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

OmegaCAM Pipeline User Manual

VLT-MAN-ESO-19500-XXXX

Issue 0.5

Date 2008-01-28

Prepared:	ESO OmegaCAM Pipeline Team	2006-01-11	
	Name	Date	Signature
Approved:	P.Ballester		
	Name	Date	Signature
Released:	M. Peron		
	Name	Date	Signature

This page was intentionally left blank

Contents

1 Introduction

1.1 Purpose

The OmegaCAM pipeline is a subsystem of the *VLT Data Flow System* (DFS). Its target user is ESO *Data Flow Operations* (DFO) in the generation of master calibration data, in the reduction of scientific exposures, and in the data quality control. It should also serve as a quick look tool for *Paranal Science Operations* (PSO). Additionally, the OmegaCAM 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 OmegaCAM data reduction sequence with the OmegaCAM pipeline.

This manual is a complete description of the data reduction recipes implemented by the the OmegaCAM pipeline, reflecting the status of the OmegaCAM pipeline as of 25.January.2008 (version 0.5.0).

1.2 Acknowledgements

The software package on which the OmegaCAM pipeline is based was in large parts developed by the OmegaCAM Consortium, and it is still the foundation of the current OmegaCAM pipeline.

This release benefits also from the feedback provided by the OmegaCAM SV team and OmegaCAM instrument operations team. In particular we would like to thank ....
....

1.3 Scope

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

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

Additional information on CFITSIO, the Common Pipeline Library (CPL) and ESOREX can be found respectively at [4], [5], [6]. The Gasgano tool is described in [14]. A description of the instrument is in [7]. The OmegaCAM instrument user manual is in [8] while results of Science Verifications (SV) are at [9].

1.4 Reference documents

[1] OmegaCAM Pipeline Users' Manual VLT-MAN-ESO-19500-3600
http://www.eso.org/projects/dfs/dfs-shared/web/vlt/vlt-instrument-pipelines.html
[2] Current pipeline status
http://www.eso.org/observing/dfo/quality/pipeline-status.html
[3] ESO-Data Flow Operation home page http://www.eso.org/observing/dfo/quality/
[4] CFITSIO home page http://heasarc.nasa.gov/fitsio/fitsio.html
[5] CPL home page http://www.eso.org/cpl
[6] ESOREX home page http://www.eso.org/cpl/esorex.html
[7] OmegaCAM home page http://www.eso.org/instruments/ocam/
[8] VLT OmegaCAM User Manual VLT-MAN-ESO-14700-3517
http://www.eso.org/instruments/ocam/usermanual.html
[9] OmegaCAM SV home page http://www.eso.org/science/vltsv/ocamsv/
[10] VLT Data Flow System Specifications for Pipeline and Quality Control
VLT-SPE-ESO-19600-1233
[11] DFS Pipeline & Quality Control – User Manual VLT-MAN-ESO-19500-1619
[12] ESO DICB – Data Interface Control Document GEN-SPE-ESO-00000-0794
[13] Common Pipeline Library User Manual VLT-MAN-ESO-19500-2720
[14] Gasgano User's Manual VLT-PRO-ESO-19000-1932
[15] OmegaCAM Calibration Plan VST-PLA-OCM-23100-3090
[16] OmegaCAM Users and Programmers Manual VST-MAN-OCM-23100-3126

2 Overview

In collaboration with instrument consortia, the Pipeline Systems Department (PSD) of the Software Development 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 OmegaCAM jitter stacks; bias-corrected, flat-fielded images). The accuracy of the science products is limited by the quality of the available master calibration products and by the algorithmic implementation of the pipelines themselves. In particular, adopted automatic reduction strategies may not be suitable or optimal for all scientific goals.

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

ESO offers two front-end applications for launching pipeline recipes, *Gasgano* [14] and *EsoRex*, both included in the pipeline distribution (see Appendix A, page 50). These applications can also be downloaded separately from `http://www.eso.org/gasgano` and `http://www.eso.org/cpl/esorex.html`. An illustrated introduction to Gasgano is provided in the "Quick Start" Section of this manual (see page 12).

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

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

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

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

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

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

3 OmegaCAM Instrument Description

OmegaCAM has been developed under ESO contract by the OmegaCAM Consortium, headed by the NOVA (*Nederlandse Onderzoekschool Voor Astronomie*). The instrument has been made available to the community and started operations in Paranal on July 1^{st}, 2007.

In this chapter a brief description of the OmegaCAM instrument is given. A more complete documentation can be found in the OmegaCAM User Manual, downloadable from `http://www.eso.org/instruments/ocam/`.

3.1 Instrument overview

The primary goal of OmegaCAM at the VST is to provide faint and/or rare objects for VLT spectroscopic follow-up. In addition it will offer a powerful survey capability per se in a large variety of astrophysical domains. Examples are 1) search for trans-neptunian objects; 2) micro-lensing surveys in the Galactic bulge region; 3) search for distant clusters of galaxies.

OmegaCAM is designed around a 16k x 16k mosaic of 2k x 4k thin (high q.e.) CCDs. Two of ESO's FIERA systems will be used as the controllers and permit the entire mosaic to be read out in about 45s. The useful wavelength range is 330 to 1000 nm. The total field at the Cassegrain focus of the 2.6-m VST is 1°x 1°, with a one-pixel sampling of 0.21″. A retractable Atmospheric Dispersion Compensator is available for observations at blue and ultraviolet wavelengths and/or large air mass (e.g. in the Magellanic Clouds). Up to 12 filters can be mounted simultaneously. Two auxiliary CCDs each (of the same type as the scientific ones) are controlled by a third FIERA and employed for autoguiding and for wavefront curvature sensing, respectively, with feedback loops to the tracking and active optics systems of the VST. The instrument will be used 100% in Service Mode.

The primary filter set of OmegaCAM will be a set of Sloan u', g', r', i' and z' filters. In addition, there will be Johnson B and V filters for stellar work and for cross-calibrating the photometric systems, a Strömgren v filter, an Hα filter consisting of 4 segments with redshifts of up to 10,000 $km.s^{-1}$, and a segmented ugri filter for efficient photometric monitoring of the sky.

OmegaCAM has four main observing modes for which dedicated templates have been developed. The modes are:

- Dither mode has offsets matching the size of the maximum gaps of the CCDs in the focal plane, which are $\approx$ 380 pixels ($\approx$ 80 ″). It will be operated with N=5 pointings on the sky as the default value. Dither mode is optimized for maximum sky coverage.

- Jitter mode has offsets matching the smallest gaps of the CCDs, which are $\approx$ 5 pixels. Jitter mode is optimized for observations requiring maximum homogeneity of the context map and in observations for which the wide gaps are not so critical, but require a well-mapped smoothly varying PSF.

- Stare mode has one fixed pointing position, but the same position of the sky may be re-observed multiple times. This mode is the main workhorse for monitoring the instrument.

- SSO or Solar System Objects supports the data taking of Solar System objects with non-siderial tracking of the telescope. The DFS pipeline does not support the processing of these data.

- Mosaic mode which aims to image areas of the sky larger than 1 °of diameter. This mode is not supported by the DFS pipeline.

An observing strategy uses one or more of the basic observing modes. It also defines a number of additional instructions for scheduling of the observations. The OmegaCAM observing strategies are:

- Standard strategy which consists of a single observation.
- Deep strategy for deep integrations, possibly taken at selected atmospheric conditions over several nights.
- Freq strategy which frequently monitors the same field on time scales ranging from minutes to months.
- Mosaic strategy maps areas of the sky which are larger than 1 de°.

Figure 3.1.0: The OmegaCAM instrument.

4 Quick start

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

4.1 OmegaCAM pipeline recipes

The current OmegaCAM pipeline is based on a set of 8 stand-alone recipes involved in the data reduction cascade:

ocam_readnoise to measure the CCD read noise as a standard health check.

ocam_qcheck to generates a quick check on the detector responsivity for one chip.

ocam_dark to measure CCD dark current and the rate of cosmic rays events.

ocam_gain to determine CCD gain and the variation with time.

ocam_mbias to create a master bias and a hot pixels map.

ocam_mdome to create a master dome flat field and a cold pixels map.

ocam_mflat to create a master twilight, a master flat and a bad pixels map.

ocam_reduce_std for standard stars field data reduction and the determination of a zeropoint.

ocam_reduce_sci for science data reduction.

4.2 An introduction to Gasgano and EsoRex

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

The process of data classification and association is known as data organisation.

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 tool *Gasgano* .

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

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

```
gasgano &
```

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

The data are hierarchically organised as preferred by the user. After each file name are shown the classification, the instrument setup id (which indicates the band), the instrument pre-optic (which indicates the camera setting), the template exposure number and the number of exposures in the template, and the value of the DPR.TYPE.

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

Frames can be selected from the main window for being processed by the appropriate recipe: on Figure 4.2.2, a three bias frames are selected and sent to the *ocam_mbias* recipe. This will open a *Gasgano* recipe execution window (see Figure 4.2.3), having all the specified files listed in its *Input Frames* panel.

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

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

Please refer to the *Gasgano User's Manual* [7] for a more complete description of the *Gasgano* interface.

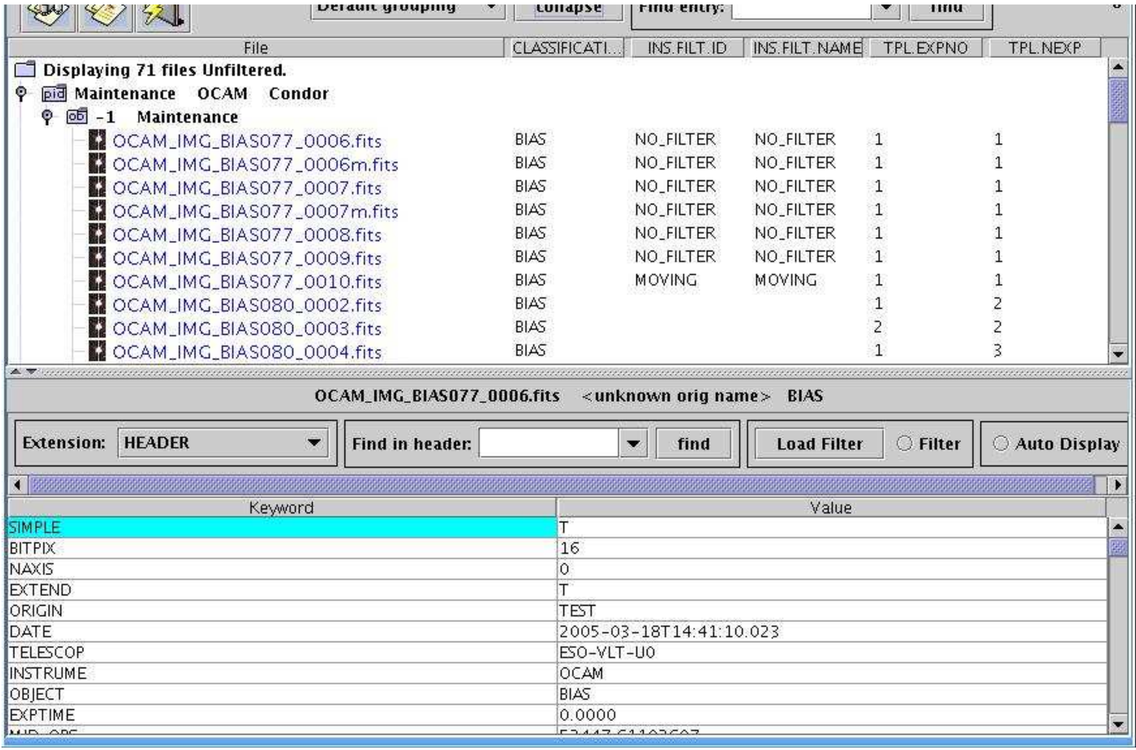

Figure 4.2.1: *The Gasgano main window.*

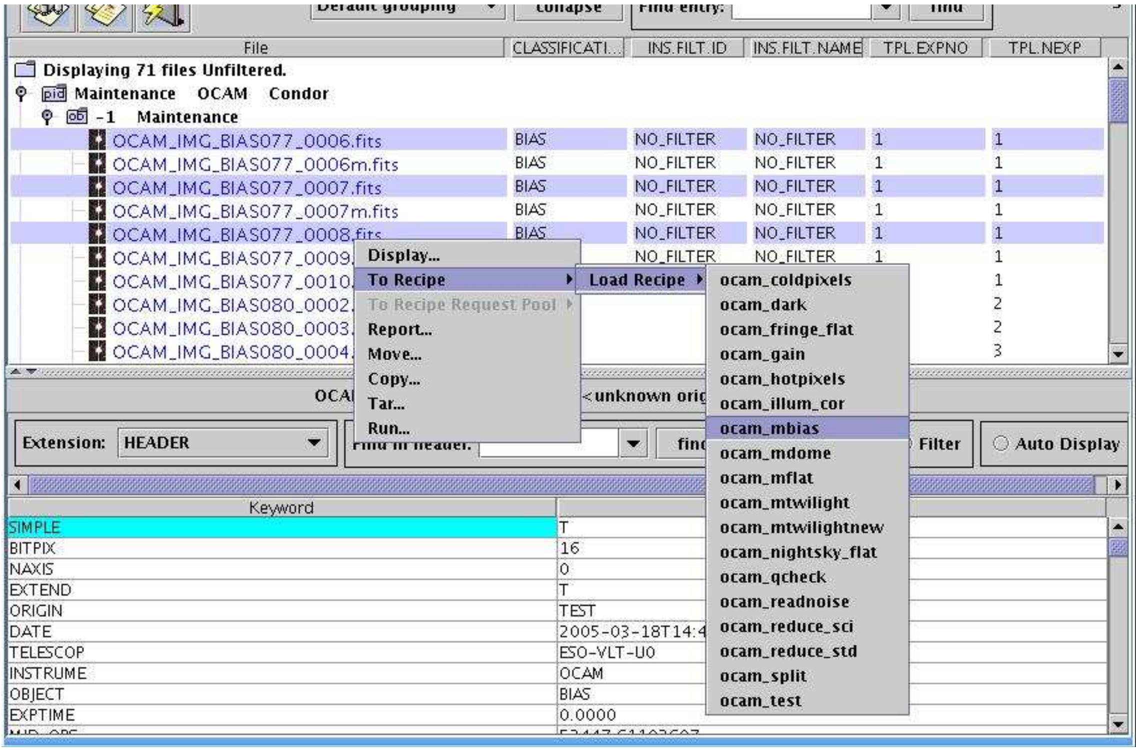

Figure 4.2.2: *Selecting files to be processed by an OmegaCAM pipeline recipe.*

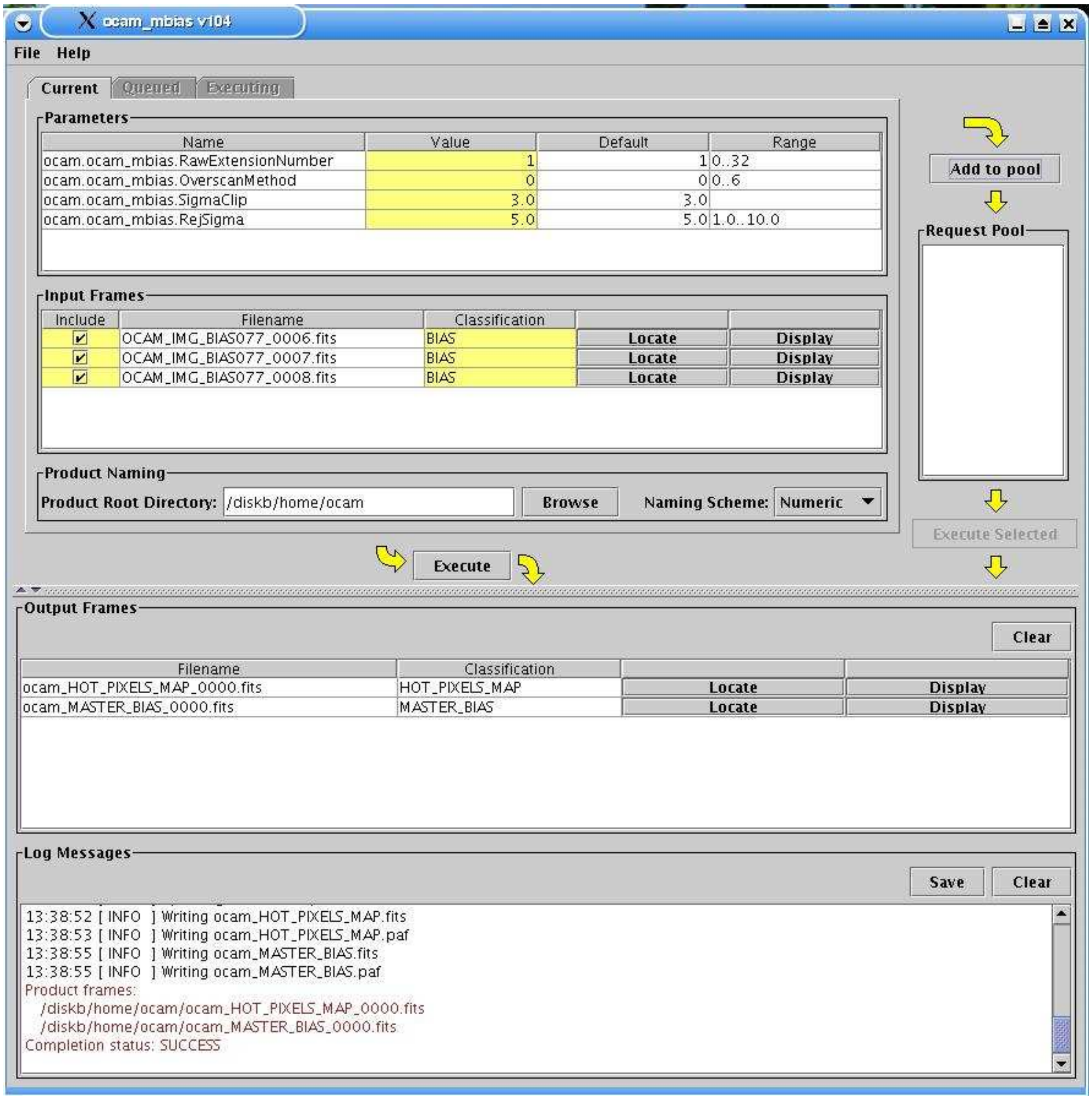

Figure 4.2.3: *The Gasgano recipe execution window.*

4.2.2 Using EsoRex

EsoRex is a command line utility for running pipeline recipes. It may be embedded by users into data reduction scripts for the automation of processing tasks. On the other side, *EsoRex* doesn't offer all the facilities available with *Gasgano*, and the user must classify and associate the data using the information contained in the FITS header keywords (see Section **??**, page **??**). 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 [1] in an ASCII file, the *set-of-frames* (SOF), that is required when launching a recipe. [2]

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

```
/file_path/OCAM_IMG_FLAT076_0007.fits TWILIGHT_FLAT
/file_path/OCAM_IMG_FLAT076_0008.fits TWILIGHT_FLAT
/file_path/OCAM_IMG_FLAT076_0009.fits TWILIGHT_FLAT
/file_path/ocam_MASTER_BIAS.fits MASTER_BIAS
/file_path/ocam_MASTER_FLAT_DOME.fits MASTER_FLAT_DOME
/file_path/ocam_HOT_PIXELS_MAP.fits HOT_PIXELS_MAP
/file_path/ocam_COLD_PIXELS_MAP.fits COLD_PIXELS_MAP
```

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

Note that the OmegaCAM pipeline recipes do not verify in any way the correctness of the classification tags specified by the user in the SOF. The reason of this lack of control is that the OmegaCAM recipes are just the DRS component of the complete pipeline running on Paranal, where the task of data classification and association is carried out by separate applications. Moreover, using *Gasgano* as an interface to the pipeline recipes will always ensure a correct classification of all the data frames, assigning the appropriate DO category to each one of them (see Section 4.2.1, page 13).

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

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

esorex [esorex_options] recipe_name [recipe_options] set_of_frames

To get more information on how to customise ESOREX (see also [7]) run the command:

esorex - -help

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

esorex - -create-config

[1]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*

[2]The set-of-frames corresponds to the *Input Frames* panel of the *Gasgano* recipe execution window (see Figure 4.2.3, page 16).

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

esorex - -recipes

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

esorex - -params recipe_name

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

esorex - -help recipe_name

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

esorex - -man-page recipe_name

Recipe configuration: Each pipeline recipe may be assigned an *EsoRex* configuration file, containing the default values of the parameters related to that recipe.[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 *ocam_mflat* has its *EsoRex* generated configuration file named `ocam_mflat.rc` , and is generated with the command:

esorex - -create-config ocam_mflat

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

```
# --oc-meth
# Overscan Correction Method: (0 -- 6)
ocam.ocam_mflat.oc_method=0
```

In this example, the parameter `ocam.ocam_mflat.oc_method` is set to the value `0`. 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 (`-oc-meth`) .

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

The command

esorex - -create-config recipe_name

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

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

esorex - -recipe-config=my_alternative_recipe_config

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

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

[3]The *EsoRex* recipe configuration file corresponds to the *Parameters* panel of the *Gasgano* recipe execution window (see Figure 4.2.3, page 16).

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

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

esorex ocam_mflat flat.sof

The recipe parameters can be modifyed 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 *ocam_mflat* recipe *oc-meth* parameter to `2`, the following should be typed:

esorex ocam_mflat - -oc-meth=2 flat.sof

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

4.3 Example of data reduction using EsoRex

A simple, typical data reduction procedure is described here.[5]

We suggest the user to organize the data per type, observed band and camera setting. Dark frames may be grouped per detector DIT, frames to compute distortion and frames to compute detector non linearities may be organized per observed band. The detector DIT is given by the value of the FITS keyword DET DIT[6]. The observed band is indicated by the value of the FITS keyword INS SETUP ID. The camera setting is indicated by the value of INS OPTI1 NAME. In the examples below we suppose the user has data acquired in band V and with the 100 mas pre-optic setting, and DIT=600. In the following examples `/path_raw/` indicates the full path to the source tree directory containing raw data.

Dark Frames: those frames are characterized by DPR.TYPE='DARK',

```
/path_raw/OCAM_IMG_DARK101_0001.fits DARK
/path_raw/OCAM_IMG_DARK101_0002.fits DARK
/path_raw/OCAM_IMG_DARK101_0003.fits DARK
```

Dome flat field frames: those frames are caracterized by DPR.TYPE='FLAT,LAMP'

```
/path_raw/OCAM.2005-02-26T20:09:50.882.fits DOME_FLAT
/path_raw/OCAM.2005-02-26T20:10:07.455.fits DOME_FLAT
/path_raw/OCAM.2005-02-26T20:10:23.047.fits DOME_FLAT
/path_raw/OCAM.2005-02-26T20:10:38.240.fits DOME_FLAT
/path_raw/OCAM.2005-02-26T20:10:56.403.fits DOME_FLAT
/path_raw/OCAM.2005-02-26T20:11:31.128.fits DOME_FLAT
/path_raw/OCAM.2005-02-26T20:11:59.183.fits DOME_FLAT
/path_raw/OCAM.2005-02-26T20:12:27.247.fits DOME_FLAT
```

[5]The procedure using *Gasgano* is conceptually identical.
[6]We omit here the prefix HIERARCH ESO

5 Known Problems

6 Instrument Data Description

OmegaCAM data can be separated into *raw* and *product* frames. Raw frames are the unprocessed output of the OmegaCAM instrument observations, while product frames are either the result of the OmegaCAM pipeline processing (as reduced frames, master calibration frames, etc.), or are outsourced (as standard stars catalogs, astrometric catalogs, 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* [7], that apply the same set of classification rules. The association of a raw frame with calibration data (*e.g.*, of a science frame with a master bias frame) can be obtained by matching the values of a different set of header keywords.

Each kind of raw frame is typically associated to a single OmegaCAM 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.

A product frame may be input to more than one OmegaCAM pipeline recipe, but it may be created by just one pipeline recipe (with the same exceptions mentioned above). In the automatic pipeline environment a product data frame alone would not trigger the execution of any recipe.

In the following all raw and product OmegaCAM data frames are listed, together with the keywords used for their classification and correct association. The indicated *DO category* is a label assigned to any data type after it has been classified, which is then used to identify the frames listed in the *Set of Frames* (see Section 4.2.2, page 17).

The OmegaCAM instrument has only one mode, which is IMAGING. Raw science frames are only distinguished by their observing mode, which can be *DIRECT*, *JITTER* or *DITHER*. Their intended use is implicitly defined by the assigned recipe.

- **Bias:**
  DO category: `BIAS`
  Processed by: `ocam_mbias`

Classification keywords:	Association keywords:	Note:
`DPR CATG = CALIB`		
`DPR TYPE = BIAS`		
`DPR TECH = IMAGE`		
`INSTRUME = OCAM`		

- **Dark current:**
  DO category: `DARK`
  Processed by: `ocam_dark`

Classification keywords:	Association keywords:	Note:
`DPR CATG = CALIB`	`INSTRUME = OCAM`	Instrument used
`DPR TYPE = DARK`		
`DPR TECH = IMAGE`		
`INSTRUME = OCAM`		

- **Quick Check:**
  DO category: LIFETEST
  Processed by: ocam_qcheck

Classification keywords:	Association keywords:	Note:
DPR CATG = CALIB		
DPR TYPE = FLAT,DOME,LIFETEST	INSTRUME = OCAM	Instrument used
DPR TECH = IMAGE		
INSTRUME = OCAM		

- **Shutter Test:**
  DO category: SHUTTER TEST
  Processed by: ocam_shutterdelay

Classification keywords:	Association keywords:	Note:
DPR CATG = CALIB		
DPR TYPE = FLAT,DOME,SHUTTERTEST		
DPR TECH = IMAGE		
INSTRUME = OCAM		

- **Read Noise:**
  DO category: READNOISE
  Processed by: ocam_readnoise

Classification keywords:	Association keywords:	Note:
DPR CATG = CALIB		
DPR TYPE = BIAS,READNOISE		
DPR TECH = IMAGE		
INSTRUME = OCAM		

- **Dome Flat Field:**
  DO category: FLAT_DOME
  Processed by: ocam_mdome

Classification keywords:	Association keywords:	Note:
DPR CATG = CALIB		
DPR TYPE = FLAT,DOME	INSTRUME = OCAM	Instrument used
DPR TECH = IMAGE	INS FILT[1-4] ID	Filter unique ID
INSTRUME = OCAM	DET WIN1 DIT1	actual subintegration time

- **Twilight flat field:**
  DO category: FLAT_SKY
  Processed by: ocam_mflat

Classification keywords:	Association keywords:	Note:
DPR CATG = CALIB		
DPR TYPE = FLAT,SKY	INSTRUME = OCAM	Instrument used

Classification keywords:	Association keywords:	Note:
DPR TECH = IMAGE	INS FILT[1-4] ID	Filter unique ID
INSTRUME = OCAM	DET WIN1 DIT1	actual subintegration time

- **Gain:**

  DO category: GAIN
  Processed by: ocam_lingain

Classification keywords:	Association keywords:	Note:
DPR CATG = CALIB		
DPR TYPE = FLAT,DOME,DETGAIN	INSTRUME = OCAM	Instrument used
DPR TECH = IMAGE	INS FILT[1-4] ID	Filter unique ID
INSTRUME = OCAM		

- **Standard stars field:**

  DO category: STANDARD
  Processed by: ocam_reduce_std

Classification keywords:	Association keywords:	Note:
DPR CATG = CALIB		
DPR TYPE = STD,ZEROPOINT	INSTRUME = OCAM	Instrument used
DPR TECH = IMAGE,DIRECT	INS FILT[1-4] ID	Filter unique ID
INSTRUME = OCAM		

  DO category: STANDARD
  Processed by: ocam_reduce_std

Classification keywords:	Association keywords:	Note:
DPR CATG = CALIB		
DPR TYPE = STD,EXTINCTION	INSTRUME = OCAM	Instrument used
DPR TECH = IMAGE,DIRECT	INS FILT[1-4] ID	Filter unique ID
TPL ID = OCAM_img_cal_monit		
INSTRUME = OCAM		

- **Scientific observation:**

  DO category: SCIENCE_DIRECT
  Processed by: ocam_reduce_sci

Classification keywords:	Association keywords:	Note:
DPR CATG = SCIENCE		
DPR TYPE = OBJECT	INSTRUME = OCAM	Instrument used
DPR TECH = IMAGE,DIRECT	INS FILT[1-4] ID	Filter unique ID
INSTRUME = OCAM		

  DO category: SCIENCE_JITTER
  Processed by: ocam_reduce_sci

Classification keywords:	Association keywords:	Note:
DPR CATG = SCIENCE		
DPR TYPE = OBJECT	INSTRUME = OCAM	Instrument used
DPR TECH = IMAGE,JITTER	INS FILT[1-4] ID	Filter unique ID
TPL ID = OCAM_img_obs_jitter		
INSTRUME = OCAM		

DO category: SCIENCE_DITHER
Processed by: ocam_reduce_sci

Classification keywords:	Association keywords:	Note:
DPR CATG = SCIENCE		
DPR TYPE = OBJECT	INSTRUME = OCAM	Instrument used
DPR TECH = IMAGE,DITHER	INS FILT[1-4] ID	Filter unique ID
TPL ID = OCAM_img_obs_dither		
INSTRUME = OCAM		

7 Static Calibration Data

In the following all static calibration frames used in the OmegaCAM pipeline are listed, together with the keywords used for their correct association. The indicated *PRO.CATG* is a label assigned to any product data type, which is then used to identify the calibration frames listed in the *Set of Frames* (see Section 4.2.2, page 17).

7.1 Monitoring Report table

This table contains the monitoring of any short term variability related to the transparency of the atmosphere (atmospheric extinction). It is a mandatory input for the recipe to reduce a standard star field..............

PRO.CATG: `MONITOR_REPORT`

7.2 Extinction Curve table

It contains a table describing the extinction curve per unit airmass as a function of wavelength............ It is a mandatory input for the recipe to reduce a standard star field................

PRO.CATG: `EXTINCTION_CURVE`

7.3 Photometric catalog

The photometric catalog contains standard stars from Stetson, Landolt and others taken by the Consortium (TBD). (see Table 7.4.1). This table contains 196949 sources in the following format.

PRO.CATG: `REFERENCE_STAR_CATALOG`

7.4 Astrometric catalog

The astrometric catalogue used to astrometrically calibrate OmegaCAM science data are the USNOA2 catalogs. The whole USNOA2 catalogue is available as a series of 100 FITS tables, accessed by means of a single master FITS table. The USNOA2 catalogue has been split into ranges of 1.5 degress in DEC. Each single FITS table contains sources in the whole range of RA and in 1.5deg of DEC. The master_usnoa2.fits table lists all these files with their RA and DEC ranges, along with the number of sources in each file. The format of the master usnoa2 table is described below in Table **??**.

PRO.CATG: `MASTER_USNOA2_CATALOGUE`

7.5 Focal Plane Coefficients

This table provides a set of polynomial coefficients to the focal plane for each OmegaCAM CCD. This table is a mandatory input for the illumination correction recipe.

PRO.CATG: ILLUM_FIT_PAR

Column name	Explanation
SeqNr	Sequence Number
origin	Origin of the observation
Name	Star identification string
Ra	RA of star
Ra_err	Error in RA of star
Dec	Dec of star
Dec_err	Error in Dec of star
Epoch	Epoch of observation
Flag	Flag
JohnsonU	U magnitude of star
JohnsonU_err	Error in U magnitude of star
JohnsonB	B magnitude of star
JohnsonB_err	Error in B magnitude of star
JohnsonV	V magnitude of star
JohnsonV_err	Error in V magnitude of star
CousinsR	R magnitude of star
CousinsR_err	Error in R magnitude of star
CousinsI	I magnitude of star
CousinsI_err	Error in I magnitude of star
SloanU	U magnitude of star
SloanU_err	Error in U magnitude of star
SloanG	G magnitude of star
SloanG_err	Error in G magnitude of star
SloanR	R magnitude of star
SloanR_err	Error in R magnitude of star
SloanI	I magnitude of star
SloanI_err	Error in I magnitude of star
SloanZ	Z magnitude of star
SloanZ_err	Error in Z magnitude of star

Table 7.3.1: *Photometric catalog entries.*

Column name	Explanation
USNOA2_CAT	Name of FITS file
RA_min	Starting RA
RA_max	Ending RA
Dec_min	Starting DEC
Dec_max	Ending DEC
N_sources	Number of sources in table

Table 7.4.1: *Master USNOA2 catalogue entries.*

8 Data Reduction

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

8.1 Data reduction overview

The calibration of science data can be divided in three steps. 1) Removing the effects of bias and differential gain; 2) relating the overall gain, and hence counts to a photometric scale; 3) relating the x,y coordinates to an astrometric reference system.

The calibration data necessary to remove the detector signature and gain variation over an image are:

- **Bias:** to subtract residual pattern in the bias offset.
- **Flatfields:** to correct for non-uniform gain.
- **Fringe maps:** to remove the fringe patterns present in images taken with certain filters.
- **Weight maps:** to determine the relative contribution of each pixel when combining images.

The main OmegaCAM data reduction problems to solve are the following.

- Correct for the detector signature: bad pixels, detector contribution to the measured signal, flat fielding, correct pixel to pixel gain variations .
- Incorporate the detection of cosmic rays and satellite tracks.
- Measure the astrometric distortions and offsets
- Apply photometric calibrations.
- Combine the images to construct a clean, maximally uniform image of the FOV of OmegaCAM.

8.2 Required input data

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

- Reference files:
  - A table with polynomial coefficients of the focal plane fit.
  - A catalog of standard stars.

  - An astrometry reference catalog (USNOA).
  - A monitoring report table of the atmosphere.
  - An extinction curve.

- Raw frames:
  - Bias frames.
  - Darks, to determine the dark current.
  - Dome and Twilight Flat fields, to determine master flats.
  - Science frames for fringing correction.
  - Science frames for Night Sky correction (called night sky flats).
  - Standard star fields to determine the photometric zero point.
  - Science frames, to finally do science.

- Calibration data products.
  - Bad pixel maps, to correct for the detector defects.
  - Master flats, to correct for different detector pixel efficiencies.
  - Master biases, to correct for the instrument bias.

Calibration data products can be generated from raw data using the pipeline recipes. Alternatively the user may use calibration products obtained from the ESO archive or from the ESO Data Flow Operation department. Calibration frames such as flats and bad pixels maps depend on the observed band. The astrometric reference catalog and the master bias do not depend on the observed band. Science data requiring a master bias need to have matching values of the FITS keyword HIERARCH ESO DET DIT.

8.3 Reduction cascade

The OmegaCAM data reduction follows the following sequence. A short description of the available recipes is given in section 4.1. In parenthesis we provide the value of the DO category corresponding to each frame.

- Run **ocam_mbias** on a set of raw biases (BIAS) to determine the master bias (MASTER_BIAS) and the hot pixels map (HOT_PIXELS_MAP).
- Run **ocam_mdome** on a set of raw dome flat fields (DOME_FLAT), a MASTER_BIAS and a HOT_PIXELS_MAP to determine the master dome flat field (MASTER_DOME_FLAT) and the cold pixels map (COLD_PIXELS_MAP).
- Run **ocam_mflat** on a set of twilight flat fields (TWILIGHT_FLAT), a MASTER_BIAS, a HOT_PIXELS_MAP, a COLD_PIXELS_MAP and a MASTER_DOME_FLAT to determine the master twilight flat field (MASTER_TWILIGHT_FLAT), a super master flat field (MASTER_FLAT) and a bad pixels map (BAD_PIXELS_MAP).
- Run **ocam_reduce_std** on a set of standard stars field (STANDARD), a MASTER_BIAS, a MASTER_FLAT, a BAD_PIXELS_MAP, a MONITOR_REPORT, a EXTINCTION_CURVE, and REFERENCE_STAR_CATALOG to reduce the standard stars field and derive a value for the zeropoint (ZEROPOINTS) and the extinction of the observations.

- Run **ocam_reduce_sci** on a set of scientific data (DIRECT, JITTER or DITHER) and a MASTER_BIAS, a MASTER_FLAT, a BAD_PIXELS_MAP, a ZEROPOINTS and optionally a ILLUM_CORR, a MASTER_FRINGE_FLAT and a MASTER_NIGHT_SKY to reduce science data.

The main data products involved in the data reduction cascade are indicated in the OmegaCAM association map shown in Figure **??**. It shows the dependencies among raw data, calibration products and recipes involved in the correction of the instrument signature and reduction of science data. Examples of sets of input frames (SOF) for each recipe are provided in section 9.

9 Pipeline Recipes Interface

We provide in this section examples of the required input data and their tags for each recipe. In the following we assume that /path_file_raw/filename_raw.fits and /path_file_cdb/filename_cdb.fits are existing FITS files (e.g. /data1/ocam/OCAM.2006-08-16T02:54:04.353.fits and /cal/ocam/cal/ZEROPOINTS.fits).

We also provide a list of the pipeline products for each recipe, indicating their default names (eventually renamed by esorex if instructed to do so), the value of the FITS keyword HIERARCH ESO PRO CATG (in short PRO.CATG) and a short description. The relevant keywords are PRO.CATG, used to classify each frame and to associate to each raw frame the proper calibration frame:

Association keyword	Information
HIERARCH ESO INS SETUP ID	band
HIERARCH ESO INS OPTI1 NAME	Pixel scale
HIERARCH ESO DET DIT	Integration time

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. Quality control parameters are also listed and explained. Those are stored in relevant pipeline products. More information on instrument quality control can be found on http://www.eso.org/qc

9.1 ocam_readnoise

As a standard health check, the CCD read noise is measured in ADU's from two bias exposures. The output for this recipe is a table containing QC parameters.

9.1.1 Input

```
/path_file_raw/OCAM.2006-08-16T01:24:53.060.fits READNOISE
/path_file_raw/OCAM.2006-08-16T01:09:22.061.fits READNOISE
```

9.1.2 Quality control

Statistics of the difference between the two images The mean and median of the resulting frame are monitored, as well as the rms scatter.These values are saved in the keywords QC.MEAN.DIFF, QC.MEDIAN.DIFF and QC.READNOISE.ADU.

9.1.3 Parameters

parameter	alias	default

ocam.ocam_readnoise.ExtensionNumber	extn	1
ocam.ocam_readnoise.NumberIter	niter	5
ocam.ocam_readnoise.RejSigma	sig-clip	5.0
ocam.ocam_readnoise.PAF	paf	TRUE

ExtensionNumber: Number of the FITS extension to load for all input raw images present in the SOF. Allowed values are 1 through 32, 1 being the default value. A 0 extension number means that all extensions will be loaded.

NumberIter: Maximum number of iterations to use when calculating statistics. Allowed values 0 – 10.

RejSigma: Sigma clipping threshold used to calculate statistics. Allowed values 1.0 – 10.0.

PAF: Boolean value to create PAF files. 1(Yes), 0(No)

9.2 ocam_gain

This recipe takes pairs of raw dome flats with equal exposure times. Each raw frame is trimmed, overscan-corrected (when appropriate) and de-biased. For each pair the sum and difference images are computed. Next the standard deviation of the difference image and the median of the sum image are measured and recorded as QC parameters.

The total number of raw dome flats has to be larger than four. For each exposure time, there has to be two different raw dome flats.

9.2.1 Input

```
/path_file_raw/OCAM.2006-08-16T01:24:53.070.fits GAIN
/path_file_raw/OCAM.2006-08-16T01:09:22.071.fits GAIN
/path_file_raw/OCAM.2006-08-16T00:53:51.072.fits GAIN
/path_file_raw/OCAM.2006-08-16T00:38:14.074.fits GAIN
/path_file_raw/OCAM.2006-08-16T00:38:15.075.fits GAIN
/path_file_raw/OCAM.2006-08-16T00:38:18.076.fits GAIN
/path_file_cdb/ocam_MASTER_BIAS.fits MASTER_BIAS
```

9.2.2 Quality control

Gain The parameter gain is calculated as the slope of the fit to the median vs. rms of the differences of the images.

9.2.3 Parameters

parameter	alias	default
ocam.ocam_gain.ExtensionNumber	extn	1
ocam.ocam_gain.OverscanMethod	oc-meth	0
ocam.ocam_gain.NumberIter	niter	5
ocam.ocam_gain.RejSigma	rej-sigma	5.0
ocam.ocam_gain.PAF	paf	TRUE

ExtensionNumber: Number of the FITS extension to load for all input raw images present in the SOF. Allowed values are 1 through 32, 1 being the default value. A 0 extension number means that all extensions will be loaded.

OverscanMethod: Detailed description is given in Section 10.2.1. Allowed values are 0 – 6.

NumberIter: Maximum number of iterations to use when calculating statistics. Allowed values 0 – 10.

RejSigma: Sigma clipping threshold used to calculate statistics. Allowed values 1.0 – 10.0.

PAF: Boolean value to create PAF files. 1(Yes), 0(No)

9.3 ocam_dark

This recipe measures the dark current and the particle event rate. It takes at least three dark exposures of one hour each and a Master Bias as input. The only product of this recipe is the mean of all input raw dark frames.

9.3.1 Input

```
/path_file_raw/OCAM.2006-08-16T02:24:53.904.fits DARK
/path_file_raw/OCAM.2006-08-16T02:09:22.905.fits DARK
/path_file_raw/OCAM.2006-08-16T02:53:51.906.fits DARK
/path_file_cdb/ocam_MASTER_BIAS.fits MASTER_BIAS
```

9.3.2 Output

default recipe file name	PRO.CATG	short description
ocam_MEAN_DARK.fits	MEAN_DARK	Mean of the input dark images

9.3.3 Quality control

Dark current It registers the dark current for each CCD in ADU/pixel/hour in the parameter QC.DARK.CURRENT.

Particle event rate The number of particles in the images are monitored by DFO. This number is registered in particles/cm^2/hour in the parameter QC.PARTICLE.RATE.

9.3.4 Parameters

parameter	alias	default
ocam.ocam_dark.ExtensionNumber	extn	1
ocam.ocam_dark.OverscanMethod	oc-meth	0
ocam.ocam_dark.RejThre	rej-thre	5.0
ocam.ocam_dark.DetThre	det-thre	6.0
ocam.ocam_dark.MaxIter	maxiter	3
ocam.ocam_dark.MaxCurrent	maxcur	5.0
ocam.ocam_dark.MaxDiffCurrent	maxdifcur	0.2
ocam.ocam_dark.PAF	paf	TRUE

The values of OCAM_BIN_PATH and OCAM_CONFIG_PATH are passed to the recipe at compile time. The recipes will look for executables and configuration files in two locations ./bin and ./config, respectively, under the working directory (**–prefix** path used for the installation of the pipeline).

ExtensionNumber: Number of the FITS extension to load for all input raw images present in the SOF. Allowed values are 1 through 32, 1 being the default value. A 0 extension number means that all extensions will be loaded.

OverscanMethod: Detailed description is given in Section 10.2.1. Allowed values are 0 – 6.

RejThre: The threshold to reject outlying pixels.

DetThre: The detection threshold for cosmic ray event.

MaxIter: The maximum number of iterations in statistics.

MaxCurrent: The maximum dark current in ADU/pixel/hour.

MaxDiffCurrent: The maximum difference of dark current between previous ??.

PAF: Boolean value to create PAF files. 1(Yes), 0(No)

9.4 ocam_mbias

This recipe creates a Master Bias for a particular chip. It trims and overscan the bias frames, average them and reject 5σ outliers. It also creates a map of the hot pixels from the resulting image.

9.4.1 Input

```
/path_file_raw/OCAM.2006-08-16T02:54:04.353.fits BIAS
/path_file_raw/OCAM.2006-08-16T02:53:37.089.fits BIAS
/path_file_raw/OCAM.2006-08-16T02:52:23.028.fits BIAS
/path_file_raw/OCAM.2006-08-16T02:51:59.774.fits BIAS
```

```
/path_file_raw/OCAM.2006-08-16T02:50:38.991.fits BIAS
/path_file_raw/OCAM.2006-08-16T02:50:11.797.fits BIAS
/path_file_raw/OCAM.2006-08-16T02:49:04.887.fits BIAS
/path_file_raw/OCAM.2006-08-16T02:48:36.792.fits BIAS
/path_file_raw/OCAM.2006-08-16T02:47:28.191.fits BIAS
/path_file_raw/OCAM.2006-08-16T02:47:07.438.fits BIAS
```

9.4.2 Output

default recipe file name	PRO.CATG	short description
ocam_MASTER_BIAS.fits	MASTER_BIAS	Master Bias image
ocam_HOT_PIXELS_MAP.fits	HOT_PIXELS_MAP	A map of the hot pixels

9.4.3 Quality control

The pipeline computes the number of hot pixels in the master bias image and the mean, median and standard deviation of the bias image pixel values.

Hot Pixels Number The computed hot pixels is given in QC.NUMBER.HOT.PIXELS.

Statistics of the master bias The following statistics are calculated: QC.MASTER.BIAS.MEAN, QC.MASTER.BIAS.M and QC.MASTER.BIAS.STDEV.

9.4.4 Parameters

parameter	alias	default
ocam.ocam_mbias.ExtensionNumber	extn	1
ocam.ocam_mbias.OverscanMethod	oc-meth	0
ocam.ocam_mbias.SigmaClip	sig-clip	3.0
ocam.ocam_mbias.RejSigma	rej-sigma	5.0
ocam.ocam_mbias.PAF	paf	TRUE

ExtensionNumber: Number of the FITS extension to load for all input raw images present in the SOF. Allowed values are 1 through 32, 1 being the default value. A 0 extension number means that all extensions will be loaded.

OverscanMethod: Detailed description is given in Section 10.2.1. Allowed values are 0 – 6.

SigmaClip: Sigma Clipping Threshold for bias images. Allowed values 1.0 – 10.0.

RejSigma: Rejection threshold for outlying pixels in Hot Pixels Map creation. Allowed values 1.0 – 10.0.

PAF: Boolean value to create PAF files. 1(Yes), 0(No)

9.5 ocam_mdome

This recipe creates a Master Dome Flat for a particular chip. The raw dome flats are averaged with a rejection threshold, which can be controlled by modifying the sigma clipping value in the input parameter (sig-clip). A map of the cold pixels is also calculated from the resulting image. Some of the input calibration frames are optional such as the Master Fringe Flat and the Master Night Sky Flat.

9.5.1 Input

```
/path_file_raw/OCAM.2006-08-16T01:24:53.070.fits DOME_FLAT
/path_file_raw/OCAM.2006-08-16T01:09:22.905.fits DOME_FLAT
/path_file_raw/OCAM.2006-08-16T00:53:51.890.fits DOME_FLAT
/path_file_raw/OCAM.2006-08-16T00:38:14.994.fits DOME_FLAT
/path_file_raw/OCAM.2006-08-16T00:38:15.998.fits DOME_FLAT
/path_file_cdb/ocam_MASTER_BIAS.fits MASTER_BIAS
/path_file_cdb/ocam_HOT_PIXELS_MAP.fits HOT_PIXELS_MAP
/path_file_cdb/ocam_MASTER_NIGHT_SKY_FLAT MASTER_NIGHT_SKY_FLAT
/path_file_cdb/ocam_MASTER_FRINGE_FLAT MASTER_FRINGE_FLAT
```

9.5.2 Output

default recipe file name	PRO.CATG	short description
ocam_FLAT_DOME.fits	MASTER_FLAT_DOME	Master dome flat field
ocam_COLD_PIXELS_MAP.fits	COLD_PIXELS_MAP	Map of cold pixels

9.5.3 Quality control

Number of Cold Pixels The number of cold pixels is calculated on the master dome flat and saved in QC.NUMBER.COLD.PIXELS.

Statistics of the master dome flat The mean, median and standard deviation of the counts in the master dome flat are monitored by DFO. These values are saved in the keywords QC.MASTER.DOME.MEAN, QC.MASTER.DOME.MEDIAN and QC.MASTER.DOME.STDEV.

9.5.4 Parameters

parameter	alias	default
ocam.ocam_mdome.ExtensionNumber	extn	1
ocam.ocam_mdome.OverscanMethod	oc-meth	0
ocam.ocam_mdome.SigmaClip	sig-clip	3.0
ocam.ocam_mdome.LowThre	low	0.96
ocam.ocam_mdome.HighThre	high	1.04

ocam.ocam_mdome.BinPath	bin-path	OCAM_BIN_PATH
ocam.ocam_mdome.SexConfig	sex-config	OCAM_CONFIG_PATH/ocam.sex
ocam.ocam_mdome.SexConv	sex-conv	OCAM_CONFIG_PATH/ocam.conv
ocam.ocam_mdome.SexParam	sex-param	OCAM_CONFIG_PATH/ocam.param
ocam.ocam_mdome.SexNnw	sex-nnw	OCAM_CONFIG_PATH/ocam.nnw
ocam.ocam_mdome.PAF	paf	TRUE

The values of OCAM_BIN_PATH and OCAM_CONFIG_PATH are passed to the recipe at compile time. The recipes will look for executables and configuration files in two locations ./bin and ./config, respectively, under the working directory (**–prefix** path used for the installation of the pipeline).

ExtensionNumber: Number of the FITS extension to load for all input raw images present in the SOF. Allowed values are 1 through 32, 1 being the default value. A 0 extension number means that all extensions will be loaded.

OverscanMethod: Detailed description is given in Section 10.2.1. Allowed values are 0 – 6.

SigmaClip: Sigma Clipping Threshold for dome images. Allowed values 1.0 – 10.0.

LowThre: Low threshold for cold pixels map. Allowed values 0.90 – 1.00.

HighThre: High threshold for cold pixels map. Allowed values 1.00 – 1.10.

BinPath: Absolute path to any external executable program.

SexConfig: Absolute pathname to Sextractor configuration file.

SexConv: Absolute pathname to Sextractor convolution mask file.

SexParam: Absolute pathname to Sextractor parameters file.

SexNnw: Absolute pathname to Sextractor neural network config file.

PAF: Boolean value to create PAF files. 1(Yes), 0(No)

9.6 ocam_mflat

This recipe is a combination of two recipes which will create a Master Twilight Flat, a Master Flat and a Map of the Bad Pixels. If there is a master dome flat in the set of frames, then a master flat will also be created, otherwise it will be duplicated from the newly created master twilight flat. It is **strongly** recommended to use a master dome flat as input to create the master flat. A combination of the two hot and cold pixels maps will provide a bad pixels map also as a product of this recipe.

9.6.1 Input

```
/path_file_raw/OCAM.2006-08-16T01:24:53.070.fits TWILIGHT_FLAT
/path_file_raw/OCAM.2006-08-16T01:09:22.905.fits TWILIGHT_FLAT
/path_file_raw/OCAM.2006-08-16T00:53:51.890.fits TWILIGHT_FLAT
/path_file_raw/OCAM.2006-08-16T00:38:14.994.fits TWILIGHT_FLAT
/path_file_raw/OCAM.2006-08-16T00:38:15.998.fits TWILIGHT_FLAT
/path_file_cdb/ocam_MASTER_BIAS.fits MASTER_BIAS
/path_file_cdb/ocam_HOT_PIXELS_MAP.fits HOT_PIXELS_MAP
/path_file_cdb/ocam_COLD_PIXELS_MAP.fits COLD_PIXELS_MAP
/path_file_cdb/ocam_MASTER_DOME_FLAT.fits MASTER_DOME_FLAT
/path_file_cdb/ocam_MASTER_NIGHT_SKY_FLAT MASTER_NIGHT_SKY_FLAT
/path_file_cdb/ocam_MASTER_FRINGE_FLAT MASTER_FRINGE_FLAT
```

9.6.2 Output

default recipe file name	PRO.CATG	short description
ocam_MASTER_FLAT_TWILIGHT.fits	MASTER_FLAT_TWILIGHT	Master twilight flat field
ocam_MASTER_FLAT.fits	MASTER_FLAT	Master flat field
ocam_BPM_MAP.fits	BAD_PIXELS_MAP	Map of all bad pixels

9.6.3 Quality control

Number of Bad Pixels QC.BAD.PIXELS.

Statistics of the master twilight flat The mean, median and standard deviation of the counts in the master twilight flat are monitored by DFO. These values are saved in the keywords QC.MASTER.TWILIGHT.MEAN, QC.MASTER.TWILIGHT.MEDIAN and QC.MASTER.TWILIGHT.STDEV.

Statistics of the master flat The mean, median and standard deviation of the counts in the master flat are monitored by DFO. These values are saved in the keywords QC.MASTER.FLAT.MEAN, QC.MASTER.FLAT.MEDIAN and QC.MASTER.FLAT.STDEV.

9.6.4 Parameters

parameter	alias	default
ocam.ocam_mflat.ExtensionNumber	extn	1
ocam.ocam_mflat.OverscanMethod	oc-meth	0
ocam.ocam_mflat.SigmaClip	sig-clip	3.0
ocam.ocam_mflat.GaussianFiltSize	gau_filt_sz	9.0
ocam.ocam_mflat.MirrorXpix	mr_xpix	75
ocam.ocam_mflat.MirrorYpix	mr_ypix	150

ocam.ocam_mflat.PAF	paf	TRUE

The values of OCAM_BIN_PATH and OCAM_CONFIG_PATH are passed to the recipe at compile time. The recipes will look for executables and configuration files in two locations ./bin and ./config, respectively, under the working directory (**–prefix** path used for the installation of the pipeline).

ExtensionNumber: Number of the FITS extension to load for all input raw images present in the SOF. Allowed values are 1 through 32, 1 being the default value. A 0 extension number means that all extensions will be loaded.

OverscanMethod: Detailed description is given in Section 10.2.1. Allowed values are 0 – 6.

SigmaClip: Sigma Clipping Threshold for twilight flat images. Allowed values 1.0 – 10.0. item [Gaussian-FiltSize:] Standard deviation of Gaussian used to filter low/high spatial frequency components. Allowed values ??

MirrorXpix: Width of X region for mirroring edges (FFT continuity). Allowed values ??

MirrorYpix: Width of Y region for mirroring edges (FFT continuity). Allowed values ??

PAF: Boolean value to create PAF files. 1(Yes), 0(No)

9.7 ocam_illum_cor

This recipe is used to derive an illumination correction frame for one particular chip and filter. The recipe always takes as input a fit of the illumination correction over the full focal plane for a given filter.

9.7.1 Input

```
/path_file_raw/OCAM.2006-08-16T01:24:53.070.fits ILLUM_FIT_PAR
```

9.7.2 Output

default recipe file name	PRO.CATG	short description
ocam_ILLUM_CORR.fits	ILLUM_CORR	Illumination Correction frame

9.7.3 Quality control

TBD

9.8 ocam_reduce_std

This recipe takes an image of a standard stars field as an input raw frame and as calibration frames it takes a master bias, a master flat, a bad pixels map (optional) a illumination correction image (optional), a master fringe flat (optional) a master night sky flat (optional) a catalog of standard stars a table with a monitoring report of the atmosphere and an extinction curve. The image is reduced as any other science frame and the stars are associated to the stars on the reference catalog to derive a zeropoint and a value for the extinction of the night.

9.8.1 Input

```
/path_file_raw/OCAM.2006-08-16T01:24:53.070.fits STD_ZEROPOINT
/path_file_cdb/ocam_MASTER_BIAS.fits MASTER_BIAS
/path_file_cdb/ocam_MASTER_FLAT.fits MASTER_FLAT
/path_file_cdb/ocam_reference_catalog.fits REFERENCE_STAR_CATALOG
/path_file_cdb/ocam_monit_report.fits MONITOR_REPORT
/path_file_cdb/ocam_extinction_curve.fits EXTINCTION_CURVE
/path_file_cdb/ocam_BAD_PIXELS_MAP.fits BAD_PIXELS_MAP
/path_file_cdb/ocam_ILLUM_CORR.fits ILLUM_CORR
/path_file_cdb/ocam_MASTER_NIGHT_SKY_FLAT.fits MASTER_NIGHT_SKY_FLAT
/path_file_cdb/ocam_MASTER_FRINGE_FLAT.fits MASTER_FRINGE_FLAT
```

9.8.2 Output

default recipe file name	PRO.CATG	short description
ocam_REDUCED_STD_STAR.fits	REDUCED_STD_STAR	Reduced standard star field image
ocam_ZEROPOINTS.fits	ZEROPOINTS	Table with zeropoints

9.8.3 Quality control

TBD

9.8.4 Parameters

parameter	alias	default
ocam.ocam_reduce_std.ExtensionNumber	extn	1
ocam.ocam_reduce_std.OverscanMethod	oc-meth	0
ocam.ocam_reduce_std.LowThreSatuPixel	lt-satu	5.0
ocam.ocam_reduce_std.HighThreSatuPixel	ht-satu	5e+04
ocam.ocam_reduce_std.SigmaFringeScaling	sig-fr-sc	5.0
ocam.ocam_reduce_std.LowThreFringe	lt-fr	1.5
ocam.ocam_reduce_std.HighThreFringe	ht-fr	5.0
ocam.ocam_reduce_std.BackRestore	backg-rest	FALSE

ocam.ocam_reduce_std.BinPath	bin-path	OCAM_BIN_PATH
ocam.ocam_reduce_std.SexConfig	sex-config	OCAM_CONFIG_PATH/ocam.sex
ocam.ocam_reduce_std.SexConv	sex-conv	OCAM_CONFIG_PATH/ocam.conv
ocam.ocam_reduce_std.SexParam	sex-param	OCAM_CONFIG_PATH/ocam.param
ocam.ocam_reduce_std.SexNnw	sex-nnw	OCAM_CONFIG_PATH/ocam.nnw
ocam.ocam_reduce_std.PrephotomConfig	prephotom	OCAM_CONFIG_PATH/prephotom.conf
ocam.ocam_reduce_std.PAF	paf	TRUE

The values of OCAM_BIN_PATH and OCAM_CONFIG_PATH are passed to the recipe at compile time. The recipes will look for executables and configuration files in two locations ./bin and ./config, respectively, under the working directory (**–prefix** path used for the installation of the pipeline).

ExtensionNumber: Number of the FITS extension to load for all input raw images present in the SOF. Allowed values are 1 through 32, 1 being the default value. A 0 extension number means that all extensions will be loaded.

OverscanMethod: Detailed description is given in Section 10.2.1. Allowed values are 0 – 6.

PAF: Boolean value to create PAF files. 1(Yes), 0(No)

9.9 ocam_reduce_sci

This recipe takes a single SCIENCE frame from a DIRECT mode template or a series of frames from a JITTER or a DITHER mode. In all three modes, the images are reduced and calibrated photometrically and astrometrically. In JITTER and DITHER modes, the recipe will attempt to create a new Master Night Sky Flat and if the filter requires, also a new Master Fringe Flat. The product of the DIRECT mode is a regridded science image and the product of the JITTER and DITHER modes are the stacked images of all the input frames, regridded to a common target grid. A weight map is also saved as a product.

9.9.1 Input for DIRECT mode

```
/path_file_raw/OCAM.2006-08-16T01:24:53.070.fits SCIENCE
/path_file_cdb/ocam_MASTER_BIAS.fits MASTER_BIAS
/path_file_cdb/ocam_MASTER_NIGHT_SKY_FLAT.fits MASTER_NIGHT_SKY_FLAT
/path_file_cdb/ocam_MASTER_FRINGE_FLAT.fits MASTER_FRINGE_FLAT
/path_file_cdb/ocam_MASTER_FLAT.fits MASTER_FLAT
/path_file_cdb/ocam_BAD_PIXELS_MAP.fits BAD_PIXELS_MAP
/path_file_cdb/ocam_ILLUM_CORR.fits ILLUM_CORR
/path_file_cdb/ocam_ZEROPOINTS.fits ZEROPOINTS
```

9.9.2 Output for DIRECT mode

default recipe file name	PRO.CATG	short description
ocam_CALIBRATED_SCIENCE.fits	CALIBRATED_SCIENCE	Photometrically and astrometrically calibrated science
ocam_WEIGHT_MAP.fits	WEIGHT_MAP	Weight map of science image

9.9.3 Quality control for DIRECT mode

TBD

9.9.4 Input for JITTER or DITHER modes

```
/path_file_raw/OCAM.2006-08-16T01:24:53.070.fits JITTER (or DITHER)
/path_file_raw/OCAM.2006-08-16T01:24:53.071.fits JITTER
/path_file_raw/OCAM.2006-08-16T01:24:53.072.fits JITTER
/path_file_raw/OCAM.2006-08-16T01:24:53.073.fits JITTER
/path_file_raw/OCAM.2006-08-16T01:24:53.074.fits JITTER
/path_file_cdb/ocam_MASTER_BIAS.fits MASTER_BIAS
/path_file_cdb/ocam_MASTER_NIGHT_SKY_FLAT.fits MASTER_NIGHT_SKY_FLAT
/path_file_cdb/ocam_MASTER_FRINGE_FLAT.fits MASTER_FRINGE_FLAT
/path_file_cdb/ocam_MASTER_FLAT.fits MASTER_FLAT
/path_file_cdb/ocam_BAD_PIXELS_MAP.fits BAD_PIXELS_MAP
/path_file_cdb/ocam_ILLUM_CORR.fits ILLUM_CORR
/path_file_cdb/ocam_ZEROPOINTS.fits ZEROPOINTS
```

9.9.5 Output for JITTER or DITHER modes

default recipe file name	PRO.CATG	short description
ocam_COADDED_JITTER_SCIENCE.fits	COADDED_JITTER_SCIENCE	stacked and calibrated image
ocam_WEIGHT_COADDED_JITTER_SCIENCE.fits	WEIGHT_MAP	Weight map of stacked image

9.9.6 Quality control for JITTER or DITHER modes

TBD

9.9.7 Common Parameters for SCIENCE, JITTER and DITHER modes

parameter	alias	default

ocam.ocam_reduce_sci.ExtensionNumber	extn	1
ocam.ocam_reduce_sci.OverscanMethod	oc-meth	0
ocam.ocam_reduce_sci.SaveAllProducts	saveall	FALSE
ocam.ocam_reduce_sci.LowThreSatuPixel	lt-satu	5.0
ocam.ocam_reduce_sci.HighThreSatuPixel	ht-satu	5e+04
ocam.ocam_reduce_sci.DetectionThreSatellite	det-sate	5.0
ocam.ocam_reduce_sci.HoughThreshold	hough-thre	1e+03
ocam.ocam_reduce_sci.Nskyflat	nsky	FALSE
ocam.ocam_reduce_sci.FringeCor	fringecor	FALSE
ocam.ocam_reduce_sci.SigmaFringeScaling	sig-fr-sc	5.0
ocam.ocam_reduce_sci.LowThreFringe	lt-fr	1.5
ocam.ocam_reduce_sci.HighThreFringe	ht-fr	5.0
ocam.ocam_reduce_sci.BackRestore	backg-rest	FALSE
ocam.ocam_reduce_sci.BinPath	bin-path	OCAM_BIN_PATH
ocam.ocam_reduce_sci.SexConfig	sex-config	OCAM_CONFIG_PATH/ocam.sex
ocam.ocam_reduce_sci.SexConv	sex-conv	OCAM_CONFIG_PATH/ocam.conv
ocam.ocam_reduce_sci.SexParam	sex-param	OCAM_CONFIG_PATH/ocam.param
ocam.ocam_reduce_sci.SexNnw	sex-nnw	OCAM_CONFIG_PATH/ocam.nnw
ocam.ocam_reduce_sci.SexCosmic	sex-cosmic	OCAM_CONFIG_PATH/ocam.cosmic.sex
ocam.ocam_reduce_sci.SexCosmicParam	sex-cosmic-param	OCAM_CONFIG_PATH/ocam.cosmic.param
ocam.ocam_reduce_sci.SexCosmicFilt	sex-cosfilt	OCAM_CONFIG_PATH/cosmic.ret
ocam.ocam_reduce_sci.SexCosmicDet	sex-cosmic-det	5.0
ocam.ocam_reduce_sci.PrephotomConfig	prephotom	OCAM_CONFIG_PATH/prephotom.conf
ocam.ocam_reduce_sci.PreastromConfig	preastrom	OCAM_CONFIG_PATH/preastrom.conf
ocam.ocam_reduce_sci.AssociateConfig	associate	OCAM_CONFIG_PATH/associate.conf
ocam.ocam_reduce_sci.MakesscConfig	make_ssc	OCAM_CONFIG_PATH/make_ssc.conf
ocam.ocam_reduce_sci.AstromConfig	astrom	OCAM_CONFIG_PATH/astrom.conf
ocam.ocam_reduce_sci.MakedistConfig	make_dist	OCAM_CONFIG_PATH/make_distort.conf
ocam.ocam_reduce_sci.AstromDetThre	astrom-thre	10.0
ocam.ocam_reduce_sci.USNOADIR	USNOADIR	/home/ocam
ocam.ocam_reduce_sci.SwarpConfig	swarpconf	OCAM_CONFIG_PATH/swarp.conf
ocam.ocam_reduce_sci.PAF	paf	TRUE

The values of OCAM_BIN_PATH and OCAM_CONFIG_PATH are passed to the recipe at compile time. The recipes will look for executables and configuration files in two locations ./bin and ./config, respectively, under the working directory (**–prefix** path used for the installation of the pipeline).

ExtensionNumber: Number of the FITS extension to load for all input raw images present in the SOF. Allowed values are 1 through 32, 1 being the default value. A 0 extension number means that all extensions will be loaded.

OverscanMethod: Detailed description is given in Section 10.2.1. Allowed values are 0 – 6.

SaveAllProducts: Boolean value to save all intermediate files in disk. 1(TRUE), 0(FALSE).

LowThreSatuPixel: Low threshold for calculating the saturated pixels map.

HighThreSatuPixel: High threshold for calculating the saturated pixels map.

DetectionThreSatellite: Minimum SNR for pixels to contribute to Hough map.

HoughThreshold: Threshold for satellite tracks in Hough image.

Nskyflat: Boolean value to create master night sky flat. 1(TRUE), 0(FALSE).

FringeCor: Boolean value to correct for fringing. 1(TRUE), 0(FALSE).

SigmaFringeScaling: Sigma threshold in image data for scaling estimate (FringeCor=TRUE).

LowThreFringe: Lower bound of fringes to include in scaling (FringeCor=TRUE).

HighThreFringe: Higher bound of fringes to include in scaling (FringeCor=TRUE).

BackRestore: Boolean value to restore background after illumination correction. 1(TRUE), 0(FALSE).

BinPath: Absolute path to any external executable program.

SexConfig: Absolute pathname to Sextractor configuration file.

SexConv: Absolute pathname to Sextractor convolution mask file.

SexParam: Absolute pathname to Sextractor parameters file.

SexNnw: Absolute pathname to Sextractor neural network config file.

SexCosmic: Absolute pathname to Sextractor cosmic detection mode config file.

SexCosmicParam: Absolute path to Sextractor cosmic parameters file.

SexCosmicFilt: Absolute path to Sextractor filter mask for cosmic detection mode.

SexCosmicDet: Detection threshold for detecting cosmic rays in Sextractor.

PrephotomConfig: Absolute path to LDAC prephotom config file.

PreastromConfig: Absolute path to LDAC preastrom config file.

AssociateConfig: Absolute path to LDAC associate config file.

MakesscConfig: Absolute path to LDAC make_ssc config file.

AstromConfig: Absolute path to LDAC astrom config file.

MakedistConfig: Absolute path to LDAC make_distort config file.

AstromDetThre: Astrometric Detection threshold.

USNOADIR.: Full pathname of the USNOA catalogs directory.

SwarpConfig: Absolute path to SWARP config file.

PAF: Boolean value to create PAF files. 1(Yes), 0(No)

10 Algorithms-TBD

In this section the data reduction procedures applied by all pipeline recipes currently in use (see Section 4.1) are described in some detail. Common algorithms such as as cosmic rays removal or bad pixel cleaning, are described separately.

10.1 General Algorithms-TBD

10.1.1 Bad pixel cleaning-TBD

Bad pixel cleaning consists of replacing any bad pixel value with an estimate based on a set of surrounding *good* pixel values. This operation is generally applied to science product frames, having little or no sense when applied to master calibration products. Nevertheless all the OmegaCAM pipeline recipes allow bad pixel cleaning on any product frame, for debug reasons or for any other purpose that may be appropriate.

The routine currently used by the OmegaCAM pipeline recipes performs a bad pixel correction based on the content of a given bad pixel table (`CCD_TABLE`). If the number of bad pixels is more than 15% of the total number of CCD pixels, the correction is not applied.

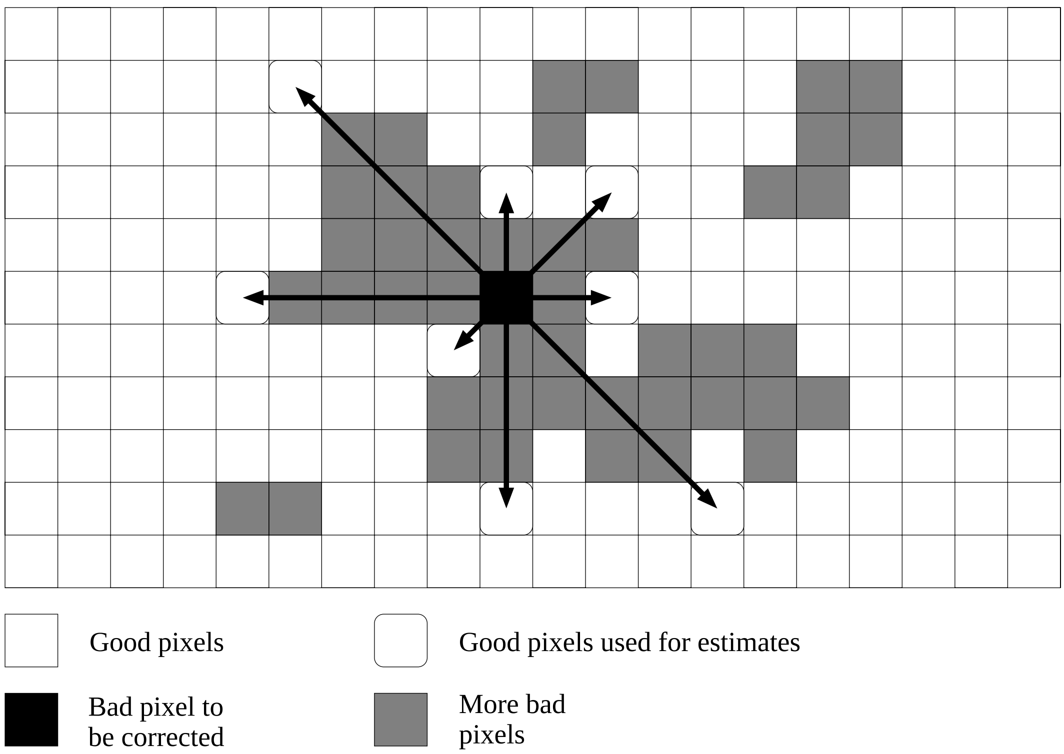

Figure 10.1.1: *Good pixels to be used in the estimate of a given bad pixel are searched along the indicated directions.*

Any bad pixel is given a new value, computed as follow: the closest good pixels along the vertical, the horizontal, and the two diagonal directions are found (see Figure 10.1.1). This search is done within a distance of 100 pixels. If no good pixel is found within this range, then the bad pixel is not corrected. All the good pixels found within range will be used to compute the bad pixel value.

For each of the four fundamental directions, an estimate of the considered bad pixel can generally be obtained. If two good pixel values are available for a given direction, the estimate is their linear interpolation at the bad pixel position. If just one good pixel value is available for a given direction, then the value itself will be the estimate of the bad pixel value. No estimate can be obtained from directions where no good pixel was found.

If the available number of estimates is greater than 1, the bad pixel value is taken as the median of the estimates (defining the median of an even number of values as the mean of the two central values), otherwise it is simply set to the single estimate available.

10.1.2 Cosmic rays removal-TBD

The core of a cosmic rays removal procedure is to determine what is and what is *not* a cosmic ray. The algorithm used for this purpose by the OmegaCAM pipeline recipes is t

Initially all pixels having an abnormal excess with respect to the local noise level are flagged as possibly belonging to a cosmic ray event (that typically would involve a group of contiguous pixels). A candidate is selected at any pixel (x, y) having a value $F(x, y)$ exceeding a given threshold. This threshold, expressed in units of noise sigma, is specified by the recipe parameter *CosmicsThreshold*. A value 4.0 gives typically good results. The theoretical noise $N(x, y)$ of the image at any given pixel position (x, y) is estimated in ADU as

$$N(x, y) = \sqrt{r^2 + \frac{M(x, y)}{g}}$$

where $M(x, y)$ is the median value of the 8 pixels surrounding the (x, y) position and r is the read-out-noise, both in ADU, and g is the gain factor in e^-/ADU. Then a pixel (x, y) is taken as a cosmic ray candidate if

$$F(x, y) > k \cdot N(x, y)$$

with k the number of noise sigmas used in thresholding.

After this step is completed, all the groups of contiguous cosmic rays candidates are identified. For each group, the position of its maximum pixel value is determined, and the mean $\overline{F}_8$ of its 8 surrounding pixels is computed. A given group will be taken as a cosmic ray event if it fulfils the condition

$$F_{max} - S > R \cdot (\overline{F}_8 - S)$$

where F_{max} is the maximum pixel value within the considered group, S the fundamental background level (corresponding to the sky level in imaging science exposures), and R is a shape parameter for discriminating between objects and cosmic rays. The ratio R is specified by the recipe parameter *CosmicsRatio*. A value of 2.0 gives typically good results.

Once all the pixels affected by cosmic ray events has been located and listed in a cosmic ray events table, their values are interpolated using the procedure described in Section 10.1.1. If a bad pixel table is also given to a recipe, then the bad pixels are avoided in the interpolation procedure.

10.2 Recipes Algorithms

10.2.1 Bias subtraction

The signal in raw scientific frames contains a component that is due to a bias current. This component shows up as an offset to the signal. The bias-offset has the following characteristics: i) the bias level grows to its asymptotic level in the first few hundred lines, and ii) the bias level depends on the total signal in a given line. Therefore, an initial bias correction–the overscan correction (see a list below with all overscan correction methods), is applied by averaging the overscan pixels for each line, and subtracting this value from that line.

- **Overscan correction methods:**
  - **0:** apply no overscan correction on the images (default method).
  - **1:** use median of the prescan regions in the x-direction.
  - **2:** use median of the overscan regions in the x-direction.
  - **3:** use median of the prescan regions in the y-direction.
  - **4:** use median of the overscan regions in the y-direction.
  - **5:** use the per-row value of the prescan regions in the x-direction.
  - **6:** use the per-row values of the overscan regions in the x-direction.

In addition, the bias offset exhibits a residual pattern, which is measured by the master bias frame. To construct the master bias a series of 10 zero-second bias exposures is overscan-corrected, and then averaged, rejecting 5σ outliers (σ = dispersion of the 10 bias exposures of individual pixels), due to particle hits during read-out.

As the readout noise dominates the rms scatter in the bias frames, while the shot noise of the sky background dominates the rms scatter on the sky images, which is nominally much larger than the readout noise, it is sufficient to characterize the bias value at individual pixels with an accuracy of (readout noise/$\sqrt{10}$).

The resulting master bias frame (`MASTER_BIAS`) is used for the correction of all frames. The master bias is subtracted from the raw data frame, whose overscan regions are then trimmed away using the same overscan method applied to construct the master bias.

10.2.2 Flat field correction

A master flat field (`MASTER_FLAT`) is a suitable combination of one or more of the available flat fields (`MASTER_DOME_FLAT, MASTER_TWILIGHT_FLAT, MASTER_NIGHT_SKYFLAT`). A method whereby the master dome flat is used to measure the pixel-to-pixel (small-scale) variation, and the master twilight flat is used to measure the large scale variation, would provide a first-order approximation of the master flat field. These spatial frequencies are separated using a Fourier technique. Night-sky flats are created from raw science or standard data, flat-fielded with this master flat field and can be used to improve the quality of the master flat. The next paragraphs will explain how the different flat fields are created in the OmegaCAM pipeline.

Master dome flats are obtained through an average with sigma rejection procedure on a stack of raw dome flats, intended to reduce photon shot noise and remove cosmic rays. In the case of a master twilight flat, the intended purpose is to remove any contamination (including stars) present on individual raw twilight flats and reduce photon shot noise.

The following procedure is used to create both, master dome and master twilight flats. First a mask is constructed from the input hot and cold pixel maps. The raw flat frames are trimmed, overscan corrected (always using the same method used for other input calibration frames) and de-biased. Each frame is normalized to one illumination level (median pixel value 1.0). A mean frame is built for each input frame and pixels are rejected using a 5σ threshold. A median is computed from a stack of the resulting frames and the error of each input frame is calculated such as $\sigma = \sqrt{g}ain * counts$. Another rejection of 5σ outliers is performed on each input frame. The resulting master dome/twilight flat field will be the mean of the remaining data.

A similar algorithm as above is used to create a master night sky flat (`MASTER_NIGHT_SKYFLAT`) . Images are stacked and a median average is calculated, intended to remove any non-systematic effects (objects, cosmic rays, satellite tracks, etc.). Bad pixels are assigned a value of 0 before the normalization of the median image is obtained.

The final master flat field (`MASTER_FLAT`) is obtained in several steps. Low spatial frequencies are extracted from the master dome and master twilight flats by the process indicated below. The high spatial frequencies of the dome flat are obtained by dividing the dome flat by its low spatial frequency component. The low spatial frequencies of the twilight flat are then multiplied by the high spatial frequencies of the dome flat.

All bad pixels in the input images are replaced by the median value of the pixels in a box around the bad pixel. To reduce problems with Fourier filtering near image edges, the size of the image is increased by mirroring the edges and corners. A two-dimensional array is created containing the equivalent of a circular Gaussian convolution function in Fourier space (taking into account the quadrant shift introduced by the Fourier transform).The Fourier transform of the image is multiplied by the Gaussian filter. The image is transformed back, and the mirrored regions removed. The resulting image is normalized, excluding the bad pixel values.

The flat field correction of the science images merely consists of dividing the science frame by the master flat field described above and produced by the recipe *ocam_mtwilightnew* .

10.2.3 Fringing correction

Fringing due to variable strength of several skylines, mostly apparent at the long wavelengths, requires a different approach to background subtraction. A suitable strategy to construct a fringed background image, usable for subtraction, thereby removing the fringe pattern, remains to be determined. The approach used to construct a master fringe correction frame (`MASTER_FRINGE_FLAT`) in the OmegaCAM pipeline is presented here. A mask is constructed using the input hot and cold pixel maps. Input frames are trimmed, overscan corrected, de-biased and flatfield. Statistics of the resulting images are estimated iteratively. Each image is then normalized by dividing by its median pixel value, taking into account the bad pixels assigned to 0 using the mask. The master fringe flat is obtained by subtracting 1.0 from the normalized image.

For science images obtained with a band that requires fringing correction, the process is done by first scaling the fringe map by assuming that the standard deviation of the pixel values in the reduced science image depends on the scale factor used for the fringe map, with a minimum σ for the best scale factor. If the pixel values in the resulting de-fringed image are given by $x_{ij} = I_{ij} - aF_{ij}$, (I is the science image, and F the fringe map) and $\sigma = \frac{1}{N-1}\sum_{ij}(x_{ij} - \mu)^2$, then assuming $\frac{d\sigma}{da} = 0$ for the best scale factor a. This results in the following formula for a:

$$a = \frac{\sum(IF) - N_{\mu I \mu F}}{\sum(F^2) - N_{\mu F \mu F}},$$

where I is the (flat-fielded, de-biased) science image, F the fringe map, N the number of pixels used in the statistics (i x j, which is not equal to NAXIS1 x NAXIS2 if there are any masked pixels), and μI and μF the average of the pixel values in I and F respectively. Pixel maps (sigma thresholding) are used to exclude bright stars and select mostly pixels in fringe minima/maxima. The science image is then subtracted from the scaled fringe map.

10.2.4 Photometry correction

This will change soon....

10.2.5 Astrometry correction

This will change soon....

A Installation

This chapter gives generic instructions on how to obtain, build and install the OmegaCAM pipeline. Even if this chapter is kept 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 OmegaCAM pipeline distribution. These release-specific instructions can be found in the file `README` located in the top-level directory of the unpacked OmegaCAM pipeline source tree. The supported platforms are listed in Section A.1. It is recommended reading through Section A.2.3 before starting the installation.

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

A.1 Supported platforms

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

- Linux (glibc 2.1 or later),

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

A.2 Building the OmegaCAM pipeline

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

A.2.1 Requirements

To compile and install the OmegaCAM pipeline one needs:

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

An installation of the Common Pipeline library (CPL) must also be available on the system. Currently the CPL version 2.0 or newer is required. The CPL distribution can be obtained from `http://www.eso.org/cpl` .

Please note that CPL itself depends on an existing cfitsio installation. The cfitsio sources are available from the CPL download page or directly from the cfitsio homepage at `http://heasarc.nasa.gov/fitsio/fitsio.htm` . In conjunction with CPL 4.0 cfitsio 2.5.10 must be used.

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

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

A.2.2 Downloading the OmegaCAM pipeline source distribution

From the ESO ftp server, `ftp://ftp.eso.org/pub/cpl/ocam` , the latest release of the OmegaCAM pipeline distribution is available.

The OmegaCAM pipeline sources are distributed as a gzipped tar archive named like `ocam-X.Y.Z.tar.gz` , where `X`and `Y` are the major and minor release numbers, and `Z` indicates the patch level (which might be missing if no patch has been released).

A.2.3 Compiling and installing the OmegaCAM pipeline

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

The OmegaCAM pipeline distribution kit 1.0 contains:

ocam-manual-1.0.pdf	The OmegaCAM pipeline manual
cpl-4.1.0.tar.gz	CPL 2.1.1
esorex-3.7.0.tar.gz	esorex 3.5.1
gasgano-2.2.3-Linux.tar.gz	GASGANO 2.2.3 for Linux
gasgano-2.2.3-SunOS.tar.gz	GASGANO 2.2.3 for SunOS
ocam-0.5.0.tar.gz	OmegaCAM pipeline 0.1.3
ocam-calib-0.1.3.tar.gz	OmegaCAM calibration files 0.1.3

Here is a description of the installation procedure:

1. First, if an appropriate version of CPL (c.f. section A.2.1) does not already exist on the system, compile and install the CPL libraries and their dependencies. For detailed instructions on how to install the CPL libraries please refer to the CPL documentation.

2. Unpack the OmegaCAM pipeline sources in a choosen directory using

   ```
   $ zcat -d ocam-X.Y.Z.tar.gz | tar -xf -
   ```

   at the system prompt. This will create a directory called `ocam-X.Y.Z` containing the source tree.

3. Run the automatic installation script `setup` which is located in the top-level directory of the OmegaCAM pipeline source tree.

   If the CPL has not been installed into one of the system's standard directories, the configuration script must be told where the dependent libraries can be found. This is done by defining the environment variables `CPLDIR`.

   For example, if the CPL has been installed into `/somewhere` , (and therefore the CPL header files are located in `/somewhere/include` and the libraries are in `/somewhere/lib`) , the path assigned to the variable `CPLDIR` must be `/somewhere` , *i.e.*, the root directory of the CPL installation. The same is true for the environment variable corresponding to qfits.

   The `setup` script takes one argument which is the location where the OmegaCAM pipeline should be installed in your system. All its components will be located in the directory tree rooted at this path.

   The following example assumes that the variable `CPLDIR` is properly set or the CPL has been installed into one of the system's standard directories. This should keep the following example commands as simple as possible.

   Please note also that the usage of the default installation prefixes in the example below is just for demonstration purposes. Any directory for which one has write access can be used, although it is **not recommended** to use the distribution's source directory as the installation's target directory.

   The simplest way to set up the package is to run the following command at the system prompt from the source-tree's top-level directory:

   ```
   ./setup /usr/local/pipeline
   ```

   This script will install the OmegaCAM pipeline and all other necessary external programs and configuration files under the path given in the command line. It will create a `/usr/local/pipeline/bin` , and a `/usr/local/pipeline/config` directories containing the executables and configuration files for these programs.

After the installation has been completed the source tree is no longer needed and can be removed.

A.3 Configuring the pipeline recipe front-end applications

In this section an outline is given how to set up the recipe front-ends *EsoRex* and *Gasgano* so that the just installed OmegaCAM pipeline recipes can be executed by these applications.

For detailed instructions on how to configure the two currently available front-end applications, *EsoRex* and *Gasgano* , please refer to their documentation, available at `http://www.eso.org/cpl/esorex.html` and `http://www.eso.org/gasgano` respectively.

In the following it is assumed that the OmegaCAM pipeline was installed as described in Section A.2.3, *i.e.* , that the OmegaCAM recipes have been copied into `/usr/local/pipeline/lib/ocam/plugins/ocam-X.Y.Z` (`X`, `Y` and `Z` indicate the version number of the recipes).

A.3.1 Setting up the EsoRex command-line tool

The general syntax for the *EsoRex* front-end is the following:

esorex [*esorex_options*] *recipe_name* [*recipe_options*] *set_of_frames*

In order to execute a recipe, *EsoRex* must be told where the recipes can be found. This location can be passed to the tool using its command line option `--recipe-dir` followed by the complete path to the recipes. In this scenario the command to be executed at the shell's prompt will look like:

```
$ esorex --recipe-dir \
> /usr/local/pipeline/lib/ocam/plugins/ocam-X.Y.Z
```

However, the path to the recipe location(s) can also be set in the *EsoRex* configuration file. If an *EsoRex* configuration file does not already exist, it can be created by executing the command

```
$ esorex --create-config --recipe-dir \
> /usr/local/pipeline/lib/ocam/plugins/ocam-X.Y.Z
```

In addition to just creating the configuration file in its standard location `$HOME/.esorex/esorex.rc` the path to the recipes is also added by this command. This way for updating the configuration, can be repeated whenever a new version of the recipes is installed. It will replace the path to the recipes in the *EsoRex* configuration file.

If *EsoRex* has already been used for running recipes from other instruments, the path to the OmegaCAM recipes must be appended to the existing configuration file. To do this edit the configuration file `$HOME/.esorex/esorex.rc` . Go to the entry starting with `esorex.caller.recipe-dir` . This is a colon separated list of directories searched by *EsoRex* for recipes. To add the OmegaCAM recipes just append the OmegaCAM recipe installation directory, separated by a colon (:), to the end of this list.

To verify the updated configuration execute the following command at the shell's prompt:

```
esorex --recipes
```

This should display a list of the available recipes on the terminal screen.

At the *EsoRex* homepage, `http://www.eso.org/cpl/esorex.html` , a detailed description of the application can be found.

A.3.2 Setting up Gasgano

The OmegaCAM recipe set can be incorporated into *Gasgano*'s configuration using the *Preferences* dialog from the *File* menu. Select the tab labeled *Recipe Configuration* and press the *Add Recipe* button. A file selection dialog pops up where the recipes to add can be selected. The selection has to be confirmed and *Gasgano* must be restarted to activate the new recipe configuration. Now the recipes are seamlessly integrated into the application and the files to process can be passed to the recipes using drag'n drop (for details please have a look into the *Gasgano* User Manual [11]).

B Abbreviations and acronyms

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

C Troubleshooting Guide