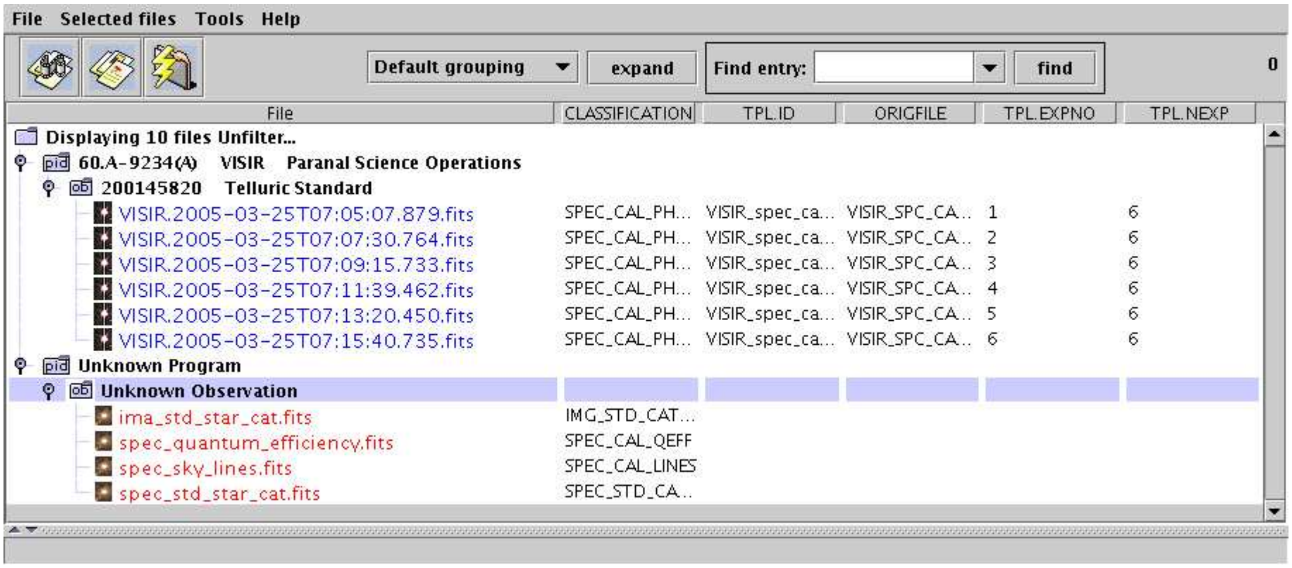

Figure 4.2.4: *The Gasgano main window with the example FITS files loaded.*

This lack of control can in some cases be an advantage. For example, observations made with the *VISIR_img_obs_AutoChopnod* template should in general be reduced by the the science observation recipe, *visir_img_combine*. However, if a photometric standard star was observed with this template, the sensitivity can still be determined by using the *visir_img_phot* recipe instead.

Such a procedure requires that the input FITS data are given a classification different from the one they should have.

If such a classification is done with FITS data that are not suitable for the given recipe, the recipe will most likely complete with a more or less descriptive error message, but there is a risk that the recipe will complete without any indication that the input is in fact invalid and the output 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

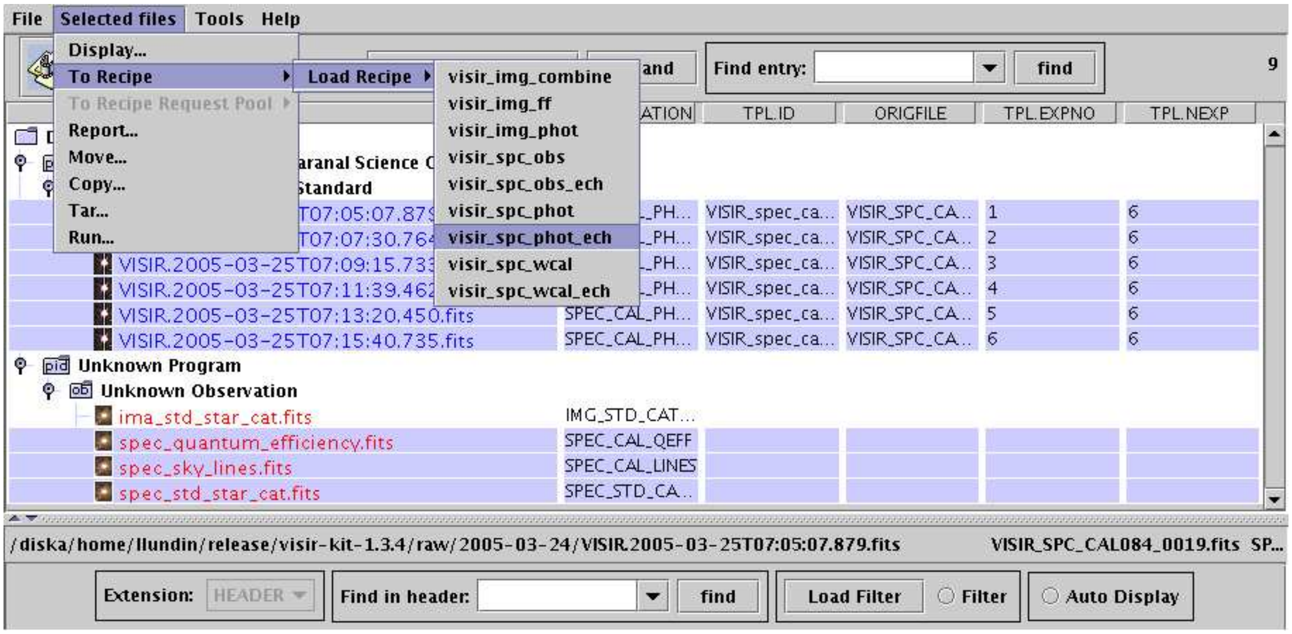

Figure 4.2.5: *The Gasgano main window with selected files and the pull-down-menu with the VISIR recipes.*

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.[3] 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 filename extension `.rc`. For instance, the recipe *visir_img_ff* has its *EsoRex* generated configuration file named `visir_img_ff.rc`, and is generated with the command:

esorex - -create-config visir_img_ff - -low=0.5

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

```
visir.visir_img_ff.low=0.5
```

In this example, the parameter visir.visir_img_ff.low is set to the value 0.5. 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.

Given a recipe named *visir_recipe_name* the command

esorex - -create-config visir_recipe_name

generates a default configuration file **visir_recipe_name.rc** in the directory **$HOME/.esorex**[4].

[3]The *EsoRex* recipe configuration file corresponds to the *Parameters* panel of the *Gasgano* recipe execution window (see figure 4.2.6 on page 21).

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

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 on page 32 and their role is described in section 10 on page 48.

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 *visir_spc_obs* for processing the files specified in the set-of-frames `visir_spc_obs.sof`:

esorex visir_spc_obs visir_spc_obs.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 *visir_img_ff* recipe *low* threshold parameter to `0.5`, the following should be typed:

esorex visir_img_ff --low=0.5 visir_img_ff.sof

Here are some more examples of running a recipe:

```
esorex --output-prefix=test visir_img_combine test.sof
esorex --msg-level=debug visir_spc_phot spc_phot.sof
esorex --time=true visir_img_phot --xcorr="15 15 15 15" img_phot.sof
```

For more information on *EsoRex*, see `http://www.eso.org/cpl/esorex.html`.

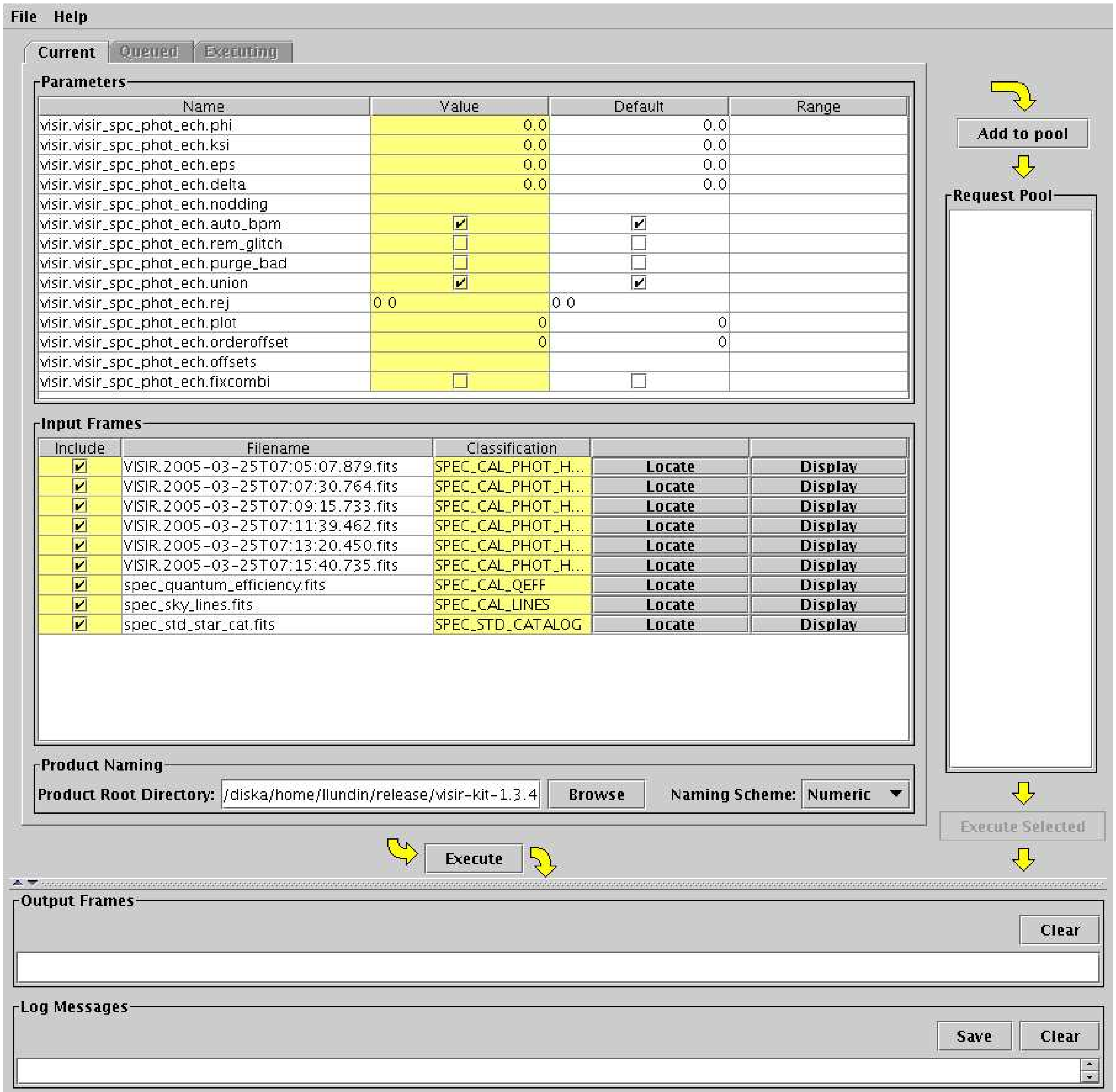

Figure 4.2.6: *The Gasgano recipe window with the recipe visir_spc_phot_ech.*

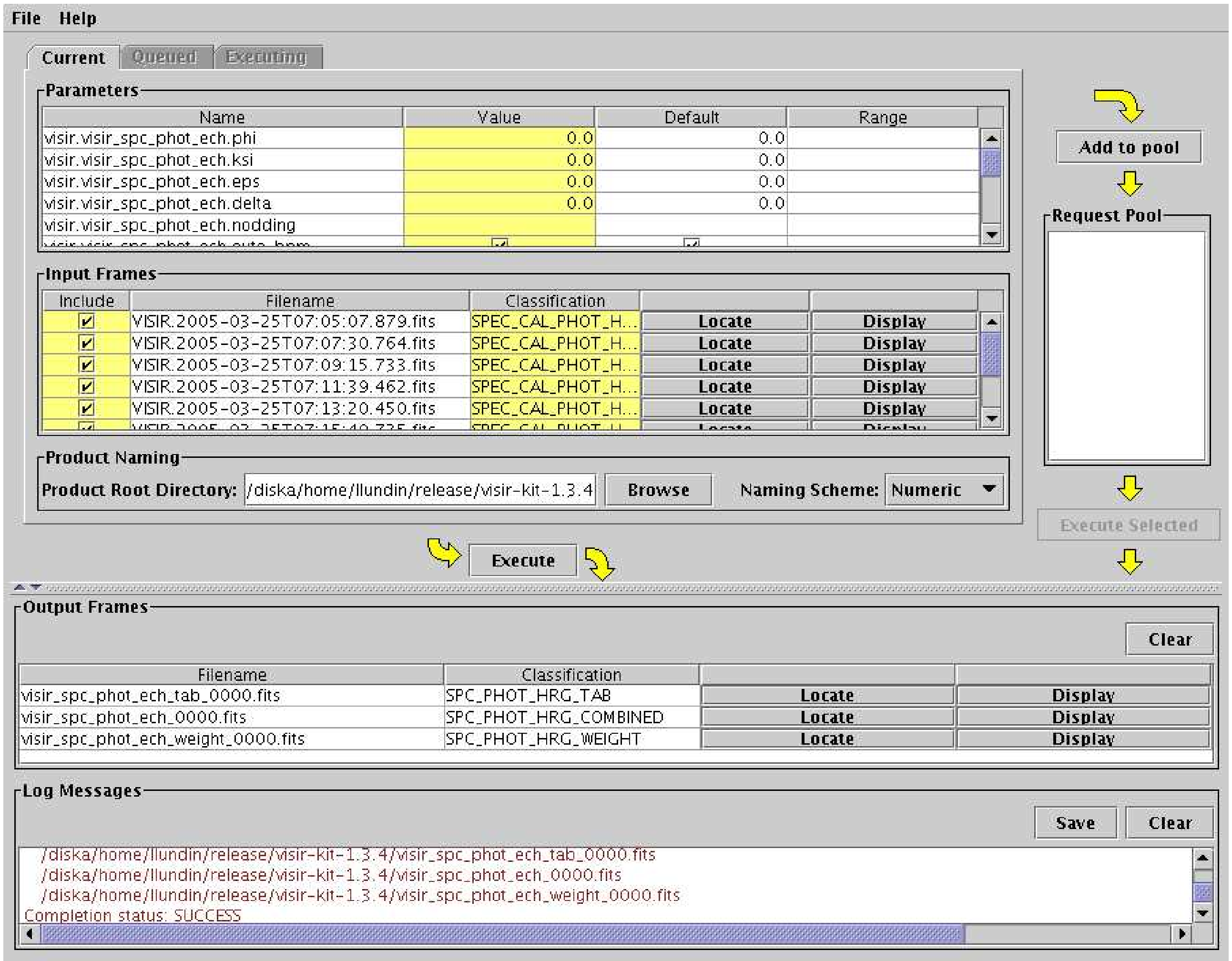

Figure 4.2.7: *The Gasgano recipe window with the recipe visir_spc_phot_ech successfully completed.*

5 Known Problems

The known problems of the VISIR pipeline version 1.3.7 are:

- Missing WCS coordinates in products with jittered input.
- The calibration table with sky emission data should be extended down to 6.9μm to allow processing of this lower wavelength range.
- The long slit spectrum extraction assumes that all beams fall on the detector.
- The spectral calibration does not take the drift of the high resolution spectroscopy scanner into account.
- In a few cases the Strehl ratio exceeds 1.

6 Instrument Data Description

VISIR data uses the FITS format and can be separated into *raw* frames and *product* frames. Raw frames are the unprocessed output of the VISIR instrument observations, while product frames are the result of the VISIR pipeline processing. In addition the VISIR pipeline uses a set of calibration (FITS-) files (standard stars catalogs, detector characteristics, etc.).

Any raw or product 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* [6], 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 standard star catalogue) 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 VISIR 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 raw and product VISIR data frames, that can be reduced by the VISIR pipeline version 1.3.7, 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 on page 17).

Raw frames can be classified as *imaging* frames or *spectroscopy* frames. Their intended use is implicitly defined by the assigned recipe.

6.1 General Data Layout

A raw VISIR file is an extension-less FITS-file. The data unit is a cube with NAXIS3= $2n + 1$ planes[5], where n is the number of chopping cycles, which is specified in the FITS-card with the key `HIERARCH ESO DET CHOP NCYCLES`. For each chopping cycle two so called Half-Cycle exposures are made, the A-image from the on-source position of the chopper, and the B-image from the off-source position of the chopper. Each half-cycle image is normalized by IRACE to an exposure time of one DIT; in other words, each half-cycle image is the average over the NDIT individual exposures. For each chopping cycle two planes are stored in the cube. The first two planes correspond to the first chopping cycle and contain:

- The Half-Cycle A-image, A_1. The pixel-values in each Half-Cycle image are offset by -32768, i.e. 32768 has to be added to each pixel in order to obtain the physical pixel value.
- The difference between the two Half-Cycle images, A_1-B_1.

Similarly, the $(2 \times i - 1)$th and $(2 \times i)$th planes correspond to the ith chopping cycle and contain

- The Half-Cycle A-image, A_i, stored with an offset identical to A_1.
- The average of the current and all previous Half-Cycle difference images, (A_1-B_1 + A_2-B_2 + ... + A_i-B_i)/i.

The last plane of the cube contains the average of all Half-Cycle difference images, i.e. it is identical to the $(2 \times n)$th plane.

[5] Before 2004-08-31 another data layout was used. The description of this now obsolete format is limited to the statement that it is also supported by the VISIR pipeline

6.2 General frames

These are data that can be obtained using any of the two instrument modes (imaging, spectroscopy), as is the case for flat field exposures. The keyword `ESO INS MODE` is set accordingly to 'IMG' for imaging frames, and to 'SPC' for spectroscopy frames, to indicate the intended use for the data.

- **Flat field:**

  Processed by: `visir_img_ff`

  Association keywords: `INSTRUME = VISIR`

  Classification:

DPR.CATG	DPR.TYPE	DPR.TECH	DO Category
CALIB	FLAT	IMAGE,DIRECT	IM_CAL_FLAT
CALIB	FLAT	SPECTRUM,DIRECT	SPEC_CAL_FLAT
TECHNICAL	FLAT	IMAGE,DIRECT	IM_TECH_FLAT
TECHNICAL	FLAT	SPECTRUM,DIRECT	SPEC_TECH_FLAT

See [4] for a definition of the values of DPR.CATG, DPR.TYPE and DPR.TECH.

6.3 Imaging frames

- **Science Observation:**

  Processed by: `visir_img_combine`

  Association keywords: `INSTRUME = VISIR`

  Classification:

DPR.CATG	DPR.TYPE	DPR.TECH	DO Category
SCIENCE	OBJECT	IMAGE,CHOPNOD,JITTER	IM_OBS_CHO_NOD_JIT
SCIENCE	OBJECT	IMAGE,CHOPPING,JITTER	IM_OBS_CHO_JIT
SCIENCE	OBJECT	IMAGE,NODDING,JITTER	IM_OBS_NOD_JIT
SCIENCE	OBJECT	IMAGE,DIRECT,JITTER	IM_OBS_DIR_JIT

- **Standard Star:**

  Processed by: `visir_img_phot`

  Association keywords: `INSTRUME = VISIR`

  Classification:

DPR.CATG	DPR.TYPE	DPR.TECH	DO Category
CALIB	STD	IMAGE,CHOPNOD	IM_CAL_PHOT

See [4] for a definition of the values of DPR.CATG, DPR.TYPE and DPR.TECH.

6.4 Spectroscopy frames

These frames are generated with the VISIR spectrometer.

- **Long Slit Wavelength Calibration:**
  Processed by: `visir_spc_wcal`
  Association keywords: `INSTRUME = VISIR`

  Classifi cation:

DPR.CATG	DPR.TYPE	DPR.TECH	DO Category
CALIB	WAVE	SPECTRUM,DIRECT	SPEC_CAL_LMR_WCAL

- **Long Slit Science Observation:**
  Processed by: `visir_spc_obs`
  Association keywords: `INSTRUME = VISIR`

  Classifi cation:

DPR.CATG	DPR.TYPE	DPR.TECH	DO Category
SCIENCE	OBJECT	SPECTRUM,CHOPNOD	SPEC_OBS_LMR

- **Long Slit Standard Star:**
  Processed by: `visir_spc_phot`
  Association keywords: `INSTRUME = VISIR`

  Classifi cation:

DPR.CATG	DPR.TYPE	DPR.TECH	DO Category
CALIB	STD	SPECTRUM,CHOPNOD	SPEC_CAL_PHOT

- **Echelle Wavelength Calibration:**
  Processed by: `visir_spc_wcal_ech`
  Association keywords: `INSTRUME = VISIR`

  Classifi cation:

DPR.CATG	DPR.TYPE	DPR.TECH	DO Category
CALIB	WAVE	ECHELLE	SPEC_CAL_HRG_WCAL

- **Echelle Science Observation:**
  Processed by: `visir_spc_obs_ech`
  Association keywords: `INSTRUME = VISIR`

  Classifi cation:

DPR.CATG	DPR.TYPE	DPR.TECH	DO Category
SCIENCE	OBJECT	ECHELLE	SPEC_OBS_HRG

- **Echelle Standard Star:**

  Processed by: `visir_spc_phot_ech`

  Association keywords: `INSTRUME = VISIR`

  Classifi cation:

DPR.CATG	DPR.TYPE	DPR.TECH	DO Category
CALIB	STD	ECHELLE	SPEC_CAL_PHOT_HRG

See [4] for a definition of the values of DPR.CATG, DPR.TYPE and DPR.TECH.

7 Static Calibration Data

This section describes the input required by the VISIR pipeline in addition to the instrument data. This input data is all stored as single-extension FITS tables.

7.1 Imaging Standard Stars

The catalogue of imaging standard stars currently contains information about 425 stars. For each star the catalogue contains the following information:

- The star name, e.g. HD108903.
- Spectral type, e.g. M3.5V.
- Right ascension and declination.
- The flux $[Jy]$ for each of the 23 supported imaging filters.
- The flux $[Jy]$ for each of the 6 supported spectroscopy filters.

This catalogue is stored in the file `ima/cal/ima_std_star_cat.fits`.

See also [8].

7.2 Spectroscopy Standard Stars

The catalogue of spectroscopy standard stars currently contains information about 469 stars, namely the 425 standard stars used for imaging and an additional 44 Hipparcos standard stars. For each star the catalogue contains the following information:

- The star name, e.g. HD108903 or HIP100469.
- Right ascension and declination.
- The model flux $[mJy]$ for 2300 wavelengths in the range 5 to 28 μm.

This catalogue is stored in the file `spec/cal/spec_std_star_cat.fits`.

See also [8].

7.3 Spectrometer Detector Quantum Efficiency

The spectrometer detector quantum efficiency at various wavelengths is stored in `spec/cal/spec_quantum_efficiency.fits`. The quantum efficiency ranges from about 1% to close to 70% (at 11.9μm). See figure 7.3.1 on the facing page.

See also [8].

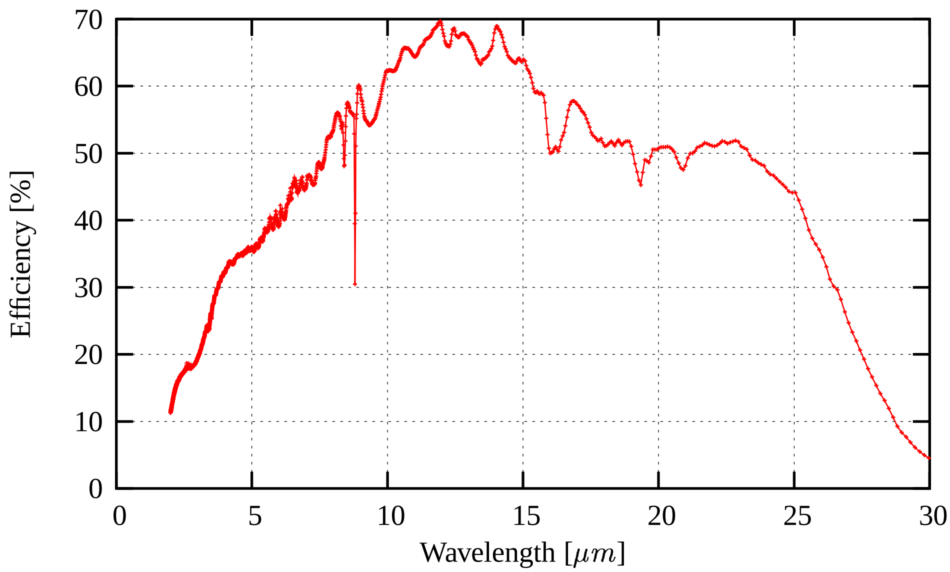

Figure 7.3.1: The Detector Quantum Efficiency.

7.4 Atmospheric Emission Spectrum

The atmospheric emission spectrum (normalized to 1) has been created using the HITRAN database of molecular line parameters and the US Standard Atmosphere atmospheric profile, for an altitude of 2600m and 1.5mm of precipitable water vapor at zenith.

The atmospheric emission is stored in `spec/cal/spec_sky_lines.fits`.

See also [8].

8 Data Reduction

In this section the usage of the VISIR pipeline recipes is described.

8.1 Reduction Cascade

The reduction cascade is a schematic representation of the recipe dependencies for pipeline products. There are currently no dependencies in the reduction cascade of the VISIR pipeline, with the exception that the output bad-pixel map of the **visir_img_ff** recipe may optionally be used as input by the other recipes.

8.2 VISIR pipeline recipes

The VISIR pipeline version 1.3.7 offers a set of 9 stand-alone recipes, intended for these fundamental operations:

Creation of general calibration data:

visir_img_ff: Creates a flat field and a bad pixel map from a sequence of flat fields at different exposure levels in both imaging and spectroscopy. See also the *VISIR Calibration Plan* (VLT-PLA-VIS-14300-0009) section 4.6.

Imaging data reduction:

visir_img_combine: Combines a stack of chopped, nodded and optionally jittered images into a single image and an additional contribution map.

Creation of calibration data for Imaging:

visir_img_phot: Determines the conversion factor between the number of detector counts and the astrophysical source flux using a catalogue of photometric standard stars and a set of images of a star from this catalogue. The photometric sensitivity is determined as well. Creates the same products as **visir_img_combine**. See also the *VISIR Calibration Plan* (VLT-PLA-VIS-14300-0009) sections 4.1.2 and 4.5.

Spectroscopic data reduction:

visir_spc_wcal: Estimates the dispersion relation using the atmospheric spectrum in a long-slit spectroscopy half-cycle frame. Creates a spectral table. See also the *VISIR Calibration Plan* (VLT-PLA-VIS-14300-0009) sections 5.1.1 and 5.2.

visir_spc_obs: Wavelength calibration as **visir_spc_wcal** followed by spectrum extraction from a combined image. Creates the same products as **visir_spc_phot**.

visir_spc_wcal_ech: Same as **visir_spc_wcal**, but intended for the spectrometer echelle instead of the long slit.

visir_spc_obs_ech: Same as **visir_spc_obs**, but intended for the spectrometer echelle instead of the long slit.

Creation of calibration data for Spectroscopy:

visir_spc_phot: Determines for the spectrometer long slit the conversion factor between the number of spectrometer detector counts and the astrophysical source flux using a catalogue of spectro-photometric standard stars. The photometric sensitivity is determined as well. Creates a combined image, a weight map and a spectral table. See also the *VISIR Calibration Plan* (VLT-PLA-VIS-14300-0009) sections 5.1.2 and 5.3.

visir_spc_phot_ech: Same as **visir_spc_phot**, but intended for the spectrometer echelle instead of the long slit..

Section 9 on the next page gives a general description on the use of recipes, together with more detailed information on the individual recipes.

8.3 Unsupported Observation Modes

A few VISIR templates lead to observations that are currently not supported by the pipeline. Amongst the currently offered science templates, this is only the case for the `VISIR_img_obs_GenericChopNod` template, used for raster (or mosaic) imaging.

Partial processing by the VISIR pipeline is however possible. Multiple images obtained at a same telescope positions can be reduced with the `visir_img_combine` recipes by manually setting the DO Category in EsoRex or the Classification field in Gasgano to `IM_OBS_DIR_JIT`. Sets of pairs of images obtained at two different telescope positions (object and sky) can be reduced in the same way.

9 Pipeline Recipe Interfaces

This section provides for each recipe examples of the required input data (and their classification). In the following we assume that /path_file_raw/filename_raw.fits and /path_file_cal/filename_cal.fits are existing FITS files (e.g. /data1/visir/com2/VISIR.2004-09-01T09:47:39.316.fits and /cal/visir/spec/cal/spec_quantum_efficiency.fits).

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

For each recipe we also list in a table the input parameters (as they appear in the recipe configuration file), the corresponding aliases (the corresponding names to be eventually set on command line) and their default values. Also quality control parameters are listed. Those are written in the headers of the 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 used in the ESO pipeline operations for quality control. The information in this file is the quality control data also found in the recipe products and as such this intermediate file can be ignored.

9.1 visir_img_ff

The VISIR pipeline recipe *visir_img_ff* creates a flat field and a bad pixel image from a sequence of flat fields at different exposure levels in both imaging and spectroscopy.

9.1.1 Input files for visir_img_ff

The input Set-Of-Frames shall specify at least two files with one of the DO categories:

DO Category	Type	Explanation
IM_CAL_FLAT	Raw Frame	Calibration Exposure
IM_TECH_FLAT	Raw Frame	Calibration Exposure
SPEC_CAL_FLAT	Raw Frame	Calibration Exposure
SPEC_TECH_FLAT	Raw Frame	Calibration Exposure

9.1.2 Input Parameters for visir_img_ff

The recognized recipe options are

Parameter	Possible Values (with **default**)	Explanation
low	float, **0.2**	Low threshold for the bad pixel map.
high	float, **5.0**	High threshold for the bad pixel map.

9.1.3 Products from visir_img_ff

Successful completion of this recipe will, depending on the input DO category, create one of these pairs of FITS-files

File name	Product Category (PRO CATG)	Explanation
visir_img_ff.fits	IMG_FF	The flat field (imaging).
visir_img_ff_bpm.fits	IMG_BPM	The map of bad pixels (imaging).
visir_img_ff.fits	SPEC_FF	The flat field (spectroscopy).
visir_img_ff_bpm.fits	SPEC_BPM	The map of bad pixels (spectroscopy).
visir_img_ff.fits	IMG_TECH_FF	Technical flat field (imaging).
visir_img_ff_bpm.fits	IMG_TECH_BPM	The map of bad pixels (imaging).
visir_img_ff.fits	SPEC_TECH_FF	Technical flat field (spectroscopy).
visir_img_ff_bpm.fits	SPEC_TECH_BPM	The map of bad pixels (spectroscopy).

The non-technical imaging bad pixel map can be used as input in the *visir_img_combine* and *visir_img_phot* recipes, and the non-technical spectroscopy bad pixel map can be used as input in the spectroscopy recipes.

9.1.4 QC Parameters from visir_img_ff

This recipe generates the Quality Control parameters `QC NBBADPIX` and `QC CAPA` and writes them in the FITS header of its products. See appendix B on page 63 for their definition.

9.2 visir_img_combine

The VISIR pipeline recipe *visir_img_combine* combines a stack of chopped, jittered and/or nodded exposures.

9.2.1 Input files for visir_img_combine

The input Set-Of-Frames shall specify at least one pair of files with one of the DO categories:

DO Category	Type	Explanation
IM_OBS_CHO_NOD_JIT	Raw Frame	Science Exposures
IM_OBS_CHO_JIT	Raw Frame	Science Exposures
IM_OBS_NOD_JIT	Raw Frame	Science Exposures
IM_OBS_DIR_JIT	Raw Frame	Science Exposures

Additionally, a calibration file with a bad pixel map with a PRO.CATG of IMG_BPM may be added to the Set-Of-Frames with DO category: IMG_BPM.

9.2.2 Input Parameters for visir_img_combine

The recognized recipe options are

<table>
<tr><td>Parameter</td><td>Possible Values (with default)</td><td>Explanation</td></tr>
<tr><td>auto_bpm</td><td>true/false</td><td>Enables/disables automatic detection and correction of bad pixels (see also section 10.1.1 on page 48)</td></tr>
<tr><td>plot</td><td>0,1,2</td><td>The recipe can produce a number of predefined plots. Zero means that none of the plots are produced, while increasing values (e.g. 1 or 2) increases the number of plots produced. If the plotting fails a warning is produced, and the recipe continues.
The default behaviour of the plotting is to use gnuplot (with option -persist). The recipe currently produces 1D-plots using gnuplot commands. The recipe user can control the actual plotting-command used by the recipe to create the plot by setting the environment variable IRPLIB_PLOTTER. Currently, if IRPLIB_PLOTTER is set it must contain the string 'gnuplot'. Setting it to 'cat > my_gnuplot_$$.txt' causes a number of ASCII-files to be created, which each produce a plot when given as standard input to gnuplot (e.g. later or on a different computer). A finer control of the plotting options can be obtained by writing an executable script, e.g. my_gnuplot.pl, that executes gnuplot after setting the desired gnuplot options (e.g. set terminal pslatex color) and then setting IRPLIB_PLOTTER to my_gnuplot.pl.
The predefined plots include plotting of images. Images can be plotted not only with gnuplot, but also using the pnm format. This is controlled with the environment variable IRPLIB_IMAGER. If IRPLIB_IMAGER is set to a string that does not contain the word gnuplot, the recipe will generate the plot in pnm format. E.g. setting IRPLIB_IMAGER to 'cat | display - &' will produce a gray-scale image using the image viewer display.</td></tr>
<tr><td>nod</td><td>filename, <none></td><td>An optional ASCII specification of the nodding positions (in case they are missing from the FITS-file).
The file must consist of one line per input FITS-file and each line must consist of an integer (which is ignored) followed by a 0 or 1 (to indicate object or sky).</td></tr>
<tr><td>off</td><td>filename, <none></td><td>An optional ASCII specification of the offsets in case they are missing from the FITS-file.
The file must consist of one line per input pair of</td></tr>
</table>

		FITS-files, and each line must consist of two numbers, which represent the shift in pixel of that image. The reference point of the coordinates of the shift may be chosen by the user. A convenient reference point causes the offset of the first image to be (0,0) and the other offsets to be relative to the first image.
g	true/**false**	Enables/disables automatic filtering of glitches. This is an experimental filtering of the input images. It is I/O- and CPU-intensive and is better left off.
p	true/**false**	Enables/disables automatic purging of half-cycle images whose median deviates more than a factor three from the mean of the medians of half-cycle images or whose standard deviation deviates more than a factor three from the mean of their standard deviations. It is I/O- and CPU-intensive with no certain improvement to the output.
union	**true**/false	Combine images using their union, as opposed to their intersection.
rej	"%u %u", **"0 0"**	Each resulting pixel is the average of the corresponding (interpolated) pixel value in each jittered image. A positive value, n1, for the first of the two integers specifies that for each pixel the smallest n1 pixel values shall be ignored in the averaging. Similarly, a positive value, n2, for the second of the two integers specifies that for each pixel the largest n2 pixel values shall be ignored in the averaging.
ref	true/**false**	Enables/disables user-defined refining of the offsets. Enabling requires a specification of object positions. See options objs and xcorr and subsection 10.1.4 on page 49.
objs	filename, **<none>**	The shift and add of images needs anchor points that typically are bright objects. These are normally detected automatically but with user-defined refining of offsets enabled, they must be provided by the user through an ASCII file containing one line per anchor point with each line consisting of its x and y coordinate (in pixels). This file is ignored with user-defined refining of offsets disabled. See also subsection 10.1.4 on page 49.
xcorr	"%u %u %u %u", **"10 10 25 25"**	If user-defined refining of offsets is enabled a cross-correlation of the images is performed. In order to speed up this process, this cross-correlation is performed only on smaller rectangles around the anchor points. The first two parameters is the half-size of this rectangle in pixels. The second pair is the maximum shift in x and y (pixels) evaluated by the cross-correlation on

the rectangle.
Used only if user-defined refining of offsets is enabled.
See also subsection 10.1.4 on page 49.

9.2.3 Products from visir_img_combine

Successful completion of this recipe will, depending on the input DO category, create one of these pairs of FITS-files

File name	Product Category (PRO CATG)	Explanation
visir_img_combine.fits	IMG_OBS_COMBINED_CNJ	the combined image (CHOPNOD).
visir_img_combine_contrib.fits	IMG_OBS_CONTRIB_MAP_CNJ	the contribution map (CHOPNOD).
visir_img_combine.fits	IMG_OBS_COMBINED_CJ	the combined image (CHOPPING).
visir_img_combine_contrib.fits	IMG_OBS_CONTRIB_MAP_CJ	the contribution map (CHOPPING).
visir_img_combine.fits	IMG_OBS_COMBINED_NJ	the combined image (NODDING).
visir_img_combine_contrib.fits	IMG_OBS_CONTRIB_MAP_NJ	the contribution map (NODDING).
visir_img_combine.fits	IMG_OBS_COMBINED_DJ	the combined image (DIRECT).
visir_img_combine_contrib.fits	IMG_OBS_CONTRIB_MAP_DJ	the contribution map (DIRECT).

See 9.10.1 and 9.10.2 on page 46 for a description of these products.

9.2.4 QC Parameters from visir_img_combine

This recipe generates the Quality Control parameters `QC BACKGD MEAN` and `QC CAPA` and writes them in the FITS header of its products. See appendix B on page 63 for their definition.

9.3 visir_img_phot

The VISIR pipeline recipe *visir_img_phot* determines the conversion factor between the number of detector counts and the astrophysical source flux using a catalogue of photometric standard stars. The photometric sensitivity is determined as well.

9.3.1 Input files for visir_img_phot

The input Set-Of-Frames shall specify at least one file with the DO category:

DO Category	Type	Explanation
IM_CAL_PHOT	Raw Frame	Calibration Exposures

Additionally, a calibration file with a bad pixel map with a PRO.CATG of IMG_BPM may be added to the Set-Of-Frames with DO category: IMG_BPM.

9.3.2 Input Parameters for visir_img_phot

The recognized recipe options are those of recipe **visir_img_combine** (see section 9.2 on page 33).

9.3.3 Products from visir_img_phot

Successful completion of this recipe will create the FITS-files

File name	Product Category (PRO CATG)	Explanation
visir_img_phot.fits	IMG_PHOT_COMBINED	the combined image.
visir_img_phot_contrib.fits	IMG_PHOT_CONTRIB_MAP	the contribution map.

See 9.10.1 and 9.10.2 on page 46 for a description of these products.

9.3.4 QC Parameters from visir_img_phot

This recipe generates the Quality Control parameters

```
QC BACKGD MEAN
QC BACKGD SIGMA
QC CAPA
QC CONVER
QC EXPTIME
QC FILTER
QC FLUXSNR
QC FLUXSNR NOISE
QC FLUXTOT
QC FWHMX
QC FWHMX NEG1
QC FWHMX NEG2
QC FWHMX POS1
QC FWHMX POS2
QC FWHMY
QC FWHMY NEG1
QC FWHMY NEG2
```

```
QC FWHMY POS1
QC FWHMY POS2
QC JYVAL
QC SENSIT
QC STARNAME
QC STREHL
QC STREHL ERROR
```

and writes them in the FITS header of its products. See appendix B on page 63 for their definition.

9.4 visir_spc_wcal

The VISIR pipeline recipe *visir_spc_wcal* estimates the dispersion relation using the atmospheric spectrum in a long-slit spectroscopy half-cycle frame.

9.4.1 Input files for visir_spc_wcal

The input Set-Of-Frames shall specify files with DO categories:

DO Category	Type	Explanation
SPEC_CAL_LMR_WCAL	Raw Frame	Calibration or Science Exposures (at least one pair)
SPEC_CAL_LINES	Calibration	Atmospheric Transmission
SPEC_CAL_QEFF	Calibration	Detector Quantum-Efficiency
BPM	Calibration	Optional bad pixel map with PRO.CATG SPEC_BPM

Input of type Calibration is static calibration data, which is part of the pipeline. The science exposures must be obtained with long slit spectroscopy (low, medium or high resolution).

9.4.2 Input Parameters for visir_spc_wcal

The recognized recipe options are

Parameter	Possible Values (with **default**)	Explanation
auto_bpm	**true**/false	See subsection 9.2 on page 33.
plot	**0**,1,2	See subsection 9.2 on page 33. The figures in section 10 on page 48 have been made with plot=2.
slit_skew	`float`, **1.6**	Distortion correction: Skew of slit [degrees] (clockwise)
spectrum_skew	`float`, **0.7**	Distortion correction: LMR Skew of spectrum [degrees] (counter-clockwise). Not used in High Resolution.
vert_arc	`float`, **1.04**	Distortion correction: LR Detector vertical curvature [pixel]. Reduced by a factor 4 in MR. Not used in HR A-side.

		Increased by a factor 115/52 in HR B-side.
hori_arc	`float`, **0.08**	Distortion correction: LMR Detector horizontal curvature [pixel]. Increased by a factor 1.5 in HR A-side. Reduced by a factor 2 in HR B-side.

Please see 10.1.2 on page 48 for a description of the distortion correction. The default values for the two parameters of the curvature correction are set to values that are optimal in low resolution spectroscopy. The recipe can only determine the actual resolution of its input data after the input parameters have been processed. The reduction- and increase-factors for the parameters of the curvature correction are therefore necessary in order for the parameter default values to yield an optimal distortion correction also in medium and high resolution. This scaling is applied also to the user supplied values of these parameters. For example, the default value of vert_arc is 1.04. If the input data is obtained in medium resolution, the value of vert_arc is reduced by a factor 4, i.e. to 0.26. If the recipe user changes this parameter to 1.20 and uses it on data obtained in medium resolution, a value of 1.20/4 = 0.30 is actually used in the correction.

9.4.3 Products from visir_spc_wcal

Successful completion of this recipe will create the FITS-file

File name	Product Category (PRO CATG)	Explanation
visir_spc_wcal_spectrum_tab.fits	SPC_WCAL_LMR_TAB	the spectral table.

See 9.10.4 on page 46 for a description of this product.

9.4.4 QC Parameters from visir_spc_wcal

This recipe generates the Quality Control parameters

```
QC BACKGD MEAN
QC CAPA
QC PHDEGREE
QC PHDISPX0
QC PHDISPX1
QC XC
QC XCDEGREE
QC XCDISPX0
QC XCDISPX1
QC XCENTROI
QC XCSHIFT
QC XCWLEN
QC XFWHM
```

and writes them in the FITS header of its products. See appendix B on page 63 for their definition.

9.5 visir_spc_obs

The VISIR pipeline recipe *visir_spc_obs* performs a wavelength calibration as **visir_spc_wcal** followed by spectrum extraction from a combined image.

9.5.1 Input files for visir_spc_obs

The input Set-Of-Frames shall specify files with DO categories:

DO Category	Type	Explanation
SPEC_OBS_LMR	Raw Frame	Long Slit Science Exposures (at least one pair)
SPEC_CAL_LINES	Calibration	Atmospheric Transmission
SPEC_CAL_QEFF	Calibration	Detector Quantum-Efficiency
BPM	Calibration	Optional bad pixel map with PRO.CATG SPEC_BPM

Input of type Calibration is static calibration data, which is part of the pipeline. The science exposures must be obtained with long slit spectroscopy (low, medium or high resolution).

9.5.2 Input Parameters for visir_spc_obs

The recognized recipe options are those of recipe **visir_spc_wcal** (see section 9.4 on page 38) and additionally these options, described in subsection 9.2 on page 33: nod, g, p, union and rej.

9.5.3 Products from visir_spc_obs

Successful completion of this recipe will create the FITS-files

File name	Product Category (PRO CATG)	Explanation
visir_spc_obs_spectrum_tab.fits	SPC_OBS_LMR_TAB	the spectral table.
visir_spc_obs.fits	SPC_OBS_LMR_COMBINED	the combined image (the 2D spectrum).
visir_spc_obs_weight.fits	SPC_OBS_LMR_WEIGHT	the weight map.

See 9.10.5, 9.10.1 and 9.10.5 on page 46 for a description of these products.

9.5.4 QC Parameters from visir_spc_obs

This recipe generates the same Quality Control parameters as those of recipe visir_spc_wcal, see subsubsection 9.4.4 on the page before.

9.6 visir_spc_phot

The VISIR pipeline recipe *visir_spc_phot* performs a wavelength calibration as **visir_spc_wcal** followed by spectrum extraction from a combined image finalized by a determination of photometric sensitivity.

9.6.1 Input files for visir_spc_phot

The input Set-Of-Frames shall specify files with DO categories:

DO Category	Type	Explanation
SPEC_CAL_PHOT	Raw Frame	Calibration Exposures (at least one pair)
SPEC_CAL_LINES	Calibration	Atmospheric Transmission
SPEC_CAL_QEFF	Calibration	Detector Quantum-Efficiency
SPEC_STD_CATALOG	Calibration	Catalogue of spectroscopy standard stars
BPM	Calibration	Optional bad pixel map with PRO.CATG SPEC_BPM

Input of type Calibration is static calibration data, which is part of the pipeline. The science exposures must be obtained with long slit spectroscopy (low, medium or high resolution).

9.6.2 Input Parameters for visir_spc_phot

The recognized recipe options are those of recipe **visir_spc_obs** (see section 9.4 on page 38) and additionally this option, described in subsection 9.2 on page 33: off.

9.6.3 Products from visir_spc_phot

Successful completion of this recipe will create the FITS-files

File name	Product Category (PRO CATG)	Explanation
visir_spc_phot_tab.fits	SPC_PHOT_TAB	the spectral table.
visir_spc_phot.fits	SPC_PHOT_COMBINED	the combined image (the 2D spectrum).
visir_spc_phot_weight.fits	SPC_PHOT_WEIGHT	the weight map.

See 9.10.6, 9.10.1 and 9.10.3 on page 46 for a description of these products.

9.6.4 QC Parameters from visir_spc_phot

This recipe generates the Quality Control parameters

```
QC BACKGD MEAN
QC CAPA
QC EXPTIME
QC SENS MEAN
QC SENS STDEV
QC STARNAME
```

and writes them in the FITS header of its products. See appendix B on page 63 for their definition.

9.7 visir_spc_wcal_ech

The VISIR pipeline recipe *visir_spc_wcal_ech* reduces echelle spectroscopy half-cycle frames in the same way as *visir_spc_wcal*.

9.7.1 Input files for visir_spc_wcal_ech

The input Set-Of-Frames shall specify files with DO categories:

DO Category	Type	Explanation
SPEC_CAL_HRG_WCAL	Raw Frame	Calibration or Science Exposures (at least one pair)
SPEC_CAL_LINES	Calibration	Atmospheric Transmission
SPEC_CAL_QEFF	Calibration	Detector Quantum-Efficiency
BPM	Calibration	Optional bad pixel map with PRO.CATG SPEC_BPM

Input of type Calibration is static calibration data, which is part of the pipeline. The science exposures must be obtained with echelle-spectroscopy.

9.7.2 Input Parameters for visir_spc_wcal_ech

The recognized recipe options are those of recipe **visir_spc_wcal** (see section 9.4 on page 38) and additionally this option:

orderoffset	`integer`, **0**	Echelle order offset. The offset is relative to the main order. The allowed range of offsets depend on the selected echelle-mode and covers 4 or 5 orders, e.g. -2,-1,0,1,2. If the main order is e.g. 8 an order offset of +1 will cause the recipe to base the data reduction on order 9. With a positive order offset the central wavelength becomes smaller while for a negative order offset the central wavelength becomes larger.

Recipe **visir_spc_wcal_ech** has a default value of 0.0 for the four options for distortion correction:

Parameter	Possible Values (with **default**)
slit_skew	`float`, **0.0**
spectrum_skew	`float`, **0.0**
vert_arc	`float`, **0.0**
hori_arc	`float`, **0.0**

9.7.3 Products from visir_spc_wcal_ech

Successful completion of this recipe will create the FITS-file

File name	Product Category (PRO CATG)	Explanation
visir_spc_wcal_ech_spectrum_tab.fits	SPC_WCAL_HRG_TAB	the spectral table.

See 9.10.4 on page 46 for a description of this product.

9.7.4 QC Parameters from visir_spc_wcal_ech

This recipe generates the same Quality Control parameters as those of recipe visir_spc_wcal, see subsubsection 9.4.4 on page 39.

9.8 visir_spc_obs_ech

The VISIR pipeline recipe *visir_spc_obs_ech* reduces echelle spectroscopy data in the same way as *visir_spc_obs*.

9.8.1 Input files for visir_spc_obs_ech

The input Set-Of-Frames shall specify files with DO categories:

DO Category	Type	Explanation
SPEC_OBS_HRG	Raw Frame	Science Exposures (at least one pair)
SPEC_CAL_LINES	Calibration	Atmospheric Transmission
SPEC_CAL_QEFF	Calibration	Detector Quantum-Efficiency
BPM	Calibration	Optional bad pixel map with PRO.CATG SPEC_BPM

Input of type Calibration is static calibration data, which is part of the pipeline. The science exposures must be obtained with echelle-spectroscopy.

9.8.2 Input Parameters for visir_spc_obs_ech

The recognized recipe options are those of recipe **visir_spc_obs** (see section 9.5 on page 40) and additionally this option, described in subsection 9.7 on page 42: orderoffset.

Just like recipe **visir_spc_wcal_ech**, recipe **visir_spc_obs_ech** has a default value of 0.0 for the four options for distortion correction:

Parameter	Possible Values (with **default**)
slit_skew	`float`, **0.0**
spectrum_skew	`float`, **0.0**
vert_arc	`float`, **0.0**
hori_arc	`float`, **0.0**

9.8.3 Products from visir_spc_obs_ech

Successful completion of this recipe will create the FITS-files

File name	Product Category (PRO CATG)	Explanation
visir_spc_obs_ech_spectrum_tab.fits	SPC_OBS_HRG_TAB	the spectral table.
visir_spc_obs_ech.fits	SPC_OBS_HRG_COMBINED	the combined image (the 2D spectrum).
visir_spc_obs_ech_weight.fits	SPC_OBS_HRG_WEIGHT	the weight map.

See 9.10.5, 9.10.1 and 9.10.3 on page 46 for a description of these products.

9.8.4 QC Parameters from visir_spc_obs_ech

This recipe generates the same Quality Control parameters as those of recipe visir_spc_wcal, see subsubsection 9.4.4 on page 39.

9.9 visir_spc_phot_ech

The VISIR pipeline recipe *visir_spc_phot_ech* reduces echelle spectroscopy data in the same way as *visir_spc_phot*.

9.9.1 Input files for visir_spc_phot_ech

The input Set-Of-Frames shall specify files with DO categories:

DO Category	Type	Explanation
SPEC_CAL_PHOT_HRG	Raw Frame	Calibration Exposures (at least one pair)

SPEC_CAL_LINES	Calibration	Atmospheric Transmission
SPEC_CAL_QEFF	Calibration	Detector Quantum-Efficiency
SPEC_STD_CATALOG	Calibration	Catalogue of spectroscopy standard stars
BPM	Calibration	Optional bad pixel map with PRO.CATG SPEC_BPM

Input of type Calibration is static calibration data, which is part of the pipeline. The science exposures must be obtained with echelle-spectroscopy.

9.9.2 Input Parameters for visir_spc_phot_ech

The recognized recipe options are those of recipe **visir_spc_phot** (see section 9.6 on page 41) and additionally this option, described in subsection 9.7 on page 42: orderoffset.

Just like recipe **visir_spc_wcal_ech**, recipe **visir_spc_obs_ech** has a default value of 0.0 for the four options for distortion correction:

Parameter	Possible Values (with **default**)
slit_skew	`float`, **0.0**
spectrum_skew	`float`, **0.0**
vert_arc	`float`, **0.0**
hori_arc	`float`, **0.0**

9.9.3 Products from visir_spc_phot_ech

Successful completion of this recipe will create the FITS-files

File name	Product Category (PRO CATG)	Explanation
visir_spc_phot_ech_tab.fits	SPC_PHOT_HRG_TAB	the spectral table.
visir_spc_phot_ech.fits	SPC_PHOT_HRG_COMBINED	the combined image (the 2D spectrum).
visir_spc_phot_ech_weight.fits	SPC_PHOT_HRG_WEIGHT	the weight map.

See 9.10.6, 9.10.1 and 9.10.3 on the next page for a description of these products.

9.9.4 QC Parameters from visir_spc_phot_ech

This recipe generates the same Quality Control parameters as those of recipe visir_spc_phot, see subsubsection 9.6.4 on page 41.

9.10 Product Description

Even though FITS files are self-documenting, the most important ones are described in the following.

9.10.1 Combined Image

This product has one of these product categories: `IMG_OBS_COMBINED`, `IMG_PHOT_COMBINED`, `SPC_OBS_LMR_COMBINED`, `SPC_OBS_HRG_COMBINED`, `SPC_PHOT_COMBINED` or `SPC_PHOT_HRG_COMBINED`.

This is the combination of a list of chopped, nodded and optionally jittered images.

9.10.2 Contribution Map

This product has one of these product categories: `IMG_OBS_CONTRIB_MAP`, `IMG_PHOT_CONTRIB_MAP`.

This is the an image with positive integer values, that for each pixel indicate the number of pixels from the nodded images that contribute to the resulting pixel in the combined image.

9.10.3 Spectroscopic Weight Map

This product has one of these product categories: `SPC_OBS_LMR_WEIGHT`, `SPC_OBS_HRG_WEIGHT`, `SPC_PHOT_WEIGHT`, `SPC_PHOT_HRG_WEIGHT`.

The pixel values in this image indicate the weight given to each pixel in the combined image when extracting the 1D-spectrum. The weights are the same in each row of the image.

9.10.4 Spectroscopic Table for Wavelength Calibration

This product has one of these two product categories: `SPC_WCAL_LMR_TAB` and `SPC_WCAL_HRG_TAB`

This table comprises 4 columns and 256 rows, one per detector row.

WLEN The wavelength of the light detected on that detector row [m].

SPC_MODEL_PH The intensity of the model spectrum at the wavelength that the physical model predicts will be detected on that detector row. The displacement (in pixels) between the two model spectra is written in the FITS card with the key `HIERARCH ESO QC XCSHIFT` [$J radian\ m^{-3} s^{-1}$].

SPC_MODEL_XC The intensity of the model spectrum at the wavelength in column `WLEN` [$J radian\ m^{-3} s^{-1}$].

SPC_SKY The intensity of the sky spectrum at the wavelength in column `WLEN` [$ADU\ s^{-1}$].

9.10.5 Spectroscopic Tables for Science Observation

This product has one of these two product categories: `SPC_OBS_LMR_TAB` and `SPC_OBS_HRG_TAB`

This table comprises 6 columns and 256 rows, one per detector row. The first 4 columns are identical those those in the previous section, the last two are:

SPC_EXTRACTED The intensity of the extracted (object) spectrum at the wavelength in column `WLEN` [$ADU\, s^{-1}$].

SPC_ERROR The error (noise per pixel) on the extracted intensity at the wavelength in column `WLEN` [$ADU\, s^{-1}$].

9.10.6 Spectroscopic Tables for Photometric Calibration

This product has one of these two product categories: `SPC_PHOT_TAB` and `SPC_PHOT_HRG_TAB`

This table comprises 8 columns and 256 rows, one per detector row. The first 6 columns are identical those those in the previous section, the last two are:

STD_STAR_MODEL The flux of the standard star at the wavelength in column `WLEN` [$W m^{-2} m^{-1}$].

SENSITIVITY The sensitivity at the wavelength in column `WLEN` [mJy].

10 Algorithms

In this section the data reduction procedures applied by the pipeline recipes currently in use (see section 8.2 on page 30) are described in some detail.

10.1 General Algorithms

Several algorithms, such as wavelength calibration or bad pixel cleaning, are used by several recipes, and are thus described separately.

10.1.1 Bad pixel detection and cleaning

The bad pixels are detected in the Half-Cycle frames as those whose pixel value exceeds the fixed limit 65000 (this comparison is done after the offset of 32768 has been added to the pixels of the Half-Cycle frames). In spectroscopic long slit mode the cleaning of bad pixels can be avoided, since the subsequent distortion correction ignores the bad pixels. Otherwise, bad pixels are cleaned by interpolation with the neighboring pixels, (using the CPL function `cpl_detector_interpolate_rejected()`, see [15]).

10.1.2 Distortion correction

In spectroscopic long slit mode the optical distortion is known analytically. This is used to directly correct the distortion, by interpolating the distortion corrected pixel value from the source pixels. This interpolation ignores source pixels that are marked as bad. The VISIR User's manual [8] describes the optical distortion correction in greater detail. In that description, Φ is equal to slit_skew, Ψ is equal to spectrum_skew, Δ is equal to hori_arc and ϵ is equal to vert_arc. The interpolation itself is done with the CPL function `cpl_image_get_interpolated()`, see [15]).

10.1.3 Creation of nodded images

The list of nodded images are created in this way:

- From each input file a single image is created. With the recommended, default settings of the parameters p=false and g=false this image is the last plane in the file, see section 6.1 on page 24.
- The images from each pair of input files are then combined. In staring mode (no nodding, i.e. both images are from the on-source position of the chopper) the average of the two images are computed. In nodding mode one image (A) is from the on-source position of the chopper, while the other (B) is from the off-source position of the chopper. In this mode the average between A and $-B$ is computed.
- The nodded images are divided by 2*DIT*, where *DIT* is the Detector Integration Time. The factor 2 is due to the fact that the on-source and the off-source images both contribute with one whole DIT.

10.1.4 Combination of nodded images

The nodded images are jittered and are thus shifted and added to form the final combined image. The offsets stored in the FITS cards with keys `HIERARCH ESO SEQ CUMOFFSETX` and `HIERARCH ESO SEQ CUMOFFSETY` of the on-source frames are used for this. A user-defined refinement of these offsets may optionally be specified with anchor-points via the recipe options described in subsection 9.2.2 on page 34. Figure 10.1.1 on page 55 shows an example of the combination of a set of nodded images, without the user-defined refinement.

10.1.5 Wavelength Calibration

The dispersion relation is approximated well by a first degree polynomial, $\lambda(i) = i \cdot \Delta\lambda + \lambda_0, i = 1, 2, \cdots, 256$, where $\lambda(i)$ is the wavelength at the center of the i'th pixel. Thus in long slit mode, $\lambda(i), i = 128.5$ is the central wavelength.

The physical model described in *Parameters for setting the VISIR Spectrometer* (VLT-TRE-VIS-14321-5046) includes as dispersion relation a first degree polynomial $\lambda_{ph}(i) = i \cdot \Delta\lambda_{ph} + \lambda_{ph,0}$, which is used as a first guess of $\lambda(i)$. It is assumed that $\Delta\lambda_{ph}$ is a sufficiently good approximation to $\Delta\lambda$, thus only λ_0 needs to be determined.

λ_0 is determined in the following way:

1 The field direction in a Half-Cycle frame is collapsed, producing a 1-dimensional spectrum of the atmosphere. See figure 10.1.2 on page 56.

2 A model spectrum is created from a model of the atmospheric emission. See figure 10.1.2 on page 56.

3 The offset, Δi, (in pixels) that maximizes the cross-correlation between these two spectra is used to determine λ_0, since $\lambda_0 - \lambda_{ph,0} = \Delta i \cdot \Delta\lambda_{ph}$. Δi is determined to an accuracy of 0.01 pixel. See figure 10.1.3 on page 57.

The model spectrum is created as follows:

- The atmospheric emission is assumed to be equivalent to that of a black body at 253 K.
- This emission is multiplied with the emissivity of the atmosphere, see section 7.4 on page 29.
- This is smoothed by convolution with a function that is itself a convolution of two functions:

A Gaussian with

$$\sigma = \frac{w_{\mathrm{FWHM}}}{2\sqrt{2\ln 2}},$$

where the spectral FWHM is defined as $w_{\mathrm{FWHM}} = \lambda L/R$, with L being the Linear dispersion and R being the spectral Resolution as defined by the optical model. With L in unit pixel$/m$ and λ in unit m, w_{FWHM} is in unit pixel.

A Top-hat with a width of $\frac{w_{slit}}{w_{\mathrm{pfov}}}$, where w_{slit} is the value of the FITS-card with the key `HIERARCH ESO INS SLIT1 WID` which has unit arcseconds and where $w_{\mathrm{pfov}} = 0.127$ arcseconds$/$pixel is the spectral slit width.

See figure 10.1.4 on page 58 for an example of such a smoothing function.

- Added to that is the emission of the telescope itself, assumed to be equivalent to that of a black body with the temperature of the main mirror (retrieved from the FITS header) and with an emissivity of 0.12.
- The model spectrum is lastly multiplied by the detector quantum efficiency.

10.1.6 Spectrum Extraction

The spectrum is extracted from the combined image with the following optimal extraction method:

- If the spectrum is obtained in long slit mode, each row is supposed to have a mean of zero. In order to ensure this the actual mean of each row is computed and subtracted from each row.
- The standard deviation of the noise in the resulting image is estimated using an iterative σ-clipping (σ=3).
- Each flux, $F(\lambda)$, in the 1D-spectrum is computed as a weighted average of the pixels in the field direction. The weights are the same for all wavelengths, they are obtained by collapsing the spectral dimension of the 2D-spectrum and normalizing the absolute flux of this 1D-image to 1.
- A pixel with an absolute value less than σ=3 times the standard deviation of the noise is considered noise and is excluded from the weighted average. See figure 10.1.5 on page 59.

For each wavelength in the extracted 1D-spectrum the noise, $\sigma(F(\lambda))$ is computed as follows:

- The above spectrum extraction identifies for each wavelength a number of pixels in the mean-corrected image as being noise. The standard deviation of these pixels is computed.
- For each wavelength the 2-norm of the spatial weights of the non-noisy pixels is computed.
- $\sigma(F(\lambda))$ is the product of these two numbers. See figure 10.1.5 on page 59.

10.1.7 Spectral Photometric Calibration

The spectral photometric calibration is carried out as follows:

- The model flux, $F_{\text{model}}(\lambda)$, is obtained from the standard star catalog for the wavelengths in the extracted spectrum.
- The sensitivity in unit mJy at 10σ in 1 hour is then computed for each wavelength λ as

$$\frac{F_{\text{model}}(\lambda) \cdot \sigma(F(\lambda)) \cdot 10 \cdot \sqrt{t/3600s}}{F(\lambda)},$$

  where $F(\lambda)$ and $\sigma(F(\lambda))$ is the extracted intensity and its error estimate at wavelength λ (see subsection 10.1.6), and where the exposure time t is DIT $\cdot$ NDIT $\cdot$ NFILES $\cdot$ NCHOP $\cdot$ 2, with the factor 2 due to the Half-Cycle chopping.

10.1.8 Imaging Photometric Calibration

The imaging photometric calibration is carried out as follows:

- The calibration is carried out on the combined image, once for each of the N_b beams ($N_b = 3$ or 4 depending on the value of the FITS card with key `ESO SEQ CHOPNOD DIR`) using a circle of radius $R = 20$ around the center of the standard star in the image.
- The background flux, $F_{i,\mathrm{bg}}$ is estimated as the median intensity of the pixels that are between R and $R + 10$ pixels away from the star center, for $i = 1, 2, ..., N_b$.
- The flux of the star, $F_i(r)$ is computed as the flux within r pixels from the star corrected for $F_{i,\mathrm{bg}}$, for $r = 1, 2, 3, ..., R$ and $i = 1, 2, ..., N_b$.
- The error on $F_i(r)$ is estimated as $\sigma(F_i(r)) = \sigma(B) \cdot \sqrt{n}$, where $\sigma(B)$ is the estimated background noise, and n is the number of pixels within the circle. $\sigma(B)$ is computed on the entire combined image with 5 iterations of $\sigma = 3$ clipping, and is written in the QC parameter `QC BACKGD SIGMA`, see appendix B.
- The radius, $r_{i,\max}$ that maximizes

  $$\frac{F_i(r)}{\sigma(F_i(r))}$$

  is determined.
- The best N_b contributions are combined:

  $$F_{\mathrm{best}} = \sum_{i=1}^{N_b} F_i(r_{i,\max}),$$

  $$\sigma(F_{\mathrm{best}}) = \sqrt{\sum_{i=1}^{N_b} \sigma(F_i(r_{i,\max}))^2}.$$
- The catalog contains a model flux, F_{model}, in unit Jy.
- The sensitivity is then computed in unit mJy at 10σ in 1 hour as

  $$\frac{10^3 \cdot F_{\mathrm{model}} \cdot \sigma(F_{\mathrm{best}}) \cdot 10 \cdot \sqrt{t/3600s}}{F_{\mathrm{best}}},$$

  where the exposure time t is calculated as in subsection 10.1.7 on the preceding page.

The conversion factor is also computed as

$$\frac{\sum_{i=1}^{N_b} F_i(R)}{F_{\mathrm{model}}}.$$

10.1.9 Computation of Strehl Ratio

The computation of the Strehl ratio is carried out on the combined image in a circle with a radius $R = 3$ arcseconds (around the center of the first positive star.

The recipe assumes that the extent of the star is limited by a circle with radius $R_{\text{star}} = 2$ arcseconds.

The Strehl ratio is computed with these steps:

- The background flux, F_{bg} is estimated as the flux of the pixels located between R_{star} and R whose intensities are in the 10th percentile and not in the 90th percentile.
- The flux of the star, F_{star} is computed as the flux within R_{star} corrected for F_{bg}, and $I_{\text{star},\max}$ is the peak intensity of the star.
- The ideal Point Spread Function is computed as the inverse Fourier Transform of the ideal Optical Transfer Function, which is based on the telescope and instrument characteristics. F_{psf} is the flux of the PSF and $I_{\text{psf},\max}$ is its peak intensity.
- The Strehl ratio is then

$$\frac{I_{\text{star},\max}}{F_{\text{star}}} / \frac{I_{\text{psf},\max}}{F_{\text{psf}}}.$$

The error bound on the Strehl ratio is

$$c \cdot \pi \cdot \sigma_R \cdot w_{\text{pfov}} \cdot R^2_{\text{star}} / F_{\text{star}},$$

where $c = 0.007/0.0271$ is determined empirically, where w_{pfov} is the imaging pixel field of view (obtained from the FITS card with key `ESO INS PFOV` with unit arcseconds/pixel), and where σ_R is the estimated noise on the pixels located between R_{star} and R, (using the CPL function `cpl_flux_get_noise_ring()`, see [15]).

10.2 Recipe Algorithms

10.2.1 visir_img_ff

This is the algorithm used by this recipe:

- For each flat-field image the median (intensity) is computed.
- For each pixel on the detector, the pixel intensity is plotted against the corresponding median.
- Ideally, all pixels should have an equal gain thus the plots should be straight lines with a slope of 1. Since this is not the case, some pixels have a relative gain greater than 1 while others have a relative gain less than 1. For each pixel this relative gain is stored in the main product of the flat-field recipe.
- Pixels with a relative gain outside the range from 1/5 to 5 are flagged as bad in the produced bad-pixel map.

10.2.2 visir_img_combine

This is the algorithm used by this recipe:

- The image combination described in subsections 10.1.3 and 10.1.4 on page 49.

10.2.3 visir_img_phot

This is the algorithm used by this recipe:

- The image combination described in subsections 10.1.3 and 10.1.4 on page 49.
- The photometric calibration in imaging described in subsection 10.1.8 on page 51.

This recipe also computes the Strehl ratio, see subsection 10.1.9 on the preceding page.

10.2.4 visir_spc_wcal

This is the algorithm used by this recipe:

- The bad pixel detection described in 10.1.1 on page 48.
- The distortion correction described in 10.1.2 on page 48.
- The wavelength calibration described in 10.1.5 on page 49.

10.2.5 visir_spc_obs

This is the algorithm used by this recipe:

- The steps described for visir_spc_wcal in the section 10.2.4.
- The image combination described in subsections 10.1.3 and 10.1.4 on page 49.
- The spectrum extraction described in subsection 10.1.6 on page 50.

10.2.6 visir_spc_phot

This is the algorithm used by this recipe:

- The steps described for visir_spc_obs in the section 10.2.5.
- The photometric calibration described in subsection 10.1.7 on page 50.

10.2.7 visir_spc_wcal_ech

This is the algorithm used by this recipe:

- The extraction of the relevant order from the echelle, i.e. per default the main order or else on the order specified with the orderoffset recipe option.
- The steps described for visir_spc_wcal in subsection 10.2.4 on the preceding page.

10.2.8 visir_spc_obs_ech

This is the algorithm used by this recipe:

- The extraction of the relevant order from the echelle, i.e. per default the main order or else on the order specified with the orderoffset recipe option.
- The steps described for visir_spc_obs in subsection 10.2.5 on the page before.

10.2.9 visir_spc_phot_ech

This is the algorithm used by this recipe:

- The extraction of the relevant order from the echelle, i.e. per default the main order or else on the order specified with the orderoffset recipe option.
- The steps described for visir_spc_phot in subsection 10.2.6 on the preceding page.

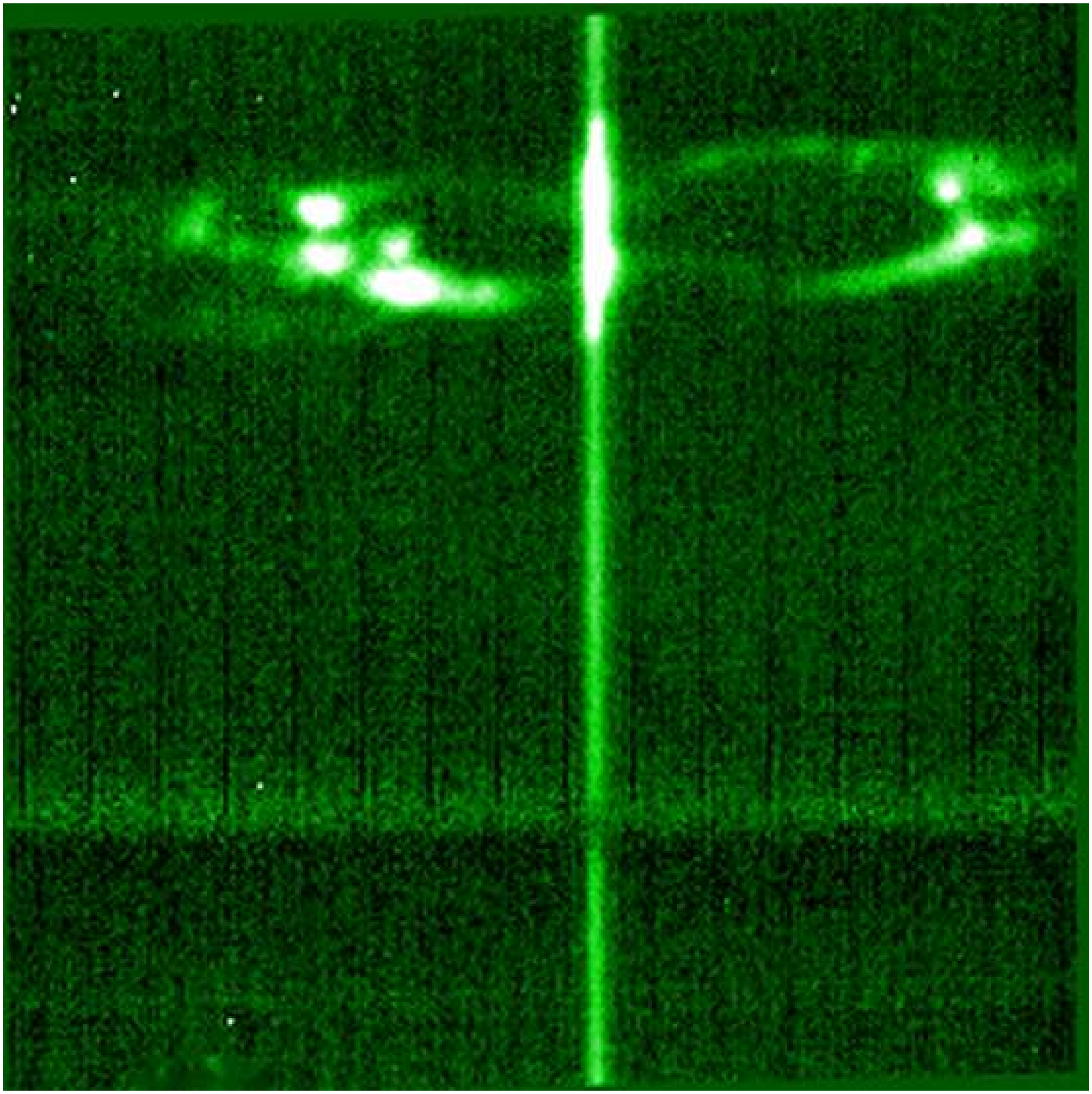

Figure 10.1.1: *An example image produced by combining a set of nodded images using the recipe visir_img_combine (the shades of green have been added afterwards).*

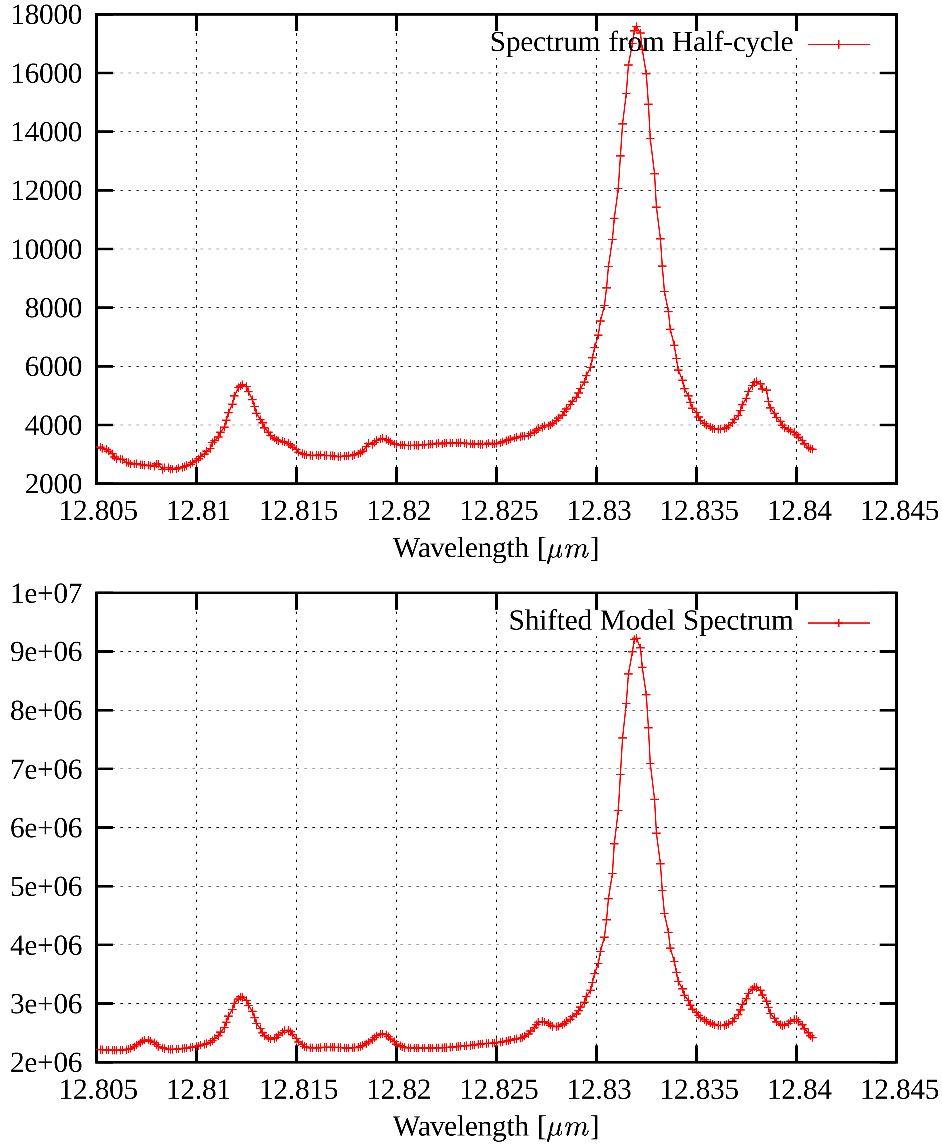

Figure 10.1.2: Example of Atmospheric spectrum from a 1/2-cycle frame and the corresponding, shifted model spectrum that maximizes the cross-correlation with that spectrum.

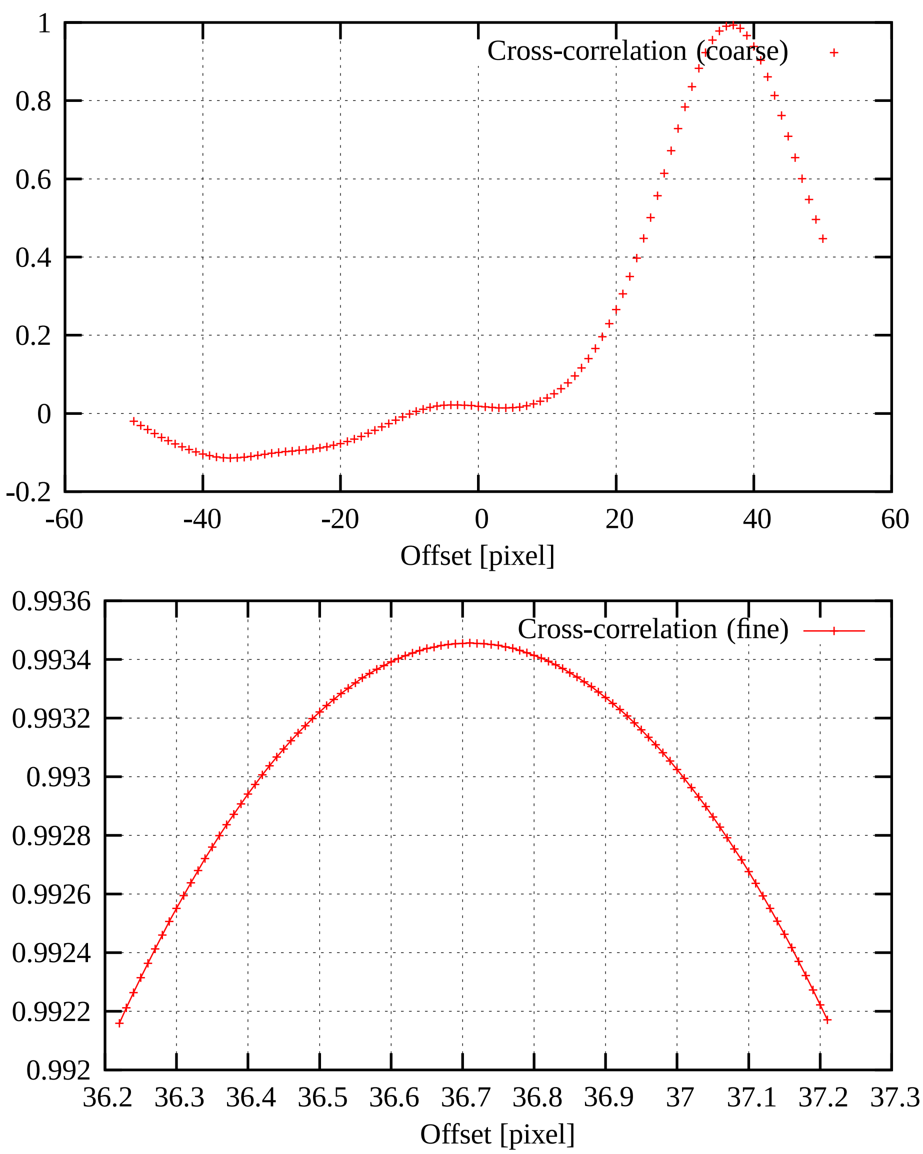

Figure 10.1.3: The cross-correlation (coarse and fine) as a function of pixel-shift.

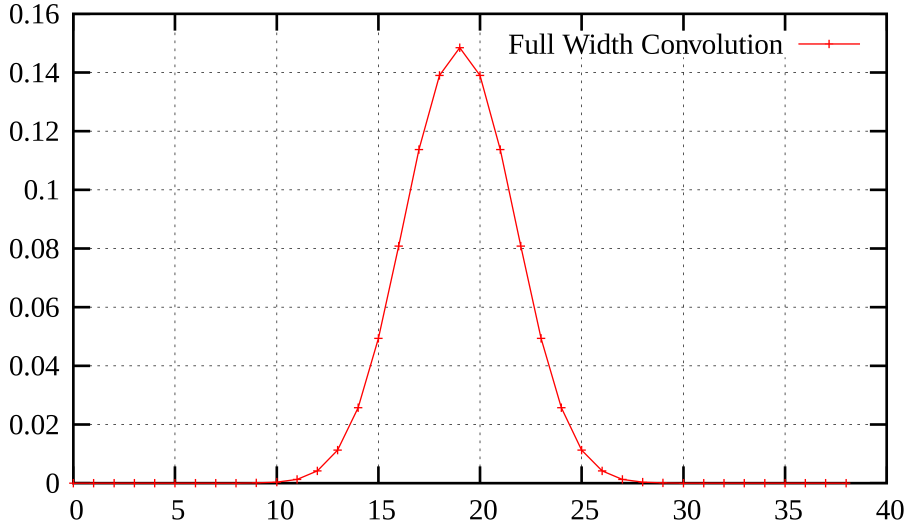

Figure 10.1.4: The symmetric convolution profile used to smooth the model spectrum. The area under the profile is 1. The instrument settings for this example is $\lambda_{central} = 12.818\mu m$ in High Resolution Long Slit mode, with $w_{slit} = 5.9$pixel and $w_{\mathrm{FWHM}} = 5.25$pixel.

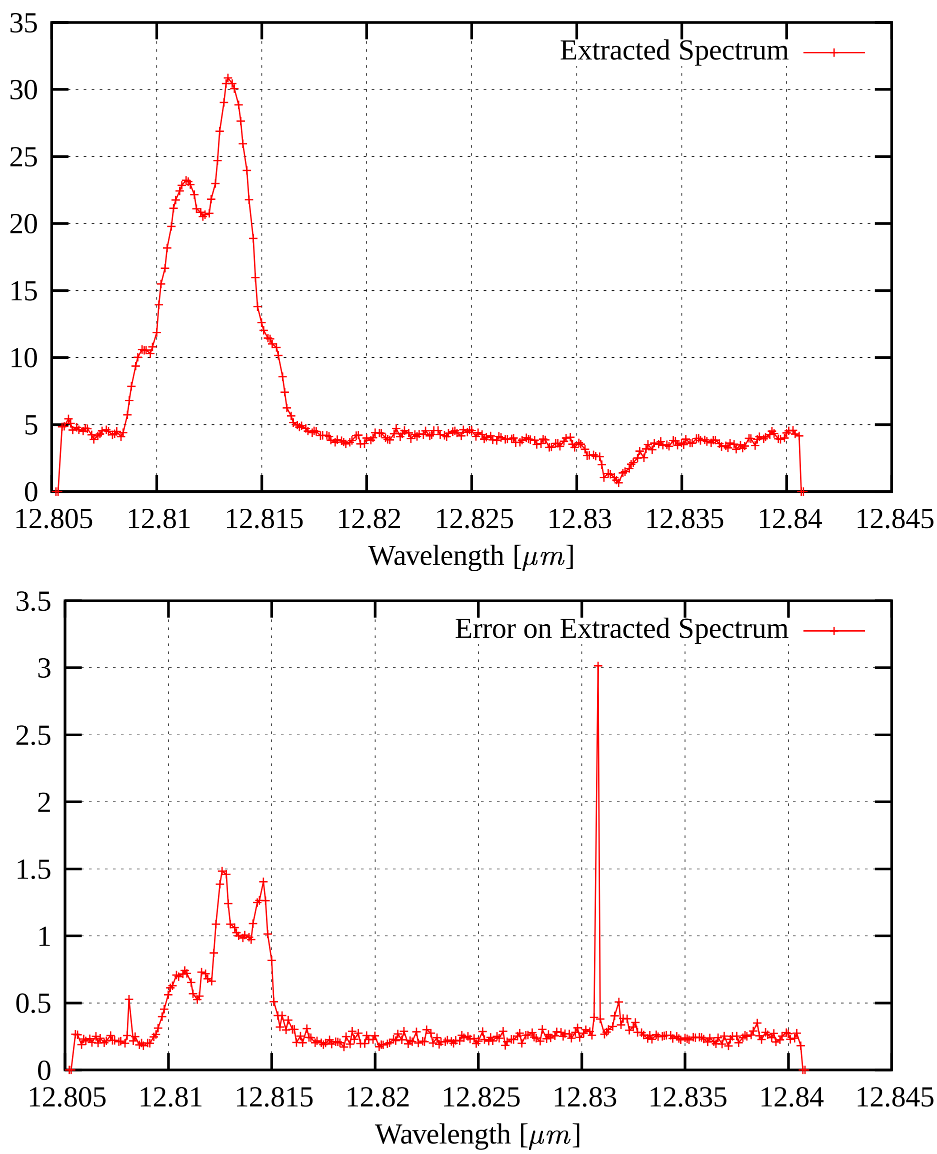

Figure 10.1.5: Example of an extracted spectrum and its error.

A Installation

This chapter gives generic instructions on how to obtain, build and install the VISIR pipeline version 1.3.7. Even if this chapter is kept as up-to-date as much as possible, it may not be fully applicable to a particular release. This might especially happen for patch releases. One is therefore advised to read the installation instructions delivered with the VISIR pipeline distribution kit. These release-specific instructions can be found in the file `README` located in the top-level directory of the unpacked VISIR pipeline source tree. The supported platforms are listed in section A.1. It is recommended reading through section A.2.3 on the facing page before starting the installation.

A bundled version of the VISIR pipeline with all the required components is available from `http://www.eso.org/pipelines`.

A.1 Supported platforms

The VISIR pipeline version 1.3.7 is verified and supported on the VLT target platforms:

- Linux Scientific Linux 4.1 (Xeon or Athlon), using gcc v. 3.3.4
- Linux Red Hat 9 (Xeon or Athlon), using gcc v. 3.2.2, 3.3, 3.4.
- Sun Solaris 2.8 (SPARC), using gcc 2.95.2, 3.3.

The usage of the GNU build tools should allow to build and run the VISIR pipeline on a variety of platforms. As such, the VISIR pipeline version 1.3.7 is known to build and to produce correct output with esorex on these platforms:

- Linux Ubuntu 5.10 (AMD64), using gcc v. 3.4.5
- Linux Fedora Core 3 (P-Mobile), using gcc v. 3.4.4
- Linux SuSE 9.3 (AMD64), using gcc v. 3.3.5
- Linux Mandrake 9.0 (P3), using gcc v. 3.2
- Linux Mandrake 8.0 (P3), using gcc v. 2.96
- Mac Darwin 7.9.0 (G4), using gcc 3.3
- Sun Solaris 2.6, using gcc 2.8.1 or native cc (WorkShop 6 v. C 5.1)

In addition, the VISIR pipeline version 1.3.7 builds on HP-UX 11.00 (11.00 using native cc or gcc 3.2, 3.3), but this platform is not supported because there are currently no front-end applications available for HP-UX. The pipeline recipes can thus only be executed on HP-UX in the limited fashion described in `visir-kit-1.3.7/visir-1.3.7/tests/README`.

A.2 Building the VISIR pipeline

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

A.2.1 Requirements

To compile and install the VISIR pipeline one needs:

- one of the C compilers listed above,
- a version of the `tar` file-archiving program,
- the GNU software: `gzip`, `perl`, `make`.

On Linux-i386 and Solaris-SPARC systems the presence of a Java Runtime Environment (JRE) version 1.4.1 will additionally allow the usage of Gasgano (part of this distribution).

A.2.2 Downloading the VISIR pipeline distribution

The VISIR pipeline (version 1.3.7) distribution is available as a gzip'ed tar-file, visir-kit-1.3.7.tar.gz from the ESO ftp server via a link from `http://www.eso.org/pipelines`.

A.2.3 Compiling and installing the VISIR pipeline

It is recommended to read through this section before starting with the installation.

1. Unpack the VISIR pipeline kit in a directory where a directory named `visir-kit-1.3.7` does not already exist, using f.ex.

   ```
   $ gzip -dc visir-kit-1.3.7.tar.gz | tar -xf -
   ```

   at the system prompt. This will create a directory called `visir-kit-1.3.7` containing the pipeline kit.

2. Change directory into `visir-kit-1.3.7`.

3. Run the installation script `./install_pipeline`.

4. The script will ask two questions:

   ```
   Where should I install the software packages ?
   ```

   with a suggested directory. In this directory the various sub-directories bin/, lib/, include/ etc. will be used for the software installation. Press <return> to accept the suggestion or enter an alternative and press <return>.

```
Where should I install the pipeline calibration files ?
```

with the previously accepted directory as a suggestion. In this directory the calibration files will be installed. Press <return> to accept the suggestion or enter an alternative and press <return>.

5. The script will then install all components and configure the front-end applications. It will finish with a list of installed pipeline recipes and a few suggested esorex commands for testing the installation. `visir-kit-1.3.7/products` contains existing recipe output from the suggested commands.

After the installation has been completed the directory `visir-kit-1.3.7` tree is no longer needed and can be removed.

A.3 Configuring the pipeline recipe front-end applications

For additional configuration of the front-end applications, please refer to their documentation, available at `http://www.eso.org/cpl/esorex.html` and `http://www.eso.org/gasgano` respectively.

B QC Parameters

This appendix describes the QC Parameters created by the VISIR pipeline:

```
Parameter Name:    QC BACKGD MEAN
Class:             header|qc-log
Context:           process
Type:              double
Value Format:      %e
Unit:              ADU
Comment Field:     Background level from Half-Cycle frames.
Description:       Background level from Half-Cycle frames. This number
                   does not include the offset correction.

Parameter Name:    QC BACKGD SIGMA
Class:             header|qc-log
Context:           process
Type:              double
Value Format:      %e
Unit:              ADU
Comment Field:     Background noise.
Description:       Background noise determined with 5 iterations of sigma=3
                   clipping.

Parameter Name:    QC CAPA
Class:             header|qc-log
Context:           process
Type:              string
Value Format:      %s
Unit:
Comment Field:     The pixel capacity (large, small or problem)
Description:       The pixel capacity (large, small or problem) based on
                   DET VOLT1 DCTA9 and DET VOLT1 DCTB9 (in imaging)
                   DET VOLT2 DCTA9 and DET VOLT2 DCTB9 (in spectroscopy).
                   If the mean of DCTA9 and DCTB9 is less than 1: small
                   If the mean of DCTA9 and DCTB9 exceeds 4.5: large,
                   otherwise problem.

Parameter Name:    QC CONVER
Class:             header|qc-log
Context:           process
Type:              double
Value Format:      %e
Unit:              ADU/Jy
```

```
Comment Field:     The conversion factor in imaging
Description:       The conversion factor in imaging

Parameter Name:    QC EXPTIME
Class:             header|qc-log
Context:           process
Type:              double
Value Format:      %e
Unit:              seconds
Comment Field:     Exposure time
Description:       Exposure time

Parameter Name:    QC FILTER
Class:             header|qc-log
Context:           process
Type:              string
Value Format:      %20s
Unit:
Comment Field:     The filter used to observe the star.
Description:       The filter used to observe the star.

Parameter Name:    QC FLUXSNR
Class:             header|qc-log
Context:           process
Type:              double
Value Format:      %e
Unit:              ADU
Comment Field:     Star flux obtained for the best SNR ratio.
Description:       Star flux obtained for the best SNR ratio.

Parameter Name:    QC FLUXSNR NOISE
Class:             header|qc-log
Context:           process
Type:              double
Value Format:      %e
Unit:              ADU
Comment Field:     Noise obtained for the best SNR ratio.
Description:       Noise obtained for the best SNR ratio.

Parameter Name:    QC FLUXTOT
Class:             header|qc-log
Context:           process
Type:              double
Value Format:      %e
```

```
Unit:               ADU
Comment Field:      Total flux of the star.
Description:        Total flux of the star.

Parameter Name:     QC FWHMX NEG1
Class:              header|qc-log
Context:            process
Type:               double
Value Format:       %e
Unit:               pixels
Comment Field:      FWHM in x of the first negative star
Description:        The Full Width at Half Maximum in x of
                    the first negative star

Parameter Name:     QC FWHMX NEG2
Class:              header|qc-log
Context:            process
Type:               double
Value Format:       %e
Unit:               pixels
Comment Field:      FWHM in x of the second negative star
Description:        The Full Width at Half Maximum in x of
                    the second negative star

Parameter Name:     QC FWHMX POS1
Class:              header|qc-log
Context:            process
Type:               double
Value Format:       %e
Unit:               pixels
Comment Field:      FWHM in x of the first positive star
Description:        The Full Width at Half Maximum in x of
                    the first positive star

Parameter Name:     QC FWHMX POS2
Class:              header|qc-log
Context:            process
Type:               double
Value Format:       %e
Unit:               pixels
Comment Field:      FWHM in x of the second positive star
Description:        The Full Width at Half Maximum in x of
                    the second positive star
```

```
Parameter Name:    QC FWHMY NEG1
Class:             header|qc-log
Context:           process
Type:              double
Value Format:      %e
Unit:              pixels
Comment Field:     FWHM in y of the first negative star
Description:       The Full Width at Half Maximum in y of
                   the first negative star

Parameter Name:    QC FWHMY NEG2
Class:             header|qc-log
Context:           process
Type:              double
Value Format:      %e
Unit:              pixels
Comment Field:     FWHM in y of the second negative star
Description:       The Full Width at Half Maximum in y of
                   the second negative star

Parameter Name:    QC FWHMY POS1
Class:             header|qc-log
Context:           process
Type:              double
Value Format:      %e
Unit:              pixels
Comment Field:     FWHM in y of the first positive star
Description:       The Full Width at Half Maximum in y of
                   the first positive star

Parameter Name:    QC FWHMY POS2
Class:             header|qc-log
Context:           process
Type:              double
Value Format:      %e
Unit:              pixels
Comment Field:     FWHM in y of the second positive star
Description:       The Full Width at Half Maximum in y of
                   the second positive star

Parameter Name:    QC JYVAL
Class:             header|qc-log
Context:           process
Type:              double
```

```
Value Format:      %e
Unit:              Jansky
Comment Field:     Jansky value for a given star in a given band from a
                   catalog.
Description:       Jansky value for a given star in a given band from a
                   catalog.

Parameter Name:    QC NBBADPIX
Class:             header|qc-log
Context:           process
Type:              integer
Value Format:      %d
Unit:              pixels
Comment Field:     Number of bad pixels
Description:       Number of bad pixels, which per default is the number
                   of pixels whose (offset-corrected) intensity exceeds
                   65000.

Parameter Name:    QC PHDEGREE
Class:             header|qc-log
Context:           process
Type:              integer
Value Format:      %d
Unit:
Comment Field:     The degree of the model dispersion polynomial
Description:       The degree of the dispersion polynomial from the
                   physical model. It is currently 1.

Parameter Name:    QC PHDISPX0
Class:             header|qc-log
Context:           process
Type:              double
Value Format:      %e
Unit:              meter
Comment Field:     wavelength = PHDISPX0 + i * PHDISPX1, i=1,2,...
Description:       The constant term of the dispersion polynomial from
                   the physical model, wavelength = f(pixel).

Parameter Name:    QC PHDISPX1
Class:             header|qc-log
Context:           process
Type:              double
Value Format:      %e
Unit:              meter/pixel
```

```
Comment Field:     wavelength = PHDISPX0 + i * PHDISPX1, i=1,2,...
Description:       The linear term of the dispersion polynomial from
                   the physical model, wavelength = f(pixel).

Parameter Name:    QC SENSIT
Class:             header|qc-log
Context:           process
Type:              double
Value Format:      %e
Unit:              mJy/10sigma/1hour
Comment Field:     The sensitivity in imaging
Description:       The sensitivity in imaging

Parameter Name:    QC SENS MEAN
Class:             header|qc-log
Context:           process
Type:              double
Value Format:      %e
Unit:              mJy
Comment Field:     Mean of the spectroscopic sensitivities
Description:       Mean of the spectroscopic sensitivities over the whole
                   spectral range

Parameter Name:    QC SENS STDEV
Class:             header|qc-log
Context:           process
Type:              double
Value Format:      %e
Unit:              mJy
Comment Field:     Standard deviation of the spectroscopic sensitivities
Description:       Standard deviation of the spectroscopic sensitivities
                   over the whole spectral range

Parameter Name:    QC STARNAME
Class:             header|qc-log
Context:           process
Type:              string
Value Format:      %30s
Unit:
Comment Field:     Standard star name
Description:       Standard star name

Parameter Name:    QC STREHL
Class:             header|qc-log
```

```
Context:             process
Type:                double
Value Format:        %e
Unit:
Comment Field:       The strehl ratio
Description:         The strehl ratio

Parameter Name:      QC STREHL ERROR
Class:               header|qc-log
Context:             process
Type:                double
Value Format:        %e
Unit:
Comment Field:       The error bound on the strehl ratio
Description:         The error bound on the strehl ratio

Parameter Name:      QC XC
Class:               header|qc-log
Context:             process
Type:                double
Value Format:        %e
Unit:
Comment Field:       The cross-correlation, ranging from 0 to 1.
Description:         The cross-correlation between the observed sky spectrum
                     and a model spectrum that has been shifted such that it
                     has maximal cross-correlation with the observed sky
                     spectrum. Range from 0 to 1.

Parameter Name:      QC XCDEGREE
Class:               header|qc-log
Context:             process
Type:                integer
Value Format:        %d
Unit:
Comment Field:       The degree of the calibration dispersion polynomial
Description:         The degree of the dispersion polynomial from the
                     cross-correlation. It is currently 1.

Parameter Name:      QC XCDISPX0
Class:               header|qc-log
Context:             process
Type:                double
Value Format:        %e
Unit:                meter
```

```
Comment Field:     wavelength = XCDISPX0 + i * XCDISPX1, i=1,2,...
Description:       The constant term of the dispersion polynomial from
                   the cross-correlation, wavelength = f(pixel).

Parameter Name:    QC XCDISPX1
Class:             header|qc-log
Context:           process
Type:              double
Value Format:      %e
Unit:              meter/pixel
Comment Field:     wavelength = XCDISPX0 + i * XCDISPX1, i=1,2,...
Description:       The linear term of the dispersion polynomial from
                   the cross-correlation, wavelength = f(pixel). It
                   is currently equal to PHDISPX1.

Parameter Name:    QC XCENTROI
Class:             header|qc-log
Context:           process
Type:              double
Value Format:      %e
Unit:              pixel
Comment Field:     The x-centroid of the spectrums brightest object
Description:       The location (centroid) in the field-direction of
                   the brightest object of the spectrum

Parameter Name:    QC XCSHIFT
Class:             header|qc-log
Context:           process
Type:              double
Value Format:      %e
Unit:              pixel
Comment Field:     The shift in pixels of the model spectrum
Description:       The shift in pixels of the model spectrum that
                   maximizes the cross-correlation between the observed
                   sky spectrum and the model spectrum. A positive number
                   means that the FITS- headers WLEN is too large.
                   The range is bound by the detector size, -256 to 256

Parameter Name:    QC XCWLEN
Class:             header|qc-log
Context:           process
Type:              double
Value Format:      %e
Unit:              meter
```

```
Comment Field:     The calibrated Central Wavelength
Description:       The actual Central Wavelength (at pixel 128.5) as
                   determined by the wavelength calibration.

Parameter Name:    QC XFWHM
Class:             header|qc-log
Context:           process
Type:              double
Value Format:      %e
Unit:              pixel
Comment Field:     The Full Width at Half Maximum of the object at XCENTROI
Description:       The Full Width at Half Maximum of the brightest object
                   of the spectrum, located at XCENTROI
```