OmegaCAM

OmegaCAM Data Flow System - Users and programmers manual

User manual for ESO-DFS

Issue: Version 2.11

Date: 23 Jul 2004

Prepared by: Rengelink, Boxhoorn, Deul, Kuijken, Valentijn, Vermeij, Begeman, Helmich

Purpose of printout: PAE THIRD DELIVERY

www.astro.rug.nl/∼omegacam

CHANGE RECORD

Version 1.0 has been delivered at the FDR. Following comments at FDR version 1.1 has been prepared, which has been approved by FDR board.

Thus version 1.1 is the FDR formal version.

Changes between Version 1.0–FINAL DESIGN REVIEW and Version 1.1–FINAL DESIGN REVIEW

Additions

New section on 2-3 day cycle specifications – 2.2 DFS requirements.
Specify all three levels of TSF's, as given in VST-SPE-OCM-23100-3064, in requirements.
Detailed estimates of observing times for photometric checks – 2.2 DFS requirements, **req.**562, **req.**563, **req.**564.
Include tilt determination – **req. 571** *Camera focus/tilt*.
CalFile– 562S *Sky brightness* – **req.**562.
Reference to **req.**563 in CA – **req.**564.
OmegaCAM DID – 1.2 Applicable documents.
Once/year dark dome test – **req.**533.
Processing of calibration data follows telescope schedule – 2.2 DFS requirements, 6.1 Data reduction software requirements. Expanded the glossary with all term used in the document.

Updates

Clarify fast recipe for Technical Specifications conformance – **req.**562, **req.**563.
Erroneous references to darkcurrent check for **req. 547** *Quick detector responsivity check* removed.
Nonexistent **CalFile– 561** removed – **req.**533.
Stars have to be observed during the night – **req.**525.
req. 571 *Camera focus/tilt* is no longer a workhorse/*doit* – 1.4 Abbreviations and Acronyms, 5.10 On site quick look analysis.
Exposure times TBC during commissioning – **req.**561.
More accurate description of the algorithm – **req.**523.
Lamp procedure TBC. – 5.4 Detectors operational specific calibrations.
Removed reference to QCO – **seq.– 631**.
Mention acceptance of multi-extension FITS files – **seq.– 631**.
Target-related template parameters (only) where applicable – 4.4 Observing Templates.
Reworded sentence about DFS-pipeline – 5.10 On site quick look analysis.
Reworded sentence about modules – 6.1 Data reduction software requirements.
Rotator *offset* angle – 4.4 Observing Templates.
Use plots for analysis – **req. 571** *Camera focus/tilt*.

Rephrased in terms of provided functionality instead of needed functionality – **7 FUNCTIONALITIES**.

Changes between Version 1.1–FINAL DESIGN REVIEW and Version 2.0–PAE

major changes since FDR

- settling the key bands u',g',r' and i' and filter sets

Minor changes since FDR

- req 534, 535 - uses r'filter
- req 562 - Calfile 562S is split up in its separate photometric bands
- req 569 - Calfile 569 secondary standards, internal now
- req 561 - Emphasize two shutter directions

Additions

CalFile– 548F *Illumination correction fit coefficients* – **req.**548.
CalFile– 562Su *Sky brightness-u'* ,
CalFile– 562Sg *Sky brightness-g'* ,
CalFile– 562Sr *Sky brightness-r'* ,
CalFile– 562Si *Sky brightness-i'* – **req.**562.
Cross-calibration of *composite* vs. *monolithic key* filters **CalFile– 565C** *composite -> monolithic* – **req.**565.
Recipe– PhotCal_Monitoring_Sky for on-line sky spectrum measurement – **req.**562.
INT-La Palma preparatory programme for one square degree secondary standard fields .
Added Appendix A6: sky grid of pointing positions, also used for pixel grid in co-addition

Updates

The *composite* filter is created by a different company from the one that created the *monolithic* filters – **req.**565.
Alternative analysis method for CCD Charge Transfer Efficiency – **req.**534.
CalFile– 533 replaced by **CalFile– 533P** *CCD Linearity Plot* , **CalFile– 533M** *CCD Linearity map* – **req.**533.
CalFile– 563Z,I renamed **CalFile– 563**– **req.**563.
Further specifications of focus/tilt analysis – **req.**571.

Removed

CalFile– 546W *Weightmap* – **req.**546.
On-site on sky health check monitoring using standard fields. This activity, though emphasized at FDR by ESO, was removed on request of ESO. **req. 562** *Monitoring*.

Following these changes CP Version 2.0 has been prepared for PAE.

Changes between Version 2.0–PAE and Version 2.1–PAE

Additions

Calibration Verification Matrix added as Appendix A7.
Figure 3a containing only parts of the data model relevant to ESO-DFS – section 3.
The command-line syntax for each **Recipe**. Clarify which parameters are mandatory and which are optional.
A complete description of each parameter of a **Recipe**.
Valid values or a range of valid values for each parameter of a **Recipe**.
Check reproducibility of filter exchange mechanism – **req. 542** *Flat-field - dome key bands + user bands - doit*.
Typical exposure time added – **req. 543** *Flat-field - twilight*.
Telescope should be tracking – **req. 543** *Flat-field - twilight*.
Jittered observation should be used to remove stars – **req. 543** *Flat-field - twilight*.
In each **req.** where stacking is used, explained why and how this is done.
Recipe– Illumination_Correction_Verify in **req. 548** *Illumination correction* Fully automated recipe to determine illumination correction- description of procedure and recipe.
Recipe– PhotCal_Monitoring_Sky used in **req. 562** *Photometric Calibration - monitoring* .
Recipe– PhotCal_Extract_Resulttable used in **req.**562, **req.**563 and **req.**564.
Documentation for CAT III recipes - **req. 554** *PSF anisotropy* **req. 562** *Photometric monitoring- Sky Brightness monitoring* **req. 548** *Illumination correction- Quick verification*
A description of the reference hardware used for processing has been given.
Suggested QC parameters in **CAP** section.
Output plot for **Recipe– PSF_Anisotropy** included – **req. 554** *PSF Anisotropy*.

Updates

Made names of calfiles in all figures consistent with the nomenclature used in the requirements.
Consistently used the “master” prefix in **CalFile–** name throughout the documentation.
Read noise value and gain are single numbers in the ESO context, not **CalFile–** 's .
Clarified illumination correction procedure – **req. 548** *Illumination Correction*.
Use ADU in output units – **req. 521** *CCD read noise - doit*, **req. 532** *CCD Particle Event Rate*.
TSF section did not refer to read noise template – **req. 532** *CCD Particle Event Rate*.
Changed default values for cold pixel threshold in the text – **req. 535** *CCD Cold Pixels*.
Mention astrometric pre-solution procedure to determine approximate pixel scale and position of reference pixel – **req. 551** *Position of Camera in focal plane*.
Detailed clarification of the objective – **req.**553Telescope and Field Rotator tracking.
In **req.**562, **req.**563 and **req.**564 catalogue derivation and processing of catalogues have

been separated, resulting in a new dataflow. The text of the requirements, the figures, analysis descriptions and data reduction procedures have been updated accordingly.
CalFile– 562u, **CalFile– 562g**, **CalFile– 562r**, **CalFile– 562i** replaced by generic **CalFile– 562T** *Photom + Sky*.
Full re-write of data reduction using MIDAS procedure – **req.**571.
Perform regridding without co-addition in **Recipe– Stare** in **seq.– 632**.
Ordered requirement sections by requirement number, rather than by category.
Made each requirement start at the top of a page.
Processing times updated following measurements with the specified reference hardware.
CAP sections updated to reflect the use of **Recipe** parameters.
Replaced references to "image pipeline" to "image part of the pipeline".
Changed the order of items in each requirement.
Grid on the sky has been fine-tuned – Appendix A6.

Removed

Text related to automatic trend-analysis and verification of pipeline products in all requirements.
Text suggesting that reminiscence can be removed by CCD read out – **req. 536** *CCD Hysteresis, strong signal*.
Text about reminiscence short than 60 seconds – **req. 536** *CCD Hysteresis, strong signal*.

Following these changes CP Version 2.1 has been prepared for PAE.

Changes between Version 2.1–PAE and Version 2.11–PAE

Additions

Included mandatory raw domeflats in inputs section – **req. 523** *CCD gain*.

Note added about cold restart of FIERA – **req. 541** *Bias - doit*.

Copied over number of stars for photometric accuracy, magnitude accuracy and response to SExtractor automatic aperture concerns from Photometric progress report – **req. 562** *Photometric Calibration - monitoring*, **req. 569** *Secondary Standards*.

Values included for SExtractor, LDAC and Swarp configuration parameters if they differ from the defaults for each of these packages – **req.**531, **seq.– 632**, **seq.– 633**, **seq.– 634**, **seq.– 636**.

Updates

Renamed **Recipe– Stare** to **Recipe– Reduce** and renamed **Recipe– Dither** to **Recipe– Coadd** because they do not correspond to **Mode– Stare** N= and **Mode– Dither** N= respectively – 6 DATA REDUCTION SOFTWARE SPECIFICS.

Corrected units of particle event rate – **req. 531** *CCD Dark Current - doit*, **req. 532** *CCD Particle Event Rate*.

Some CCD's will have worse CTE than the required accuracy – **req. 534** *CCD Charge Transfer Efficiency*.

Made roles of optical axis and the mechanical rotator axis more clear – **req. 551** *Position of Camera in focal plane*.

Text about use of guide stars for pointing tests made more explicit – **req. 552** *Telescope and Field Rotator tracking*.

Explained role of the ADC in the text – **req. 566** *Dependency on angle - ADC, rotator/reproducibility*.

Clarified that regridding takes place in **Recipe– Reduce** for observations done in **Mode– Stare** – 6.3 The Image Pipeline, **seq. 632** *Trim, de-bias, flatfield*, **seq. 636** *Dedithering/dejittering*.

Corrected titles of **seq.– 633** and **seq.– 634** to reflect that the seeing is determined in **seq.– 634**.

Rephrased and removed references to trend-analysis – 2 Manual, Modules, Recipes.

Removed

References to Paranal and QC1 removed from figure 3b – 3 Data Flow.

Following these changes CP Version 2.11 has been prepared for PAE.

TABLE of CONTENTS

1 INTRODUCTION . . . 10
1.1 Scope of this document . . . 10
1.2 Applicable documents . . . 13
1.3 Reference documents . . . 13
1.4 Abbreviations and Acronyms . . . 13
2 Manual, Modules, Recipes . . . 16
3 Data Flow . . . 21
4.0 OBSERVING MODES and STRATEGIES . . . 26
4.1 Observing Modes . . . 28
4.2 Observing Strategies . . . 29
4.3 Filtering . . . 30
4.4 Observing Templates . . . 30
4.5 Field correctors . . . 31
5 BASELINE CALIBRATION REQUIREMENTS . . . 32
5.0 Documentation system, **Odoco** . . . 32
5.1 Functional Checks . . . 38
5.2 Detector Electronics specific Calibrations . . . 38
5.2.1 Req.– *Cat I: CCD read noise - doit* . . . 39
5.2.2 Req.– *Cat I: Hot pixels* . . . 41
5.2.3 Req.– *Cat I: CCD gain* . . . 43
5.2.4 Req.– *Cat III: Electromagnetic Compatibility* . . . 46
5.2.5 Req.– *Cat III: Electrical cross talk* . . . 47
5.3 Detectors specific calibrations . . . 48
5.3.1 Req.– *Cat I: CCD Dark Current - doit* . . . 49
5.3.2 Req.– *Cat I: CCD Particle Event Rate* . . . 53
5.3.3 Req.– *Cat I: CCD Linearity* . . . 54
5.3.4 Req.– *Cat III: CCD Charge Transfer Efficiency* . . . 57
5.3.5 Req.– *Cat I: CCD Cold Pixels* . . . 59
5.3.6 Req.– *Cat III: CCD Hysteresis, strong signal* . . . 61
5.4 Detectors operational specific calibrations . . . 62
5.4.1 Req.– *Cat I: Bias - doit* . . . 66
5.4.2 Req.– *Cat I: Flat-field - dome key bands + user bands - doit* . . . 70
5.4.3 Req.– *Cat I: Flat-field - twilight* . . . 75
5.4.4 Req.– *Cat I: Flat-field - night sky* . . . 80
5.4.5 Req.– *Cat I: Flat-field - fringing* . . . 84
5.4.6 Req.– *Cat I: Flat-field - master flat* . . . 88
5.4.7 Req.– *Cat I: Quick detector responsivity check-doit* . . . 92
5.4.8 Req.– *Cat II: Illumination correction* . . . 94
5.5 Astrometric Calibration . . . 98

5.5.1 Req.– *Cat III: Position of Camera in focal plane* ... 100
5.5.2 Req.– *Cat III: Telescope Pointing and offsetting* ... 102
5.5.3 Req.– *Cat III: Telescope and Field Rotator tracking* ... 103
5.5.4 Req.– *Cat III: PSF Anisotropy* ... 105
5.5.5 Req.– *Cat I: The astrometric solution for templates - doit -see 6.3.4* ... 108
5.5.6 Req.– *Cat III: The astrometric solution for Guide CCD's* ... 109
5.6 Photometric Calibration ... 111
5.6.1 Req.– *Cat III: Shutter Timing* ... 115
5.6.2 Req.– *Cat I: Photometric Calibration - monitoring* ... 117
5.6.3 Req.– *Cat I: Photometric Calibration - zeropoint keybands - doit* ... 126
5.6.4 Req.– *Cat I: Photometric Calibration - zeropoint user bands* ... 132
5.6.5 Req.– *Cat I: Filter band passes - user bands vs key bands* ... 136
5.6.6 Req.– *Cat III: Dependency on angle - ADC, rotator/reproducibility* ... 137
5.6.7 Req.– *Cat III: Linearity (as a function of flux)* ... 138
5.6.8 Req.– *Cat III: Detection limit and ETC calibrations* ... 139
5.6.9 Req.– *Cat I: Secondary Standards* ... 140
5.7 Internal alignments, optics etc ... 143
5.7.1 Req.– *Cat III: Camera focus/tilt* ... 144
5.7.2 Req.– *Cat III: Ghosts - ADC* ... 150
5.8 Effect of Telescope ... 151
5.9 Workhorses and End to end tests ... 152
5.10 On site quick look analysis ... 153
Fig 6. Data Flow of Science and Calibration data ... 154
6 DATA REDUCTION SOFTWARE SPECIFICS ... 155
6.1 Data reduction software requirements- see DRS and URD v1.2 at FDR ... 155
6.2 Estimate of data volumes - see DRS and URD v1.2 at FDR ... 155
6.3 The Image Pipeline ... 155
6.3.1 Seq.– *Cat I: Ingestion and image statistics* ... 159
6.3.2 Seq.– *Cat I: Trim, de-bias, flatfield* ... 161
6.3.3 Seq.– *Cat I: Construct weigths* ... 166
6.3.4 Seq.– *Cat I: Astrometric calibration, determine seeing* ... 169
6.3.5 Seq.– *Cat I: Apply photometric calibration* ... 173
6.3.6 Seq.– *Cat I: Dedithering/dejittering* ... 174
7 FUNCTIONALITIES ... 177
7.1 Image manipulation ECLIPSE/SWARP ... 178
7.2 Catalog manipulation LDAC/ SEXTRACTOR ... 183
7.3 Header manipulation ECLIPSE ... 188
7.4 Miscellaneous functionalities ... 189
A1 LIST of CALIBRATION REQUIREMENTS ... 190
A2 LIST of RAW CALIBRATION DATA ... 191
A3 LIST of DFS I/O CALIBRATION FILES ... 192

A4 LIST of DFS INPUT REFERENCE CATALOGUES 193
A5 LIST of SEQUENTIAL STEPS in IMAGE PIPELINE 194
A6 SKYGRID of PROJECTION CENTRES 195
A7 CALIBRATION VERIFICATION MATRIX - URD 196

1 INTRODUCTION

1.1 Scope of this document

The present document provides the User and Programmer manual of the deliverable 'Data Reduction Specifications' as proposed in VLT-SPE-ESO-19000-1618: "The document defines the detailed data reduction procedures to be applied to raw data frames including input data and algorithms required".
At FDR the Consortium has delivered the document 'Data Reduction Specifications' VST-SPE-OCM-23100-3051 version 1.1, which includes various baseline requirements with respect to the data reduction software. These baseline requirements are NOT repeated here, but the specifications and descriptions have been largely expanded in the present document.

The present document describes the implementation of the **Calibration Plan** VST-PLA-OCM-23100-3090, and it is meant to guide the deliveries of data reduction modules and recipes to ESO. The document is written as a manual to developers and can eventually be used by operators, using the OmegaCAM pipeline, which will be constructed by ESO on the basis of the present delivery.

Standardization of observing modes, calibration procedures and related data reduction procedures play an import role in the **OmegaCAM** instrument concept and the present document obeys the specification of the standardized observing scenarios for science observations (**templates**) and the baseline requirements for the qualification, quality control and calibration of the instrument both during switch-on, commissioning and routine phase (**RP)**. These requirements are specified in the **User Requirement Document (URD)** (VST-SPE-OCM-23100-3050).

Implementations of these requirements are further specified in the **CP**. Details of the related data reduction are given in the following pages.

Instantiations of the Calibration Plan will form the **Schedule** of the telescope, which for the commissioning phase shall be written by the Instrument Consortium and will form a part of the **Commissioning plan** (VST-PLA-OCM-23100-3100). The present manual has been coordinated with the commissioning plan and in fact precise cross references are maintained between Commissioning Plan and this document (and **CP**).

During routine operations the schedule shall be written by ESO, but the present manual describes the recipes and modules to be used in the off-line data reduction during **RP**.

By labeling tasks in **Category I+II** and **Category III** it has been made explicit, which tasks are run as (semi-)automatized pipelines, mostly at ESO Headquarters (Cat I+II), and which tasks are run as more isolated (often Commissioning) activities. Further details on this lay-out are given in Section 2.

Figure 1 sketches that part of the organization of the **OmegaCAM** consortium which relates to the deliverables of data reduction and calibration issues. The sketch is not exhaustive, but is meant to highlight the basic relation between National tasks (mostly NL as far as data reduction is concerned, with products/results marked in blue) and tasks relating to deliverables to ESO (on the right side, with deliverables marked in red).

For some Dutch National tasks collaborations are on-going with other European data centers for wide field imaging, such as Terapix (France), the Astronomical Observatories of Capodimonte and Padova, and the Sternwarte München.

The deliverables to ESO are driven by the Agreement with ESO, which in turn refers to 'Data Flow System User Requirements' VLT-SPE-ESO-19000-1618 (DFS-req-16 in the sketch), which in turn specifies the hierarchy of deliverables: **User Requirement Document (URD)** and **Calibration Plan document** and various S/W deliverables.

While URD and CP were delivered and approved at PDR and FDR, the present document contains the delivery one level down on this sketch: recipes, modules and manuals.

The **OmegaCAM** User requirements related to the Instrument Software are specified in a separate document [VST-SPE-OCM-23100-3060].

As can be seen from the sketch, the relation to the next level: operation Paranal and pipeline DMD is complex and the definition of Categories is meant to make this more transparent.

The consortium has implemented a documentation control system with a strict documentation format for the definition of calibration **req.**'s, and named this **Odoco** (**OmegaCAM** document control). This system is explained in detail in Section 5.0. The green encircled region in the figure indicates the various documents and files which will be guarded by **Odoco**.

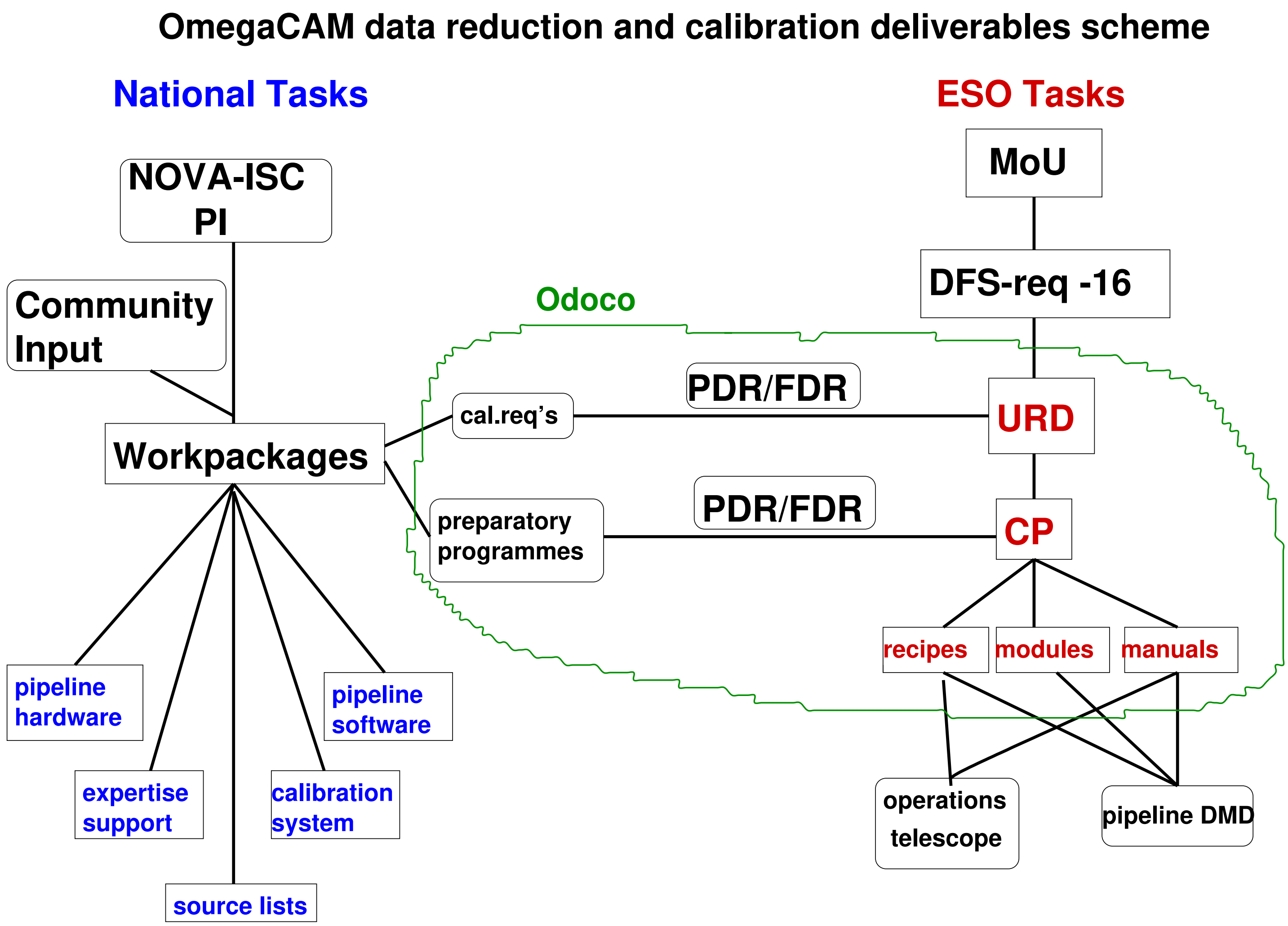

OmegaCAM data reduction and calibration deliverables scheme
National Tasks
ESO Tasks
MoU
NOVA-ISC
PI
DFS-req -16
Community
Input
Odoco
PDR/FDR
cal.req's
URD
Workpackages
PDR/FDR
preparatory
programmes
CP
recipes
modules
manuals
pipeline
hardware
pipeline
software
expertise
support
calibration
system
operations
telescope
pipeline DMD
source lists

Figure 1

1.2 Applicable documents

The following documents, of the exact issue shown, form a part of this document to the extent specified herein. In the event of conflict between the documents referenced herein and the contents of this document, the contents of this document shall be considered as a superseding requirement.

[] VST-PLA-OCM-23100-3100 OmegaCAM Commissioning Plan

[] VST-PLA-OCM-23100-3010 Project Management Plan and Project Plan and Schedule

[] VST-PLA-OCM-23100-3020 Product Assurance Plan

[] VST-PLA-OCM-23100-3080 Technical Operations and Maintenance Plan

S/W deliverables depending on the URD

[] OmegaCAM Data Interface Dictionary

[] OmegaCAM Template Signature Files

[] OmegaCAM Exposure Time Calculator

1.3 Reference documents

The following documents are referenced in this document.

[] VLT-SPE-ESO-19000-1618	1.0 21/04/1999 — VLT Data flow for the VLT instruments Requirement Specification
[]	MoU + Agreement OmegaCAM + ESO
[] VST-PLA-OCM-23100-3090	OmegaCAM Calibration Plan V2.0
[] VST-PLA-OCM-23100-3051	OmegaCAM Data Reduction Specifications V1.1
[] VST-PLA-OCM-23100-3030	Safety Analysis and Compliance Assessment
[] VST-TRE-OCM-23100-3040	Design Analysis, Performance Report
[] VST-TRE-OCM-23100-3041	Design Analysis, Performance Report: Mechanical
[] VST-ESO-OCM-23100-3042	Design Analysis, Performance Report: Detector System
[] VST-TRE-OCM-23100-3043	Design Analysis, Performance Report: Electronic
[] VST-SPE-OCM-23100-3050	Data Flow System User Requirements
[] VST-SPE-OCM-23100-3060	Instrument Software
[] VST-SPE-OCM-23100-3064	OmegaCAM Observation Software Design Description
[] VST-PLA-OCM-23100-3070	MAIT and Alignment Plan
[] VLT-MAN-ESO-19000-2050	FTU FITS Translation Utility User Manual
[]	SExtractor v2.3 User's manual
[]	Eclipse User's Guide - April 1, 2002
[]	FFTW manual 3.0.1
[]	The LDACTools Library v1.2 User's guide
[]	Pipeline Documentation Ver. 1.4
[]	SWarp v2.0 User's guide

1.4 Abbreviations and Acronyms

Abbreviations and Acronyms used in this document

A/D	Analog/digital
ACS-dbase	Astronomical Calibration Source database
ADC	Atmospheric Dispersion Corrector
ADU	Analog to Digital Unit
AGN	Active Galactic Nucleus
BRD	Baseline Requirements Document
CA	Calibration Analysis
CAP	Calibration Analysis Procedure
CCD	Charge Coupled Device
CO	Calibration Observation
CP	Calibration Plan
CTE	Charge Transfer Efficiency
CVS	Code Version System
DFS	Data Flow System
ESO	European Southern Observatory
ETC	Exposure Time Calculator
FOV	Field of View
FWHM	Full Width at Half Maximum
GRB	Gamma Ray Burst
GT	Garanteed Time
HZSS	High Redshift Supernova Search
ICS	Instrument Control Software
IWS	Instrument Workstation
ISO	Infrared Space Observatory
IST	Instrument Science Team
KBO	Kuiper Belt Object
MoU	Memorandum of Understanding
NOVA	(Dutch) Nederlandse Onderzoekschool Voor Astronomie
OaPd	Astronomical Observatory of Padua
OB	Observation block
OD	Observation Description
OPC	Observing Programme Committee
OT	Optical Transient
OT	Observing Template
PPP	Photometry Preparatory Programme
PSF	Point Spread Function
QC0	Quality Control zero
QC1	Quality Control one

QSO	Quasi-Stellar Object
RPE	Relative Pointing Error
RP	Routine Phase
RSRF	Relative Spectral Response Function
RTD	Real Time Display
SCP	Supernova Cosmology Project
SED	Spectral Energy Distribution
SSO	Solar System Object
S/N	Signal/Noise
S/W	Software
TBC	To Be Confirmed
TBD	To Be Defined
TBW	To Be Written
TCS	Telescope Control System
TP	Target Package
TSF	Template Signature File
URD	User Requirement Document
USM	Universitäts Sternwarte München
VLT	Very Large Telescope
VST	VLT Survey Telescope
WFI@2.2m	Wide Field Imager at the ESO 2.2m telescope

2 Manual, Modules, Recipes

This document follows the structure of the Calibration Plan and the Data-Reduction Specification. The key sections for data reduction are Section 5 and 6. Section 5 details the various calibration requirements, including specifications of observations and data-reduction procedures. Section 6 details application of the calibration data to the science observations, including details of the data-reduction procedures.

This document and its associated deliveries provides data reduction recipes which have the following properties:

- Recipes can be taken as black-box modules that produce a specific result based on a number of inputs. In this view, recipes are described in terms of their user-interface. In sections 5 and 6, the relevant items are:

  Recipe The name of the recipe

  Inputs The data (either raw, or calibration) used as input for the recipe

  Outputs The result produced by the recipe

  TSF The Template Signature File, which will be used to trigger the recipe

- Recipes involve a series of processing steps necessary to produce a result. In this view recipes are described in terms of algorithms and sequences of primitive data-reduction operations. In Section 5 and 6, these descriptions can be found under:

  CA The Calibration Analysis - a conceptual description of the processing steps.

  CAP The Calibration Analysis Procedure - a pseudocode description of processing steps

  Required functionalities list of functions from external libraries used in CA

The relation between different recipes is outlined in Section 3, which focusses on data-flow.

Standards for data taking and data reduction are strongly coupled in the **OmegaCAM** data reduction system. Therefore, it is useful for the present document to repeat these standards in Section 4. This includes items like observing mode, observing strategy and TSF.

All **OmegaCAM** data processing relies on 'primitive' data reduction operations implemented in external (C-)libraries. Hence, recipes build on lower-level functionalities provided by these libraries. The external libraries are described in Section 7.

This document comes with an implementation of the data-reduction procedures. The source code itself is documented within the distribution. In the remainder of this Section we outline the relation between this document and the implementation.

The delivery of OmegaCAM data reduction software consists in the first place of the textual description including the pseudo code as given in Section 5, together with a set of C libraries as described in Section 7.

Next to this, for the convenience of the integration into ESOs DFS infrastructure and as additional documentation and as example implementation of the various requirements, we include a full version of working python code mirroring the pseudo code given in Section 5.

This working version includes a filebased user interface. It will be used by the consortium during the testing of the instrument in Europe and the Commissioning in Paranal. This code comes with units tests, which were successfully applied, and a set of test data. The working code consists of a number of Python modules within the **astro** package hierarchy. The relation between the packages and the requirements in Section 5 is as follows:

astro.recipes contains selfstanding data-reduction modules with a file-based interface, corresponding to item **Recipes** of the requirements in Sections 5 and 6.

astro.main Object definitions for the CalFiles specified in **Outputs**. The production (`make()`) and verification (`verify()`) methods match to description given in the **CAs** and the pseudocode given in the **CAPs**. (Note, that an additional method `compare()` is included in the Class definitions, but is not called/used at the Recipe level. When desired, these methods can be used to compare objects with reference objects, or study there variation in time (trend analysis).

astro.external Interfaces to the 'primitive' data-processing functions provided by external C-libraries.

Also included in the software is a version of the **eclipse** image processing library which adds FITS files as primitive data type to Python, and is used to implement image-processing.

Hardware

In sections 5 and 6 estimates for processing time are given. These estimates are processing times as performed on the following hardware:

- **Intel Pentium 4 CPU 1700Mhz, 512Mb internal memory** - for all processing tasks except coaddition
- **AMD Athlon MP 2000+ (Dual CPU), 1Gb internal memory** - for coaddition

Example: Read_Noise

To clarify the way to navigate through the documentation of the various recipes, we present **req. 521** *CCD read noise* as a detailed example.

req.521 (the requirement), recipe Read_Noise.py in **astro.recipes** and ReadNoise.py in **astro.main** are closely related:

- The **Objective** in **req. 521** *CCD read noise* tells us that the purpose of this requirement is to "Measure the CCD read noise (in ADU's) as a standard health check." The **Objective** also gives a short description of the method used to determine this value.

- **req. 521** *CCD read noise* has a **Recipe** called Read_Noise with **Inputs** of 2 raw biases and a previous version of **CalFile– 521** *Readout noise*, and with **Outputs**: **CalFile– 521** *Readout noise*
- The recipe name Read_Noise corresponds to **astro.recipes.Read_Noise.py**. The documentation (reproduced below) lists de bias frames as mandatory input. Because the default recipe does not do trend analysis, no previous readout noise measurement is required. In addition two configuration parameters can be given.
- The source code of **astro.recipes.Read_Noise.py** (partly reproduced below) includes a function `make_readnoise()` This function associates the inputs with attributes of the ReadNoise object and then calls methods on a ReadNoise object to do the actual processing.
- **req. 521** *CCD read noise* details the calibration analysis (**CA**) necessary to process the read noise measurements. The pseudocode (in the **CAP**) shows how this is accomplished in terms of primitive data operations for given input and two configuration parameters
- The processing procedure is implemented as method `make()` on the ReadNoise class defined in module **astro.main.ReadNoise.py** (partly reproduced below). Note, the code also contains the methods `verify()` and `compare()` respectively, which might be useful for off-line quality verifications and trend analysis.

Note that:

1 **req.**521 is related to **astro.recipes.Read_Noise.py** through item **Recipe**

2 **req.**521's **CA** and **CAP** is implemented by **astro.main.ReadNoise.py**

3 **astro.recipes.Read_Noise.py** uses **astro.main.ReadNoise.py** to do the actual work

Documentation for recipe Read_Noise

Note the reference to **req.**521 in the description.

```
--------------------------------------------------------------------------
 CATEGORY : data processing, calibration pipeline.

 PURPOSE : generating a valid read noise calibration (Calfile 521).

 FULLFILLS : requirement 521 (Category I)
---------------------------------------------------------------------------
 This recipe is used to derive a valid calibration of the read noise
(Calfile 521) for one particular chip.

 The recipe always takes as input a list of raw bias frames. Optionally, two
 process parameters can be set : the maximum number of iterations used in
 deriving the read noise, and the rejection threshold in sigma. The defaults
 for these parameters are 5 and 5.0, respectively.
```

```
------------------------------------------------------------------------------
 Mandatory inputs :

   -i : list of input raw bias frames

 Configuration parameters (optional) :

   -max : maximum iterations (default : 5)
   -rej : rejection threshold (default : 5.0 sigma)

 Example of use :

   python Read_Noise.py -i raw1.fits raw2.fits -max 10 -rej 3.0
-----------------------------------------------------------------------------
```

Selected source code for recipe Read_Noise.

Note how processing is implemented in terms of a ReadNoise object (some lines have been omitted)

```
from astro.main.ReadNoise import ReadNoise

def make_readnoise(raw_bias_frames, rej_thresh = 5.0, max_iter = 5):

    readnoise = ReadNoise()

    readnoise.raw_bias_frames = raw_bias_frames
    readnoise.make()
    return readnoise
```

Selected source code for Read_Noise algorithm

```
class ReadNoise(DBObject, ProcessTarget):

    # define key attributes
    read_noise      = persistent('The read noise measurement', float, 0.0)
    raw_bias_frames = persistent('The raw biases, used to do the measurement',
                                 RawBiasFrame, [])
    process_params  = persistent('The process parameters',
```

```
                                    ReadNoiseParameters, None)

    def make(self):
        """Do a read_noise measurement
        """
        diff_image = (self.raw_bias_frames[0].image - \
                      self.raw_bias_frames[1].image)

        max_iter = self.process_params.MAXIMUM_ITERATIONS
        sig_thresh = self.process_params.REJECTION_THRESHOLD
        stats = diff_image.iter_stat(max_iter=max_iter, threshold=sig_thresh)

        self.read_noise = stats.stdev/math.sqrt(2)

        Message('The read noise is : %6.2f' % self.read_noise)
```

3 Data Flow

This section gives a global overview of the relation between the various calibration data. Two figures are presented that illustrate the data flow through the various parts of the pipeline in the software system. The various requirements and related output calibration files map in most cases directly to software modules. Therefore these figures are instrumental in understanding the relations between the software modules of the system.

A pipeline may be defined as a series of connected modules, where the output of one module is the input of another module. It is convenient to distinguish between different parts of the pipeline, as illustrated in Fig 3a. Here we distinguish between the bias pipeline (brown), the flat field pipeline (red), the photometry pipeline (purple), and the image pipeline (yellow). At ESO these parts are integrated in a Pipeline. Each part of the Pipeline produces a distinct result (bias, flatfield, zeropoint and calibrated science data respectively), that provides a convenient/obvious entry-point for human quality control.

Figure 3a presents the procedurized operations of the pipeline which has the goal to transfer raw images into astrometrically and photometrically calibrated images. A more detailed view of the image part of the Pipeline is given in Section 6.3.

The coloured requirement boxes in Figure 3a correspond to recipes and modules which propagate directly to the image part of the pipeline, in other words, the output calibration files of these recipes are used as input in the image part of the pipeline. The non-coloured boxes in the figure correspond to health checks which do not propagate directly into the image part of the Pipeline.

In Fig 3b we also include local, procedurized, checks on the health of the system. Some non-coloured boxes in this figure correspond to Commissioning activities which might raise flags, or trigger reports, but do not propagate directly into the DFS Pipeline.

Most of the **Category III** recipes (recipes mostly used during Commissioning) correspond to these non-coloured boxes. But note, there are four activities to be scheduled regularly (doit's daily-monthly), which do not propagate into the image part of the Pipeline, but have to be processed regularly to monitor the overall health of the system: the **Category I recipes**:
req. 547 *Quick Check*
req. 521 *Read Noise*
req. 531 *Dark Current*
req. 532 *Particle Event Rate*.

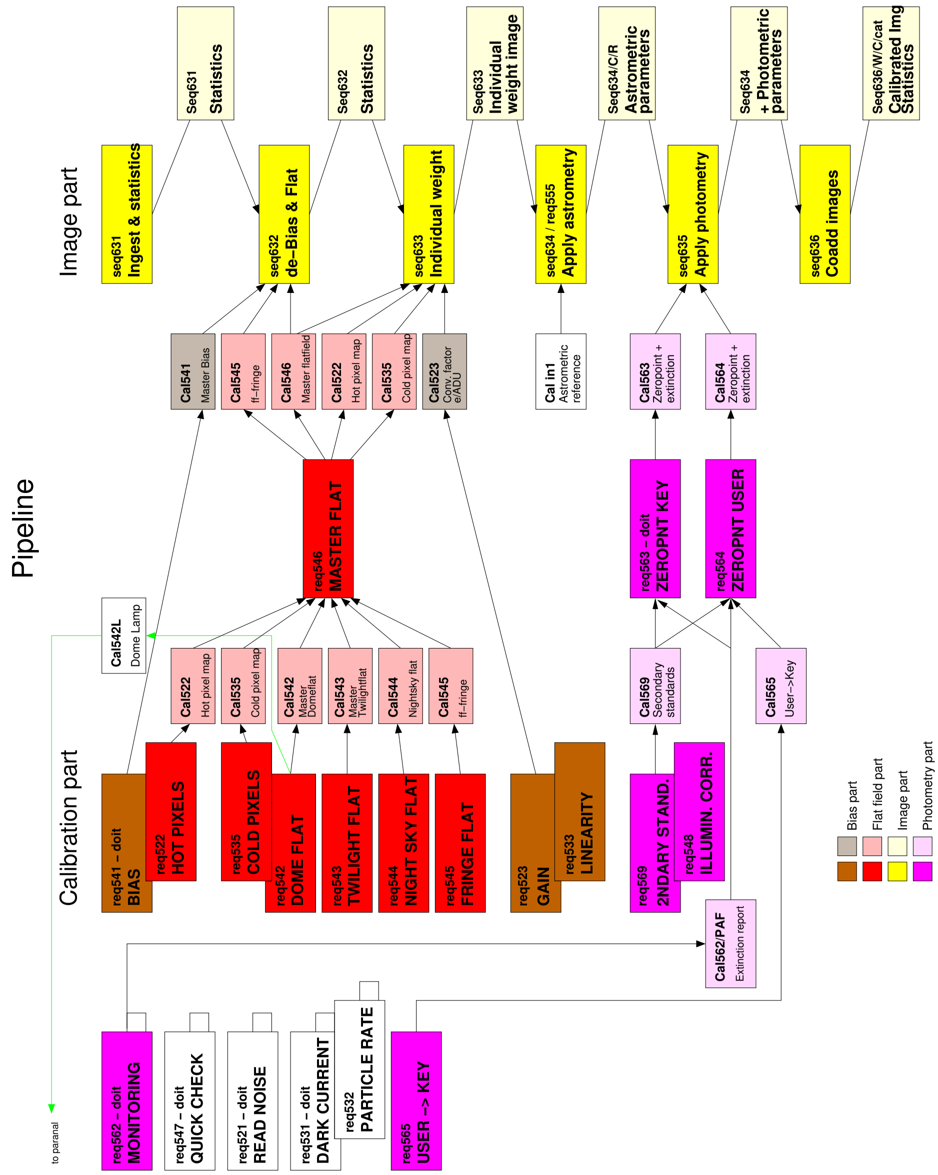


Fig 3a The **OmegaCAM** ESO-DFS data model

The OmegaCAM CP is designed in such a way that the order in time of the processing of calibration data is set by the timescale on which calibration observations are repeated: calibration data which is obtained on timescales longer than a week (for example: **req. 531** *Dark current* is done once per month) should be reduced whenever the data becomes available.

However, for timescales shorter than a week, the scheduling of data reduction for calibration data is determined by the interdependency of various calibration data. Not all calibration data is re-observed every night. Neither is the calibration data observed in the order in which they are dependent on each other within a night. Hence, the order of data reduction should be determined by the dependencies rather than by the order of observations. For example: **req. 542** *Dome flat field* can only be computed after a **req. 541** *Bias* has been computed, while biases may in fact be observed later (the next day)

Fig 3c provides such a more detailed schematic of the data flow in the bias and flat-field pipelines. These are pipelines based on observations which are repeated on short timescales. In this case the dependency graph determines the sequence of processing steps. Hence, from left-to-right, top-to-bottom, this figure serves as an overview of the order of processing of calibration data.

When calibration files are constructed, one should be aware that to construct a given calibration file, other calibration files should be available, as indicated in Fig 3c. Note in particular that ColdPixelMaps are determined from DomeFlatFrames (master dome flats), but are also an optional input while creating a DomeFlatFrame. Also note that NightSkyFlatFrames are optional input of MasterFlatFrame, but that a MasterFlatFrame is required to construct a NightSkyFlatFrame.

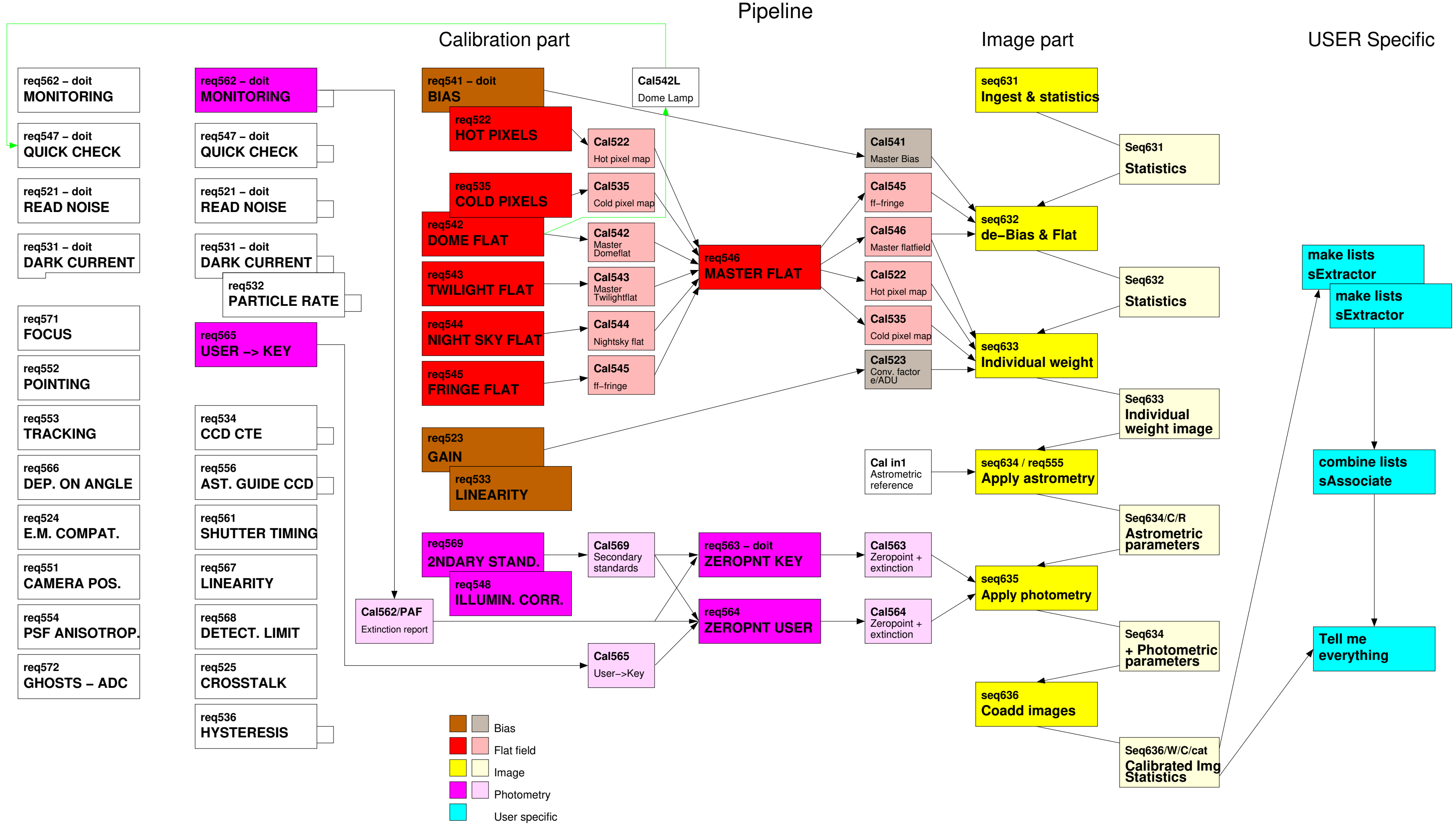


Fig 3b The OmegaCAM complete model, including local quality verifications

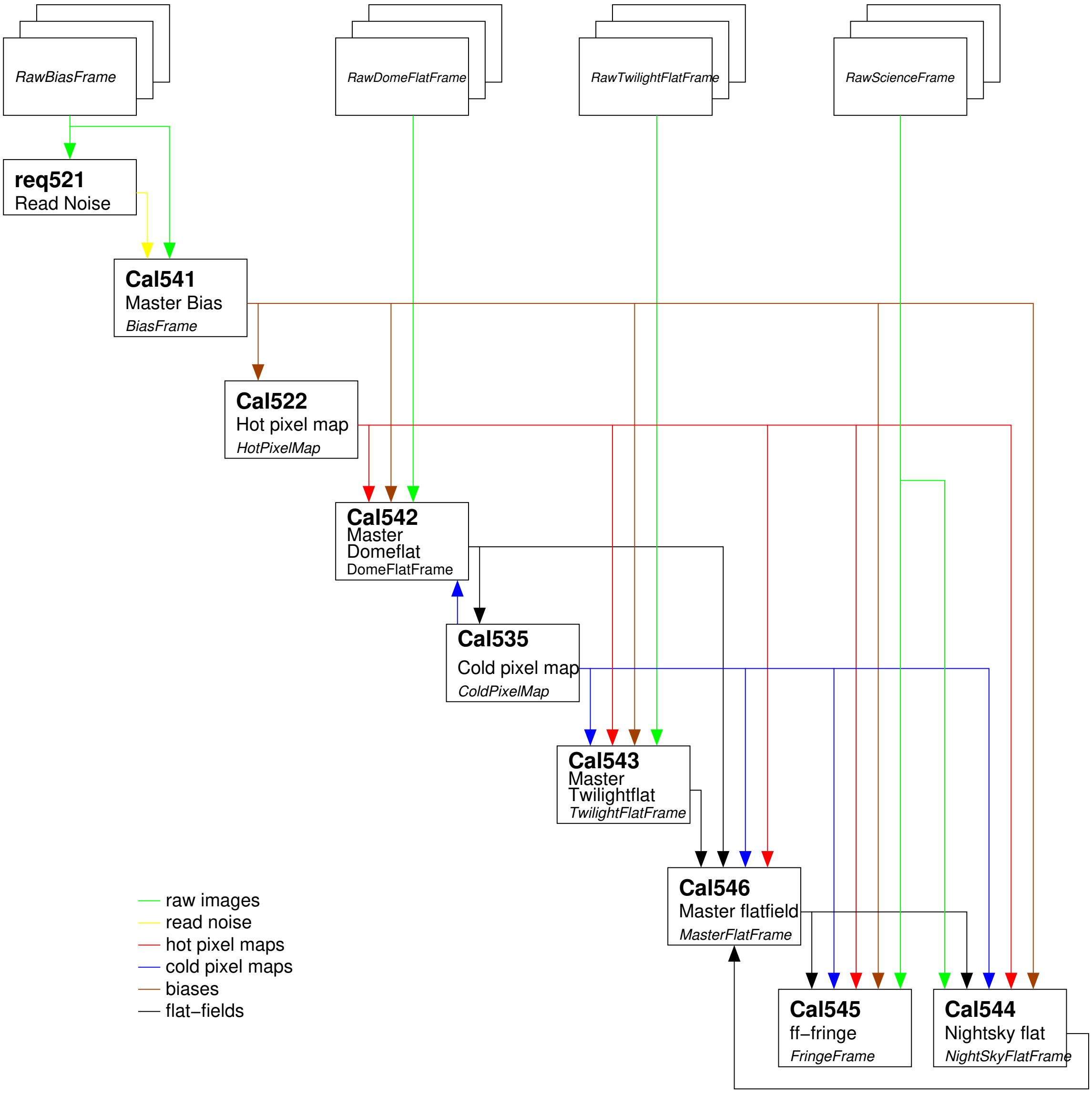


Fig 3c Data flow in calibration part of Pipeline. Code class names are given in italic.

4.0 OBSERVING MODES and STRATEGIES

OmegaCAM employs a fixed set of observing **modes**. These observing modes use a particular dithering method (Section 4.1 URD) and either use a set of **key-filters** or **User-filters** (Section 4.3 URD). The observing modes shall be implemented in **observing templates** (Section 4.4 URD).

Supervising observing **strategies** subscribe additional instructions for the scheduling. The observing strategy is copied into the FITS header, which will not be used by ESOs pipeline, but is optionally used by pipelines outside ESO.

The calibration procedures, often involving a **trend analysis** for qualification/quality control purposes, are designed to support the set of observing modes and employ as much as possible identical observing modes.

A prominent concept is the designation of **keyfilters**, which are used to monitor the behaviour of the instrument and atmospheric transparency: the observational data obtained with the keyfilters will be subject to extensive administration and trend analysis. Contrary to the key-filters, the data obtained with userfilters, in so-called **user mode**, will not be subject to trend analysis. User-mode data are cross-calibrated nightly to the key-filters. Observations taken with the Atmospheric Dispersion Corrector are considered as user mode.

The information about strategies and modes shall be strictly carried through the DFS by FITS-header items(apart from p2pp to schedular).

Glossary of OmegaCAM terms

CA	Calibration Analysis. A high-level specification of the data analysis of a requirement.
CAP	Calibration Analysis Procedure. A detailed, code-level specification of the data analysis of a requirement.
CalFile	The result of one particular step in the calibration part of the pipeline.
CP	Commissioning.
Composite filter	Filter divided into four parts, one part for each **Key band**. Used for photometric calibration.
Key filter	Used for photometric calibration, instrument and atmosphere monitoring.
Key bands	Sloan u', g', r' and i'.
Mode–	Observing technique.
Mode– Dither	Shifts larger than the gaps in the CCD mosaic.
Mode– Jitter	Few pixels shifts between sub-exposures.
Mode– Stare	Independent exposure(s) with identical pointing.

Mode– SSO	Solar System Object / Non-Siderial tracking.
Req	Shorthand for requirement. One specific task the system has to perform.
RP	Routine Phase. Standard operations of **OmegaCAM**.
Seq	One intermediate step in the image reduction part of the pipeline.
SeqFile	The intermediate result of a particular **Seq**.
Strategy–	Observing strategy.
Strategy– Deep	Does deep integrations.
Strategy– Freq	Frequently visits (monitors) the same field.
Strategy– Mosaic	Maps areas of the sky larger than 1°.
Strategy– Standard	Consists of a single observation.
User filter	Any other filter than the four **Key filters** or the **Composite filter**

4.1 Observing Modes

In order to overcome:

i) the effect of the blank columns and rows in between individual CCD chips,

ii) bad (hot or cold) pixels (columns) of the CCD chips,

iii) the effect of cosmic ray events,

dithering (i.e. obtaining very slightly offset images of the same patch of sky) will play a prominent role in the data acquisition and its associated data reduction. Dithering can involve from N = 1 to 11 different positions on the sky.

There are two spatial domains to which the dithering can be optimized:

I **Dither** offsets matching the size of the largest gaps of the CCDs in the focal plane (= above item i).

II **Jitter** offsets matching the smallest gaps in CCDs, e.g. dead columns and bad pixels (= above items ii and iii).

While **dither mode** will provide maximum, complete, sky coverage with a minimum effect of the gaps in between the CCD's, its context map will be very heterogeneous (complex). Conversely, **jitter mode** will result in a very homogeneous context map, but the gaps in between CCDs will remain unobserved and sky coverage is incomplete. Depending on the scientific objectives of the observations the User can choose between **dither mode** with maximum sky coverage (completeness) and **jitter mode** with maximum homogeneity of the acquired data, which is expected to propagate as lower effective noise in the reduced data, as is desired for e.g deep fields.

Both **dither** and **jitter** observing modes will be supported. There will also be a **stare** mode, for which there are no offsets, and a **mosaic** mode, which is used to generate images larger than a single **OmegaCAM** field.

Mode– Dither N= will be operated with N pointings on the sky, with offsets matching the maximum gap between arrays, i.e. $\sim$ 380 pixels ($\sim$ 80 arcsec). In standard observing mode N =5, maximum supported (pipeline!) value of N is 11. **Mode– Dither** N= covers all the gaps in the focal plane, thus the resultant image has no gaps within the field of view, but the context map describing the origin of the various pixels in the resultant map will be very complex. Dither mode is optimized for maximum sky coverage. **Mode– Dither** N= is used for general surveys, which want to achieve complete sky coverage.

Mode– Jitter N= will be normally operated with N=3-5 pointings, with quite small offsets ($\sim$ 5 pixels). Maximum N = 11. **Mode– Jitter** N= is optimized for observations which require maximum homogeneity of the context map, and for which the acquisition of information in the wide CCD gaps is not critical. **Mode– Jitter** N= will have a minimum of discontinuous variations of the PSF over the field of view; also all the data from a single sky pixel originate from a single CCD chip.

Mode– Stare N= will have one fixed pointing position, but the same position of the sky might be re-observed subsequently (N>1). **Mode– Stare** N= can be used to take all kinds of snapshots or calibration frames, but will also be the main workhorse for monitoring optical transients or fast moving objects. Facilities for direct on-line comparison of single **Mode– Stare** N=2 frames both for variable target monitoring and for instrument qualification monitoring will be an important aspect of this mode.

Mode– Mosaic N= will facilitate surveys that aim to image areas of the sky larger than 1° diameter, or which for any other reason want to combine data from two adjacent partly overlapping fields. The prime characteristics of this mode are:

– the coordination of the definition of a set of pointings on the sky - in the OB planning
– the combination of the data from different fields into one data product

In principle, this mode should be handled by the de-dithering (=co-addition) process, (this can be interpreted as a requirement on the co-addition code, namely produce an output image which is the rectangle of the overlapping area of several input images). If the co-addition (particularly the set of quality control parameters) and the source extraction methods are handled correctly, this mode will form a natural extension of the **dither mode** and does not require additional developments. Maximum image size supported for this mode is TBD; for mosaiced source lists the maximum covered sky area is unconstrained and data volumes could well run into the Tbyte regime. It is not planned for the standard DFS pipeline to produce mosaiced images.

Mode– SSO N= will support the data taking of Solar system objects with a non-siderial tracking of the telescope. No-autoguiding is required for this mode as in general the uncertainties of the predicted angular velocities of yet to be discovered objects are larger than the expected jitter of the telescope when the autoguiding is disabled. This mode resembles very much the **Mode– Stare** N=, however the wanted quick (<24 hour) data reduction for many fast moving targets implies that the DFS pipeline will not support the processing of these data fully. However, for observations which require a slower response (4 times 24 hour or longer) or which only require de-biasing and flatfielding, the standard pipeline can be configured to handle these data.

4.2 Observing Strategies

An **observing strategy** employs one of the basic **observing modes** and defines a number of additional instructions for specifically the scheduling of the observations. The observing strategy shall be recorded in the FITS headers of the observations. Optionally, this header information can be used in data reduction pipelines, particularly those operated by the Consortium when addressing the combination (e.g. stacking) of images. It is not expected that the ESO pipeline will recognize strategies, as the ESO pipeline will not combine various runs.

We discriminate strategy:

Strategy– standard which consists of a single observation (observation block),

Strategy– deep which does deep integrations, possibly taken at selected atmospheric conditions over several nights; the standard image pipeline will not combine images taken over several nights.

Strategy– freq which frequently visits (monitors) the same field on timescales ranging from minutes to months and has overriding priority on the telescope schedule

Strategy– Mosaïc maps areas of the sky larger than 1°, which is essentially an item for the scheduling, as the pipeline has to produce uniform quality data anyway. The combination of various field centers into one image is not considered a standard pipeline task.

4.3 Filtering

OmegaCAM employs the following type of filters (see Section 3.4 for details):

- Broad-band filters (Johnson B,V and Sloan set)
- Intermediate band (e.g. Stromgren and deep sky filters 200A)
- narrow band (e.g. O[III], Hα)

A set of **keyfilters** is defined and will be used for the photometric calibration. These **keybands** will be used for monitoring the instrument and the atmosphere at timescales from hours up to years. The keyfilters will be used for standard, beginning of the night checks, the monitoring of the atmospheric extinction and for a variety of trend analyses relating to the photometric calibration of the instrument. The photometric system of the instrument shall be continuously maintained at these **keybands**. One **composite keyfilter** will be available, which will contain in each quadrant a different passband, namely the four different keybands. The composite filter will be extensively used for quick checks, extinction measurements and other monitoring of the photometric system.

The observing modes related to the photometric calibration (**req.**'s 5.6) will employ the concept of keyfilters. The other filters (**Userbands**) will be cross calibrated versus the keyfilters (see **req.**565). Monitoring of the atmospheric extinction will be done with the keyfilters (**req.**562).

The 4 keyfilters are Sloan u' g', r' and i' with passbands very close to, but not identical, to the monolythic keyfilters.

A photometric cross calibration between the monolythic and composite keyfilters will be necessary is formulated under **req. 565** *Filter bandpasses*.

4.4 Observing Templates

The Observing Templates for **OmegaCAM** are defined in Instrument Software Functional Specifications (VST-SPE-OCM-23100-3064). This definition is based on the requirements as set out in the URD. The observation modes described in the previous section imply three basic OTs for science observations.

- **TSF– OCM_img_obs_dither**
  Observe with N pointings (default 5) in the sky, with offsets $>$ maximum gap between detectors ($>$ 390 pixels) between exposures.
- **TSF– OCM_img_obs_jitter**
  Observe with N pointings (2-5) in the sky, with small offsets ($\approx$ 5 pixels) between exposures.
- **TSF– OCM_img_obs_stare**
  Observe N (default 1) exposures with one fixed pointing position.

4.5 Field correctors

The VST and **OmegaCAM** employ two different field correctors:

Two lenses field corrector: In the U–I bands (0.320–1.014 μm) the worst case (edge of field) diffraction encircled 80% energy diameter is 1.33 pixel at Zenith. The transmission is 93%.

One lens plus Atmospheric dispersion corrector - ADC: In the B–I bands (0.365–1.014 μm) the ADC produces a diffraction encircled 80% energy diameter of 2.0 pixels (worst case at edge of field) at a Zenith distance of $50°$. The transmission is 88%. At $70°$ Zenith angle this is 2.88 pixel (2.63 average over the field). The ADC cannot be used in the U band.

For observations near Zenith the two-lens corrector will be preferred because of its higher throughput. At larger zenith distance (value depends on wavelength), the ADC will yield better imaging performance.

The standard (=key) observing mode is with the two-lens corrector.

The ADC observing mode is considered a User mode.

5 BASELINE CALIBRATION REQUIREMENTS

5.0 Documentation system, Odoco

The trajectory of the data through the DFS shall be guided by the **OmegaCAM** documentation system, **Odoco**. The set-up of the specifications of the calibrations and its data reduction is done in purely requirement driven fashion.

The calibration documentation system (**Odoco**) is a collection of files which contains a description of both the requirements on **OmegaCAM** calibrations, as well as detailed descriptions of the envisaged Calibration Observations (**CO's**) and their analysis and data-reduction. The **Odoco** is meant to avoid unnecessary duplication of work for the documentation of the various stages of the project; from the definition of basic requirements, to implementations and final operation and user manuals. It should also aid the documentation of development work. The **Odoco** was originally developed for ISO and provided a full uplink system (IOCD). The present version, adapted for **OmegaCAM**, contains essentially only the part which deals with text, pseudo code, recipe's and automatic document creation. The **Odoco** system is essentially a set of TeX macros, with LaTeX emulation, together with some supporting C routines. The whole present document is generated by **Odoco**, but particularly this section, listing the baseline calibration requirements, uses some more advanced **Odoco** options.

The very strict document control of the original IOCD will not be maintained, but **CVS** (Concurrent Versions System) will be used for local versioning control. Official versions of documents shall be filed separately.

The contents of the **Odoco** will be continuously evolving and fonts are chosen for optimal reading on a computer monitor screen.

In **Odoco**, calibrations are specified under requirement subsections (**req.**'s) which are labeled with 3-digit numbers. Each subsection contains a number of items: e.g. the **objective** of the requirement, a description of a specific Calibration Observation or a cross-reference to the use of a CO that has been defined under another requirement. Also, the end-results have been specified and the text contains various descriptions for both Template Signature Files (**TSF**'s) which define the creation of the observation blocks (**OB**) and for the off-line data analysis. Overall priorities have been defined (essential, very important, desirable) and are specified under the item **priority**.

The chosen items for the descriptions of the requirements (**req.**'s) match well to the items needed for the **recipes** for DFS data reductions. In section 6.1 complete listings of both the **req.** items and the **recipe** items are given. The **req.**'s as listed in the **Odoco** will eventually evolve into the deliverable recipe's.

The **Odoco** is designed to provide a comprehensive and accessible documentation system of the various activities that relate to the **OmegaCAM** calibrations. It serves a variety of purposes, and facilitates the extraction of text from the **Odoco** data base into complete documents. The **Odoco** can provide the following documents:

1. **A listing of the Baseline OmegaCAM Calibration Requirements. Odoco** contains an up-to-date listing of all the baseline requirements for the **OmegaCAM** calibrations, i.e. for each requirement (**req.**) the text under the items: **Objective, When performed/frequency, Required accuracy, Priority**. See section 5 of the URD.
2. **Full documentation of the OmegaCAM calibration plan**. A detailed description of all the **OmegaCAM** calibration requirements and their implementations. See Section 5 of the Calibration Plan. Summary sections (two digit sections) have been introduced for a variety of calibration activities: e.g. detector specific calibrations, photometric calibrations etc. A general overview of the **OmegaCAM** calibrations can be obtained by printing the summary sections of the **Odoco**. In order to further ease the readability of this document, both each requirement and each calibration analysis procedure text item begins with a 'one-liner' stating the overall idea.
3. **A description of the Template Signature File necessary to produce observation blocks, TSF's.** When a requirement can not be fulfilled by means of data analysis of observations made for another requirement, **Odoco** contains a detailed description of the instrument configuration and procedures under the items **Sources, observations** and **TSF**, (TSF, Template Signature File). Note, the term **selfstanding** has an important meaning: when a requirement is selfstanding, it will normally propagate as a single dedicated calibration observation, with a single dedicated data reduction task. Conversely, non-selfstanding requirements will have more complex dependencies and often involve a data reduction of data taken for another selfstanding requirement.
   Under the item **TSF** the hierarchical structure of observation specifications is detailed (when applicable) on different lines:
   - first line: observing **Strategy**–
   - second line: observing **Mode**– N=
   - third line: generic/base **TSF**–
   - fourth line: specific/dependent == **TSF**–
4. **Description of Calibration Analysis (CA).** For each requirement, a specification of the data analysis related to the requirement is given under the item **CA**. Standard functionalities can be quoted in the optional item **Needed functionalities for CA**. A detailed description of the implementation, which could include guidelines for the data analysis or pseudo code is given under the items **CAP** (Calibration Analysis Procedure). **Inputs and Outputs** defines the various calibration tables. Thus a document listing all the text of the items **CA**, **CAP** and **Inputs and Outputs** gives a complete overview of the Calibration data reduction analysis.

5. **A reference document for timelining** OmegaCAM **Calibration observations.** The items **When performed/frequency and Estimated time needed** can be used to design a detailed **Schedule** of calibration observations both in the commissioning phase (**CP**) and during the Routine Phase (**RP**).
6. **A listing of the various requirements for an Astronomical Calibration Source data base, (ACS-dbase).**
7. The **recipes** belonging to the execution of requirements.

As **Odoco** can provide various documents with a different filtering of the source of information, each printout contains a table, listing the selection criteria. Also, the status of the printout is marked (formal issue, or private workcopy). Each printout contains this section.

On the following pages a print-out is included which is believed to be relevant for the present document.

OmegaCAM

OmegaCAM Data Flow System - Users and programmers manual

User manual for ESO-DFS

Issue: Version 2.11

Date: 23 Jul 2004

Prepared by: Rengelink, Boxhoorn, Deul, Kuijken, Valentijn, Vermeij, Begeman, Helmich

Purpose of printout: PAE THIRD DELIVERY

Selected items from Odoco file system	
Summary sections	•
Items:	
Objective	•
Fulfilling or fulfilled by	•
When performed/frequency	•
Sources, obs.,. . .	•
Inputs	•
Outputs	•
Required accuracy, constraints	•
Estimated time needed	•
Priority	•
Template Signature File.	•
Recipe	•
Calibration Analysis spec's	•
Needed Functionalities	•
CA implementation (pseudo code)	•
Status of Req.	
FLAG	

www.astro.rug.nl/∼omegacam

LAY-OUT of BASIC CALIBRATION REQUIREMENTS DETECTOR RELATED

5.1	**Functional Checks**	
5.2	**Electronics specific**	
5.2.1	CCD read noise - *doit*	**CalFile– 521**
5.2.2	CCD hot pixels	**CalFile– 522**
5.2.3	CCD gain	**CalFile– 523**
5.2.4	Electromagnetic compatibility	
5.2.5	Electrical cross talk	
5.3	**Detectors specific**	
5.3.1	CCD Dark current - *doit*	**CalFile– 531**
5.3.2	CCD Particle event rate	**CalFile– 532**
5.3.3	CCD linearity	**CalFile– 533(PM)**
5.3.4	CCD Charge transfer efficiency	**CalFile– 534**
5.3.5	CCD Cold pixels	**CalFile– 535**
5.3.6	CCD hysteresis	**CalFile– 536**
5.4	**Detectors operational**	
5.4.1	Bias - *doit*	**CalFile– 541**
5.4.2	Flat field - Dome	**CalFile– 542(L)**
5.4.3	Flat field - Twilight	**CalFile– 543**
5.4.4	Flat field - Night sky	**CalFile– 544**
5.4.5	Flat field - fringing	**CalFile– 545**
5.4.6	Flat field - master flat	**CalFile– 546**
5.4.7	Quick detector check	**CalFile– 547(r)**
5.4.8	Illumination correction	**CalFile– 548**

LAY-OUT OF BASIC CALIBRATION REQUIREMENTS cont'd

5.5	**Astrometric**	
5.5.1	Focal plane position of camera	
5.5.2	Telescope Pointing	
5.5.3	Telescope and rotator tracking	
5.5.4	PSF anisotropy	**CalFile– 554**
5.5.5	The astrometric solution - *doit*	**SeqFile– 634(CR)**
5.5.6	Astrometry - Guide CCD's	**CalFile– 556**
5.6	**Photometric**	
5.6.1	Shutter timing	
5.6.2	Photometric monitoring	**CalFile– 562(T)**
5.6.3	Zeropoint key bands *doit*	**CalFile– 563**
5.6.4	Zeropoint user bands	**CalFile– 564**
5.6.5	Filter band pass - user– >key	**CalFile– 565(C)**
5.6.6	Rotation angle - ADC, rotator	
5.6.7	Linearity	
5.6.8	Detection limit	
5.6.9	Secondary standards	**CalFile– 569**
5.7	**Alignment**	
5.7.1	Camera focus/tilt	
5.7.2	Ghosts - ADC	
5.8	**Telescope**	
5.9	**End to end tests**	

5.1 Functional Checks

The commissioning plan [VST-PLA-OCM-23100-3100] lists a large set of acceptance tests. Most of these test are engineering tests and are not repeated here. Those engineering tests which can be executed with requirements from the URD/CP contain the proper reference to the URD in the commissioning plan.

The technical operations and maintenance plan [VST-PLA-OCM-23100-3080] describes activities during **Routine Phase (RP)**. Both the URD, CP and the DRS specifies the requirements and the activities for fulfilling these requirements during RP. The label RP always points to activities that shall be followed in the technical operations and maintenance plan.

5.2 Detector Electronics specific Calibrations

Section 5.2 contains the requirements for the characterization of the detector system on electronics level, while Section 5.3 lists more detector specific calibrations. The separation between these Sections is somewhat artificial. In Section 5.4 more daily characterizations are listed, which involve the flatfielding and de-biasing. Requirements which are foreseen to become 'workhorses' are labeled with *doit*, e.g. a quick daily evaluation of the read noise serves as a daily health check.

The CCD's are operated at one port; **electrical cross talk** is not expected to significantly affect the observations. However, as a check, the absence of significant cross-talk is verified (**req.**525).

The **hot pixel** (**req.**522) and **cold pixel** (**req.**535) characterization are combined in the **individual weight maps** (**seq.– 633**).

As the standard read-out time of the arrays is already very fast, $\sim$ 40 sec , no extra fast **read-out mode** is supported in the characterization.

CCD **rebinning mode** and **windowing mode** are not supported in the calibration and data reduction procedures.

5.2.1 Req. — *Cat I: CCD read noise - doit*

Objective:

Measure the CCD read noise (in ADU's) as a standard health check.

The read noise is measured from pairs of bias exposures. The rms scatter of the differences between two exposures is computed and divided by $\sqrt{2}$. Monitor variations. This is the first order daily health check.

Fulfilling or fulfilled by:

Selfstanding

When performed/frequency:

daytime- Commissioning, during all operations: daily health check.

Required accuracy, constraints:

Readout noise less than 5e$^-$.

Variation in readout noise w.r.t. reference value less than 0.5e$^-$.

These are lab values. The corresponding limits in ADU can be calculated using the e$^-$/ADU conversion factor from **req.**523.

Estimated time needed:

Observation: 5 min. Reduction: 5 sec/CCD.

TSF:

Mode– Stare N=2
(**TSF– OCM_img_cal_bias,** N=2)
= **TSF– OCM_img_cal_readnoise**

Inputs:

2 raw bias frames

Outputs:

CalFile– 521 *Readout noise* in ADU's
The **CalFile** correponds to QC parameter read_noise (a single number).

Recipe:

```
Read_Noise -i bias1 bias2 [-max MAXIMUM_ITERATIONS]
                          [-rej REJECTION_THRESHOLD]

bias1, bias2         : the two raw bias images
MAXIMUM_ITERATIONS   : maximum number of iterations for statistics
                       measurement (integer).
                       Range of allowed values: 2 - 10. Default: 5
REJECTION_THRESHOLD : rejection threshold for bad pixels in sigma (float).
                       Range of allowed values: 1.0 - 10.0. Default: 5.0
```

Before applying this recipe, use **Recipe– Split**—which is documented in **seq.– 631**—with the `-t bias` option to split the raw multi-extension FITS input files.

CA:

Process (make):

1. Subtract the two bias frames to produce a difference image.
2. Iteratively reject outliers in the difference image.
3. Compute the mean, median and rms of the remaining pixels of the difference image.
4. The readout noise is the rms divided by $\sqrt{2}$.

CAP:

```
diff_image = eclipse.image_sub(bias1, bias2)
stats = eclipse.iter_stat(diff_image,
                          MAXIMUM_ITERATIONS,
                          REJECTION_THRESHOLD)
read_noise  = stats.stdev / sqrt(2)
mean_diff   = stats.avg_pix
median_diff = stats.median
```

QC Parameters:

```
read_noise  : the readout noise in ADU
mean_diff   : mean difference between the two input raw bias frames
median_diff : median difference between the two input raw bias frames
```

5.2.2 Req. — *Cat I: Hot pixels*

Objective:
Determine CCD bad/hot pixels.

5σ outliers in the master bias frame are bad-hot pixels. These pixels should be recorded and ignored (assigned a weight of 0) in dedithering and dejittering, as well as source extraction. For this purpose the bad/hot pixel map is used to assign a weight of zero to the affected pixels in the weight map (**seq.– 633**). The search for hot pixels would also identify traps.

Fulfilling or fulfilled by:
Additional data reduction of **CalFile– 541** *Master Bias frame* to determine cold pixels

When performed/frequency:
daytime- Commissioning, in RP twice per week.

Required accuracy, constraints:
Number of hot pixels to be determined by experience/lab values.
The total number of bad pixels (hot pixels + cold pixels) is less than 80000 (checked in **req. 535** *Cold pixels*
Difference in number of hot pixels w.r.t. reference value, less than 100.

Estimated time needed:
Observation: None. Reduction: $<$ 20 sec./CCD.

TSF:
Use master bias (**req.**541)

Inputs:
CalFile– 541 *Master Bias frame*

Outputs:
CalFile– 522 *Bad/hot pixel map, number of hot pixels*

Recipe:
```
Hot_Pixels -i master_bias [-max MAXIMUM_ITERATIONS]
                          [-rej REJECTION_THRESHOLD]

master_bias          : master bias image
MAXIMUM_ITERATIONS   : maximum number of iterations for statistics
                       measurement (integer).
                       Range of allowed values: 2 - 10. Default: 5
REJECTION_THRESHOLD  : rejection threshold for bad pixels in sigma (float).
                       Range of allowed values: 1.0 - 10.0. Default: 5.0
```

CA:
Processing (make):
1. Iteratively estimate statistics ($mean$, σ) of the input bias.

2. Construct a pixelmap using thresholds ($-\infty$, $mean + 5\sigma$).
3. Count the number of bad pixels in the pixelmap.

CAP:

```
stats = eclipse.iter_stat(bias,
                          MAXIMUM_ITERATIONS,
                          REJECTION_THRESHOLD)
threshold = stats.avg_pix + REJECTION_THRESHOLD * stats.stdev
pixelmap = eclipse.threshold2pixelmap(bias, -1E20, threshold)

# eclipse pixelmap counts good pixels
count = pixelmap.lx * pixelmap.ly - pixelmap.count
```

QC Parameters:

count : the number of bad pixels

5.2.3 Req. — *Cat I: CCD gain*

Objective:

Determine CCD gain and variation with time
Determine the conversion factor between the signal in ADU's supplied by the readout electronics and the detected number of photons (in units e^-/ADU) and monitor variations in time.
The gain factors are needed to convert ADU's in raw bias-corrected frames to the number of electrons, i.e. detected photons.
Take two series of 10 dome flatfield exposures with wide range of exposure times. Derive the rms of the differences of two exposures taken with similar exposure (integration time). Exposure differences of pairs should not exceed 4%. The regression of the square of these values with the median level yields the conversion factor in e^-/ADU (assuming noise dominated by photon shot noise).

Fulfilling or fulfilled by:

Selfstanding, also measures detector linearity (**req.**533).

When performed/frequency:

daytime- Commissioning, in RP once week.

Sources, observations, instrument configurations:

Dome flat field exposures on lamp, with t_{exp}= 2, 60, 50, 4, 8, 40, 30, 1, 16, 24, 24, 16, 1, 30, 40, 8, 4, 50, 60 and 2 seconds. Use r' filter.

Required accuracy, constraints:

Accuracy: In units of e^-/ADU, from lab values or found empirically. Variation in time less than 1%.

Estimated time needed:

Observations: 1 hour. Reduction: 3 min./CCD.

TSF:

Mode– Stare N=20
(**TSF– OCM_img_cal_domeflat,** t_{exp}= 2, 60, 50, 4, 8, 40, 30, 1, 16, 24, 24, 16, 1, 30, 40, 8, 4, 50, 60, 2 s.).
= **TSF– OCM_img_cal_gain**

Inputs:

Number of raw domeflats such that for each exposure time there are two different raw domeflats. The total number of raw domeflats has to be larger than 4.
A raw domeflat with a short exposure time.
A raw domeflat with a long exposure time.
CalFile– 541 *Master Bias frame*

Outputs:

CalFile– 523 *Conversion factor e^-/ADU*
The **CalFile– 523** corresponds to the QC parameter gain (single number).

Recipe:

```
Gain -i <raw_domeflat_list> -b master_bias -s short_raw_domeflat
                                           -l long_raw_domeflat
                                           [-oc OVERSCAN_CORRECTION]

raw_domeflat_list   : list of even number of raw dome flats
master_bias         : master bias frame
short_raw_domeflat  : raw dome flat - short exposure time
                      One of <raw_domeflat_list> with an exposure level
                      of about 1000 ADU
long_raw_domeflat   : raw dome flat - long exposure time
                      One of <raw_domeflat_list> with exposure level
                      of about 20000 ADU
OVERSCAN_CORRECTION : overscan correction mode (integer).
                      Description of allowed values:
                        0: apply no overscan correction (default)
                        1: use median of the prescan in the
                           x-direction
                        2: use median of the overscan in the
                           x-direction
                        3: use median of the prescan in the
                           y-direction
                        4: use median of the overscan in the
                           y-direction
                        5: use the per-row value of the prescan in
                           the x-direction
                        6: use the per-row value of the overscan in
                           the x-direction
```

Also satisfies **req. 533** *CCD Linearity*. Before applying this recipe, use **Recipe– Split**—which is documented in **seq.– 631**—with the `-t dome` option to split the raw multi-extension FITS input files.

CA:

Assume bias corrected dome flats.
nb: monitoring in time not implemented in recipe
Process (make):
1. Divide the observations into pairs of equal exposure time
2. Trim, overscan-correct, and de-bias each exposure.
3. For each pair of exposures produce a sum and a difference image.
4. For each sum image measure the median
5. For each difference image measure the standard deviation (reject 5σ outliers).
6. Perform a minimum least-squares fit of squared standard deviations (variances) as a

function of the medians divided by 2.
7. The fitted slope is the gain. The offset should be equal to the read noise[2]

CAP:

```
domeflat_pairs = make_pairs(domeflats)
rms_diff = []
median_sum = []
exptimes = []
measurements = []

for pair in domeflat_pairs:
    exptime = eclipse.header(pair[0]).get_key('EXPTIME')
    exptimes.append(exptime)

    ima1 = eclipse.trim_and_overscan(pair[0])
    eclipse.image_sub_local(ima1, bias)

    ima2 = eclipse.trim_and_overscan(pair[1])
    eclipse.image_sub_local(ima2, bias)

    exptimes.append.eclipse

    sum = eclipse.image_add(ima1, ima2)
    median = eclipse.image_getmedian(sum)
    median_sum.append(median/2)

    diff = eclipse.image
    stats = eclipse.iter_stat(diff, MAXIMUM_ITERATIONS, REJECTION_THRESHOLD)
    rms_diff.append(stats.stdev)


    measurements.append(median/2, stats.stdev**2)

coeffs, errors = least_squares_fit(measurements)
gain   = coeffs[1]
```

QC Parameters:

```
gain      : The slope fitted to the median vs. rms of the differences
            of the input images
```

5.2.4 Req. — *Cat III: Electromagnetic Compatibility*

Objective:

Verify whether any external source (e.g. dome drives, control systems) is not interfering in the CCD overall detector system, leading to additional, mostly non-white noise.
Technical specifications require less than 20% effect on read-out noise, for external interference and less than 10% effect on read-noise for internal **OmegaCAM** interference.
If electronic interference occurs then this will put constraints on the operation of the instrument. For example, if interference occurs during movement of the telescope, one cannot read the CCDs and move the telescope at the same time,.
Interference is detected by measuring the read noise (**req.**521) under operational conditions. This means doing bias measurements while the telescope and/or dome are moving.

Fulfilling or fulfilled by:

repetition of CCD read noise calibrations **req.**521.

When performed/frequency:

Day time; Commissioning; once a year; every time a major system change has been made; To be determined by experience

Sources, observations, instrument configurations:

Raw bias frames obtained while the telescope/dome are moving.

Required accuracy, constraints:

Difference between read noise under operational conditions and the standard read noise measurement should be smaller than 20% for external and 10% for internal causes of interference.

Estimated time needed:

Observations: 4 hours. Reduction: 1 min./CCD.

TSF:

Mode– Stare N=2
(**TSF– OCM_img_cal_bias,** N=2)
= **TSF– OCM_img_cal_readnoise** while telescope and/or dome are moving.

Inputs:

raw bias frames, obtained when telescope/dome were moving.

Outputs:

OK/non-conformance flag.

CA:

Run recipe read noise and compare output of read noise recipe **CalFile– 521** *read-out noise* (see **req.**521 for implementation details). The difference should be measured with an accuracy of 5%.

CAP:

(see **req.**521 for implementation details)

5.2.5 Req. — *Cat III: Electrical cross talk*

Objective:

Although crosstalk is not detectable in the WFI, and only one part per CCD is used, the sharing of one FIERA by 16 CCD's opens up the possibility of cross talk.
Observe a bright (mag 5-8) star at 16 different chips (1 FIERRA serves 16 chips).

Fulfilling or fulfilled by:

Selfstanding

When performed/frequency:

Nighttime. Commissioning.

Sources, observations, instrument configurations:

Bright star, mag 5-8

Required accuracy, constraints:

10^{-5}

Estimated time needed:

5 minutes

TSF:

Mode– Stare N=1
(**TSF– OCM_img_obs_stare,** N=1, filter=key band)

Outputs:

Conformance flag

CA:

Inspect on RTD; simultaneously CTE on bright stars can be inspected **req. 534** *CTE*

5.3 Detectors specific calibrations

5.3.1 Req. — *Cat I: CCD Dark Current - doit*

Objective:

Measure CCD dark current (in ADU/pixel/hour) for qualification purposes of the detector chain (qualification and trend analysis). The particle event rate will be determined on the fly.
Repeating the test with the dome lights on will provide information on possible light leaks.
Three one hour exposures are taken with the shutter closed. After rejection of the cosmic ray events, the signal above the bias level is the dark signal.
For the reduction of the science observations the subtraction of the sky brightness will include the dark current, and a separation of both contributions is normally not required.
Do a trend analysis.

Fulfilling or fulfilled by:

Selfstanding, also used to compute the particle event rate.

When performed/frequency:

Daytime (if dome and camera are proven to be light tight enough) - Commissioning; once per week. Alternatively, one dark frame per day could be taken, followed by a trend analysis once/month.

Sources, observations, instrument configurations:

Three 1 hour exposures with shutter closed. Dome lights either on or off for all exposures.

Required accuracy, constraints:

Dark count rate should be less than the equivalent of 3 e^-/pixel/hour in ADUs excluding bad pixels. (This corresponds to 1.5 ADU/pixel/hour for nominal gain.)
Accuracy of determining particle event rate 1 particle/cm^2/hour.
Particle event rates should be identical for each chip.

Estimated time needed:

Observation: 3 hours. Reduction: $<$ 1 min./CCD.

TSF:

Mode– Stare N=3
TSF– OCM_img_cal_dark exposure time 1 hour each, shutter closed. Dome lamp either on or off for all exposures

Inputs:

3 raw dark frames
CalFile– 541 *Master Bias frame*
Lab values

Outputs:

CalFile– 531 *Dark count rate for each CCD* in ADU/pixel/hour
CalFile– 532 *Particle event rate* in particles/cm^2/hour
In ESO DFS terminology these outputs are not a **CalFile**, but a single number.

Recipe:

```
Dark_Current -i dark1 dark2 dark3 -b bias [-oc OVERSCAN_CORRECTION]

dark1 dark2 dark3   : 3 raw dark frames
bias                : master bias frame
OVERSCAN_CORRECTION : overscan correction mode (integer).
                      Description of allowed values:
                        0: apply no overscan correction (default)
                        1: use median of the prescan in the
                           x-direction
                        2: use median of the overscan in the
                           x-direction
                        3: use median of the prescan in the
                           y-direction
                        4: use median of the overscan in the
                           y-direction
                        5: use the per-row value of the prescan in
                           the x-direction
                        6: use the per-row value of the overscan in
                           the x-direction
```

Before applying this recipe, use **Recipe– Split**—which is documented in **seq.– 631**—with the `-t dark` option to split the raw multi-extension FITS input files.

CA:

Process (make):
1. Trim, de-bias and overscan correct the input images
2. Median average the 3 debiased images.
3. Iteratively reject 5 σ outliers in the median image.
4. Compute the mean of the resulting image.
5. Use this mean to compute the dark current in ADU/pixel/hour.
6. For each de-biased and overscan corrected input image:
6.1. Use sextractor with a cosmic retina filter to create a catalog.
6.2. Count the number of objects in the catalog.
7. Use the event counts to compute the particle event rate in particles/cm^2/hour.

CAP:

```
Constants:
  MAXIMUM_ITERATIONS            : statistics measurement (integer)
                                  Value: 3
  REJECTION_THRESHOLD           : sigma rejection for bad pixels (float)
```

```
                                     Value: 5.0
  PIXELSIZE                        : OmegaCAM pixelsize in cm (float)
                                     Value: 15e-4
  COSMIC_CONFIG : SExtractor configuration parameters and their values
                  in case they differ from the SExtractor defaults.
          CHECKIMAGE_TYPE = 'SEGMENTATION'
          ANALYSIS_THRESH = 3.0
          CATALOG_TYPE    = 'ASCII'
          DETECT_MINAREA  = 1
          DETECT_THRESH   = 6.0
          GAIN            = 1
          MEMORY_PIXSTACK = 100000
          BACKPHOTO_THICK = 24
          BACK_TYPE       = 'AUTO'
          BACK_VALUE      = 0.0, 0.0
          FILTER          = 'Y'
          FILTER_NAME     = 'cosmic.ret'    # Retina filter for SExtractor
          CHECKIMAGE_NAME = 'cosmic.dat'   # Image with cosmic ray events


# preprocessing
processed = []
darks = [dark1, dark2, dark3]
for dark in darks:
    ima = eclipse.trim_and_overscan(dark)
    eclipse.image_sub_local(ima, bias)
    processed.append(ima)

# compute dark current
median_image = eclipse.cube_average_median(processed)
stats = eclipse.iter_stat(median_image,
                          MAXIMUM_ITERATIONS,
                          REJECTION_THRESHOLD)
exptime = dark1.EXPTIME
dark_current = stats.avg_pix / exptime * 3600.0

# compute the cosmic ray counts
event_counts = []
particle_event_rate_sum = 0.0
for ima, dark in zip(processed, darks):
    sextractor(ima, COSMIC_CONFIG)
```

```
    count = line_count('cosmic.dat')
    event_counts.append(count)
    count /= dark.EXPTIME/3600.0
    count /= image.xsize()*PIXELSIZE*ima.ysize()*PIXELSIZE
    particle_event_rate_sum += count

# average the individual particle event rates
particle_event_rate = particle_event_rate_sum/3
```

QC Parameters:

```
dark_current        : dark current for each CCD in ADU/pixel/hour
particle_event_rate : particle event rate in particles/cm^2/hour
```

5.3.2 Req. — *Cat I: CCD Particle Event Rate*

Objective:

Determine CCD particle event rate by evaluating dark current measurements.
Verify the absence of a local radiation source affecting the detector. The data will be inspected for significant differences of the rates on different chips, and will be screened for local effects.

Fulfilling or fulfilled by:

Data reduction of **req. 531** *Dark current*

When performed/frequency:

Commissioning and when dark current is measured.

Required accuracy, constraints:

better than 1 particle/cm^2/hour

Outputs:

CalFile– 532 *Particle event rate* in particles/cm^2/hour

Recipe:

Recipe– Dark_Current, see **req. 531** *Dark current*

CA:

See **req. 531** *Dark Current*

CAP:

See **req. 531** *Dark Current*

5.3.3 Req. — *Cat I: CCD Linearity*

Objective:

Characterize the linearity of the system over the full dynamic range of the A/D converter.

Both the overall absolute linearity of the system and the pixel-to-pixel variation in linearity are of interest.

The overall linearity of the system can be obtained by measuring the counts as a function of exposure time for a series of dome flats. The data to use for this can be the same as for the measurement of the Gain (**req.**523, q.v.)

The pixel-to-pixel variation in the linearity is obtained by dividing a flatfield with a mean exposure level of more than 30000 ADU by a flatfield with an exposure level of less than 1000 ADU. Pixels that deviate more than 5σ from the mean, in this divided image, have an anomalously high nonlinearity. This map of nonlinear pixels may be used in conjunction with the hot and cold pixel maps to produce a map of bad pixels. One can use suitable long and short exposures from the measurements of the Gain (**req.**523, q.v.) for this.

In addition, during a cloudy night, once per year, the linearity will be checked by taking various exposures with a variety of exposure times on the dome screen. These data will be subject to interactive analysis.

Fulfilling or fulfilled by:

Data reduction of **req. 523** *CCD Gain*

When performed/frequency:

daytime- Commissioning, in RP once per month, dark dome test once/year

Required accuracy, constraints:

better than 1% on the photometric scale

Estimated time needed:

Observation: None

TSF:

Use same raw data as for **req. 523** *CCD gain*

Inputs:

Raw dome flatfields
CalFile– 541 *Master Bias frame*

Outputs:

CalFile– 533P *CCD Linearity Plot* A measure of the overall nonlinearity can be obtained from this plot

CalFile– 533M *CCD Linearity map* A map of (anomalously) non-linear pixels can be used in conjunction with hot and cold pixel maps.

Recipe:

Recipe– Gain, see **req. 523** *CCD gain*

CA:

The gain recipe (**req.**523, q.v.) takes a set of raw dome flats with exposure times of 2, 60, 50, 4, 8, 40, 30, 1, 16, 24, 24, 16, 1, 30, 40, 8, 4, 50, 60 and 2 seconds, and records an average of the median for each pairs of images with the same exposure time. One can plot this median vs. the exposure time to asses the overall linearity of the CCD.

For the linearity map (**CalFile– 533M**) use:

Process (make)
1. Assume a long and short raw dome flat exposure not affected by global non-linearity issues (for example: numbers 6 and 4 of the sequence).
2. Trim, overscan correct and de-bias the long and short exposure.
3. Divide the long exposure by the short expoure.
4. Iteratively estimate statistics ($mean$, σ).
5. Construct a pixelmap using thresholds ($mean - 5\sigma$, $mean + 5\sigma$).
6. Count the number of bad pixels in the pixelmap.

CAP:

```
Constants:
  MAXIMUM_ITERATIONS  : statistics measurement (integer)
                        Value: 3
  REJECTION_THRESHOLD : sigma rejection for bad pixels (float)
                        Value: 5.0

# div = (long-bias)/(short-bias)
div = eclipse.trim_and_overscan(long)
eclipse.image_sub_local(div, bias)
short = eclipse.trim_and_overscan(short)
eclipse.image_sub_local(short, bias)
eclipse.image_div_local(div, short)

stats = eclipse.iter_stat(div, MAXIMUM_ITERATIONS, REJECTION_THRESHOLD)
thresh_lo = stats.avg_pix - REJECTION_THRESHOLD*stats.stdev
thresh_hi = stats.avg_pix + REJECTION_THRESHOLD*stats.stdev

pixelmap = eclipse.image_threshold2pixelmap(div, thresh_lo, thresh_hi)

# eclipse pixelmap counts good pixels
```

```
count = pixelmap.lx * pixelmap.ly - pixelmap.count
```

QC Parameters:

```
count         : the number of non-linear pixels
```

5.3.4 Req. — *Cat III: CCD Charge Transfer Efficiency*

Objective:

Characterize horizontal and vertical transfer efficiency (CTE) per single transfer (in units of the fraction of the charge actually transferred).

Taken from WFI@2.2m: Ten flatfields are taken with 50 vertical and 50 horizontal overscan pixels and a mean exposure level of about 20000 ADU's. The mean is computed and corrected for the bias. Average signal levels are determined in the two overscan regions as well in the light sensitive pixels just preceeding the respective overscan pixels. Any signal found in the overscan pixels was due to non-unity CTE lost from the neighbouring light-sensitive pixels. The fractional charge still remaining in the light-sensitive pixels is the CTE.

Note added after analysing Bias behaviour in overscan regions on OmegaCAM CCDs (see test report Notes on OmegaCAM bias level- Nov 2003): The test report shows that there is significant reminiscence in the detector/amplifier chain with typical time scale of about a second. Therefore both the vertical and the horizontal overscan regions are affected by this reminiscent signal (only the upper 50 rows of the Y-overscan region are free from this signal). Thus the above decribed method for WFI is not expected to deliver very useful information on CTE.

Alternative: it might be more practical and informative to inspect the tails in X and Y of very bright stars.

Fulfilling or fulfilled by:

selfstanding but related to masks for **req. 541** *Bias - doit*, **req. 522** *Hot pixels* and **req. 535** *CCD Cold Pixels*.

When performed/frequency:

daytime- Commissioning, in RP once half year *alternative* additional data reduction of **req. 525** *Cross talk*

Sources, observations, instrument configurations:

Dome flat field- lamp. Use r' filter.

Alternative none- use **req. 525** *crosstalk*.

Required accuracy, constraints:

CTE $>$ 0.999995 per parallel or serial shift.

Note, that a small fraction of the **OmegaCAM**CCDs will have a CTE slightly below these values.

Estimated time needed:

30 min.

TSF:

Mode– Stare N=10

(**TSF– OCM_img_cal_domeflat,** N=10)

= **TSF– OCM_img_cal_cte**

Inputs:

CalFile– 541 *Master Bias frame*

Outputs:

CalFile– 534 *charge transfer efficiency factors*

CA:

The originally proposed method to determine CTE will give very limited info due to significant reminiscence.

Original text: Ten flat fields are taken with 50 vertical and 50 horizontal overscan pixels and a mean exposure level of about 20000 ADU's. The mean is computed and corrected for the bias. Average signal levels are determined in the two overscan regions as well as in the light sensitive pixels just preceeding the respective overscan pixels. Any signal found in the overscan pixels was due to non-unity CTE lost from the neighbouring light-sensitive pixels. The fractional charge still remaining in the light-sensitive pixels is the CTE.

An alternative method, inspecting the tails of very bright stars in X and Y direction can be done on RTD and does not need a special recipe.

5.3.5 Req. — *Cat I: CCD Cold Pixels*

Objective:

Identify cold pixels.

Cold pixels should be recorded and ignored (assigned a weight of 0) in dedithering/dejittering and source extraction. For this purpose the bad/cold pixel map, together with the hot pixel map (**req.**522) is used to assign a weight of zero to the affected pixels in the weight map (**seq.– 633**).
Cold pixel maps are constructed from reduced dome (**req.**542) or twilight (**req.**543) flats. The flatfield is smoothed. The smoothed flat is used to flatten the flat. In this flatfielded image, pixels that are outside a given range (0.96-1.04) are taken to be cold pixels. Note that this invalidates any pixel whose gain differs significantly from its immediate neighbours. In particular, this also identifies pixels that are bright relative to their neighbours as "cold". Note, that pixels above the threshold are formally not "cold", but are flagged annyway. In the end, hot plus cold pixel map are combined in weightmap.

Fulfilling or fulfilled by:

Requires data reduction of master domeflat (**req.**542) or twilight (**req.**543) flat frames.

When performed/frequency:

daytime- Commissioning, in RP once per 3 months.

Sources, observations, instrument configurations:

Use master dome- or twilight flat. Use r' filter.

Required accuracy, constraints:

Quality Check: Number of hot pixels to be determined by experience/lab values. The total number of bad pixels (hot pixels + cold pixels) is less than 80000. Difference in number of cold pixels w.r.t. reference version less than 100.

Estimated time needed:

Observation: None. Reduction: < 20 sec./CCD.

TSF:

Use master dome- (**req.**542) or twilight flat (**req.**543)

Inputs:

CalFile– 542 *Dome flat* or **CalFile– 543** *Twilight flat*

Outputs:

CalFile– 535 *Cold pixel map*

Recipe:

```
Cold_Pixels -i flat [-low THRESHOLD_LOW] [-high THRESHOLD_HIGH]

flat            : the master dome- or twilight flat
THRESHOLD_LOW   : the lower flagging threshold (float)
                  Range of allowed values: 0.90 - 1.00. Default: 0.96
```

```
THRESHOLD_HIGH : the higher flagging threshold (float)
                 Range of allowed values: 1.00 - 1.10. Default: 1.04
```

(could be incorporated in **Recipe– Dome_Flat** or **Recipe– Twilight_Flat**)

CA:

1. Make a smoothed flatfield using sextractor.
2. Divide the flatfield by the smoothed flatfield.
3. Construct a pixelmap using thresholding.
4. Count the number of bad pixels in the pixelmap.

Note that we use sextractor to produce the smoothed image. Sextractor uses a robust algorithm to estimate the background on a grid and interpolate between these grid points. By measuring this background for the flatfield we essentially have a fast smoothing algorihm with a large kernel, that is relatively insensitive to bad pixels.

CAP:

```
# make a smoothed 'background'
sextractor(flat, DETECT_THRESH=1000,
                 CHECKIMAGE_TYPE='BACKGROUND',
                 CHECKIMAGE_name=smoothed_flat)

# flatten the flat
normalized = eclipse.image_div(flat, smoothed_flat)

# threshold the bad pixels
pixelmap = eclipse.threshold2pixelmap(normalized,
                                      THRESHOLD_LOW,
                                      THRESHOLD_HIGH)

# eclipse pixelmap counts good pixels
count = pixelmap.lx * pixelmap.ly - pixelmap.count
```

QC Parameters:

```
count   : number of cold pixels
```

5.3.6 Req. — *Cat III: CCD Hysteresis, strong signal*

Objective:

Quantify the effect of CCD signal reminiscence at timescales *larger* than 60 sec.
Reminiscence of a strong signal (a saturated star) in subsequent observations ("ghosts") is a potentially debilitating problem for data reduction and interpretation. The absence of this effect should be verified by observations of very bright objects and subsequent dark exposures.
If "sources" are detected in the dark frames at the pixel positions of bright sources in the first field, signal reminiscence is a problem, which can be characterized by the decay time of a strong signal.

Fulfilling or fulfilled by:

Selfstanding

When performed/frequency:

Commissioning.

Sources, observations, instrument configurations:

Observe a field containing very bright objects followed by a few dark exposures. Number of dark exposures to be determined by experience.

Estimated time needed:

30 minutes

TSF:

Mode– stare N=2
TSF– OCM_img_obs_stareand **TSF– OCM_img_cal_dark**

Outputs:

When significant effect is detected:
CalFile– 536 *CCD Hysteresis*, containing the signal decay time.

CA:

First, quick look analysis on RTD.
Subsequent: process the bright field with standard image pipeline.
Then continue with interactive procedure. Select the bright (saturated) objects from the catalog of the bright field. Use the pixel position of these sources to inspect the corresponding positions in the dark frame. In case reminiscent signal is detected: plot the source magnitude as function of time (exposure number), and measure the decay time. If reminiscence is detected, repeat the experiment with more dark frames and determine decay-times.

5.4 Detectors operational specific calibrations

In this Section **req.**'s are listed which are essential related to the daily operations of the acquisition of the science data.

For all detector and photometry related calibrations each CCD is characterized independently of the others. (Only in the case of astrometric solutions, data of various chips is combined).

A **set of calibration lamps** together with a **dome screen** is used to monitor the health of the instrument and to measure the fine structure of the flatfield.

The calibration lamp system contains two **sets** of 4 commercial 12-24V halogen lamps each. Each set is operated independently. Each set is stabilized in current supply (one unit for whole set). Lamps are switched on/off with a gradual increase/decrease of the current over a timespan of 3-5 minutes. Implementation TBC. When operated this way, the nominal lamp instabilities are expected to be of the order of 0.05%/hour for a timespan of 200 hours of lamp operations (private comm. Philips Labs). For a nominal 1-2 hour/day of operations the lamp instability is expected to be of the order 1.5-3 % per month. The lamp instabilities are expected to be strictly linear after 100 hours of operations. When operating two sets at different rates, say one set 1 hour/day the other set at say 1 hour/two weeks a full characterization of the lamp stability can be achieved at accuracies better than a few percent on a monthly basis. Also continuity after failure of one lamp can be obtained. The accuracy is better than required as also other factors, like dust on the lamps and background light will affect the effective illumination of the screen. Altogether, the system is expected to provide control over the illumination of the screen with an accuracy better than 5-10%, which will be used for a daily health check on the overal throughput/health of the detectors (**req. 547** *Quick detector responsivity and health check*). This activity provides a deliberate redundancy with flatfield measurements on the dome screen (**req. 542** *Flat field – dome key bands–doit*), in order to provide the necessary cross-check in the off-line calibration analysis procedures. Next to the health checks taken during the night, a standard health check using a photometric standard field (also providing the absolute photometry is specified in **req. 562** *Photometric Calibration – Monitoring*). The system of lamps, flatfields measurements, quick checks using the lamps and health checks on the sky on photometric standard fields is designed to support the photometric calibration of a Survey System, for many years to come. The system provides redundancy faciltating cross-checks and has a typical maintenance/update frequency once/month.

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 $S(x, y)$ to a photometric scale

3 Relating the x, y coordinates to an astrometric reference system.

The raw science and standard images record $S(x, y)$ counts in pixel x, y, that are related to the incident photon flux $I(x, y, \lambda)$ by:

$S(x, y) = b(x, y) + G \int g(x, y, \lambda) I(x, y, \lambda) d\lambda,$

with G the ADU conversion factor, $g(x, y, \lambda)$ the quantum efficiency or **gain** as function of position and wavelength, and $b(x, y)$ the bias offset.

The photometric and astrometric calibration (steps 2 and 3) are the subject of Sections 5.5 and 5.6, respectively. Here we list the calibration data necessary to remove the effects of bias and gain variation over the image. These calibration data include:

- **Bias** to subtract residual pattern in the bias offset.
- **Flatfields** to correct for non-uniform gain.
- **Fringe maps** to remove the fringe-patterns
- **Weight maps** to determine the relative contribution of each pixel when image data are combined,

A first-order approximation of the bias level in an image is provided by the median of the overscan region. A more accurate determination of the bias offset takes into account the following two effects: 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**, is applied by averaging the overscan pixels for each line, and subtracting this value from that line. Also, experience with the WFI has shown the presence of residual patterns in the bias offset over the image area. Under the assumption that these patterns are also present in **OmegaCAM** data its characterization by means of a master bias frame is specified in **req.**541.

Note that the **OmegaCAM** calibration and image pipelines support several overscan correction methods (not doucumented separately in every requirement). It is imperative that all calibration data is derived using the same overscan correction method.

The gain, $g(x, y, \lambda)$, incorporates the wavelength-dependent pixel-to-pixel variation in transmissivity of the different lightpaths through the telescope optics, filters and detectors. The gain can be approximated with

$g(x, y, \lambda) = g_{DQE}(\lambda) g_{ff}(x, y)$,

with $g_{ff}(x, y)$ the pixel-to-pixel variation in the gain, and $g_{DQE}(\lambda)$ the overall detector quantum efficiency (zeropoint), at $g_{ff} = 1$, which is subject to photometric calibration (Section 5.5).

The characterization of the pixel-to-pixel variation of the gain, the **flatfield**, is obtained by observing a spatially uniform source of illumination. A normalized version of such an image provides a measure of the relative variation of the gain over the image area, $g_{ff}(x, y)$. Note that this flatfield measures a combination in the pixel-to-pixel gain variation and the variation in transmissivity of the different light paths through the telescope optics and filters.

The ideal flatfield observation is:
i) uniformly illuminated
ii) bright, to minimize errors due to photon noise,
iii) of constant color, preferably a color that is the same as the night sky.

Several methods to determine a flatfield will be operational. However, each method suffers from different drawbacks. The various characterizations of the flat field, the **dome**, **twilight**, and

night sky flats, are specified under the **req.**542-545, while the eventual **master flatfield** to be applied to science and standard field observations, is constructed from a suitable combination of the dome, twilight and night sky flat fields (**req.**546).

Dome flats (**req.**542) are obtained by observing in **telescope screen park position** a fixed **domescreen** *relatively* uniformly illuminated by the **calibration lamp** with a stabilizing power supply. The illumination is not sufficiently uniform to measure large scale variations, and the color is very different from the night sky.

Twilight flats (**req.**543) are based on a bright, uniform source of illumination (twilight sky), that is unfortunately not of constant brightness and color. Unfortunately the 'twilight gradient' precludes measurement of the largest scale gain variation. Also bright objects may be visible even at twilight, which provides an additional complication.

Night sky flats (**req.**544), obtained by combining a large number of science observations, most closely mimic the illumination properties of the science frames themselves. These are the only flatfields usable for measuring the largest scale gain variations. Unfortunately, the assumption that the illumination is uniform (except for astronomical sources of course), has proven to be invalid on WFI (e.g Manfroid et al, 2000). It remains to be seen how large this problem of "sky concentration" will be for **OmegaCAM**. Computing the night sky flat is also a computationally expensive process, both because of the large numbers of frames involved, and because the need for a proper masking of bright objects in the field. Fortunately, this measurement may also be usable for the fringe correction.

The master flatfield (**req.**546), to be applied to science and standard field observations, is constructed from a suitable combination of the dome, twilight and night sky flat.

The approximation that the pixel-to-pixel variation in the gain is independent of wavelength is in fact incorrect. Interference effects, mainly in the filters and thinned silicon layers of the CCDs, introduce wavelength dependent gain variations that vary on small angular scales. Since most sources are continuum sources, and only the convolution $\int g(\lambda)I(\lambda)d\lambda$ is measured, this effect can be ignored when measuring source fluxes. However, due to variable strength of several sky lines, mostly apparent at the long wavelengths, the background will exhibit so-called **fringing** patterns, which can change during the night. This requires an additional calibration step for bands redward of R: the construction of suitable fringed background images (**req.**545).

The weight map is an important auxiliary file, which is used in several image processing steps. The weight map is intimately linked to the flatfields and therefore its construction is also addressed in this section.

Whenever individual pixels are combined, either in constructing source lists necessary for photometric and astrometric calibration, or in the coaddition of different frames, the **OmegaCAM**-reduction pipeline uses variance weighting (weight= $1/\sigma^2$). The inverse variances are recorded in **weight maps**, which are computed as part of the image pipeline (ref. **seq.– 633**).

The debiased images record $S(x, y)$ counts in pixel x, y, that are related to the photon flux $I(x, y)$ by:

$S(x,y) = G g_{DQE} g_{ff}(x,y) I(x,y)$,
Since photon shot-noise is much larger than the read-out noise, the rms-noise in the raw data is given by:
$\sigma_S(x,y) = G(g_{DQE} g_{ff}(x,y) I(x,y))^{1/2} = (GS(x,y))^{1/2}$.
Once data has been flatfielded ($S' = S/g_{ff}$), the counts are given by:
$S'(x,y) = G g_{DQE} I(x,y)$,
and the rms-noise by:
$\sigma_{S'}(x,y) = G(g_{DQE} g_{ff}(x,y) I(x,y))^{1/2})/g_{ff} = (GS'(x,y)/g_{ff})^{1/2}$
The photon flux $I(x,y)$ is a sum of a uniform background I_{back} plus sources $I_{src}(x,y)$. Since, the surface brightness of the sky is (much) larger than the surface brightness of most sources, the rms-noise is given by:
$\sigma_{S'}(x,y) = G[g_{DQE} I_{back}/g_{ff}(x,y)]^{1/2} = (GS'_{back}/g_{ff})^{1/2}$
Hence, the rms-noise in an image is the product of a factor ($(GS'_{back})^{1/2}$) that is constant over one image, but will vary between images, and a factor ($g_{ff}^{-1/2}$), the inverse of the square-root of the flatfield, which varies over the image.

To account for image defects the **weight map** will give bad pixels a weight of zero. Bad pixels include hot (**req.**522) and cold (**req.**535) pixels, as well as pixels affected by saturation, cosmic rays and satellite tracks, as determined in **seq.– 633**.

5.4.1 Req. — *Cat I: Bias - doit*

Objective:

Determine master bias frame.

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**, is applied by averaging the overscan pixels for each line, and subtracting this value from that line. It is noted that the bias level may jump significantly after a cold restart of the FIERA.

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 ($\sigma =$ dispersion of the 10 bias exposures of individual pixels), due to particle hits during read-out. The resulting master bias frames will be used for the correction of all frames. For each master bias frame the mean value for each CCD chip will be determined and evaluated in a trend analysis.

As the readout noise dominates the rms scatter in the bias frames, while the shotnoise 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/$\sqrt{10}$).

A comparison with a previous master bias frame will be done as an evaluation of the overall health of the instrument and the quality of the data. This will thus measure short-term variations. Long term variations can be assessed using trend analysis.

A comparison of the mean level with laboratory values will be used as an overall quality check.

Fulfilling or fulfilled by:

Selfstanding. Raw data are also used for **req. 522** *Bad/hot pixels*

Calfiles used by: **req. 542** *Dome flat*, **req. 543** *Twilight flat* **req. 544** *Night sky flat* **seq. 632** *Trim, de-bias, flatfield*

When performed/frequency:

daytime- Commissioning, in RP initially daily. Later the frequency is to be determined by experience.

Sources, observations, instrument configurations:

10 observations with 0 (zero) seconds exposure time.

Required accuracy, constraints:

The required accuracy per pixel in the master bias frame is "nominal read-outnoise/$\sqrt{10}$".

For the quality check: Since an overscan correction is performed, the deviation of the mean level of master bias (bias level) from zero, should be less than TBD.

Estimated time needed:

Observation: 15 min. Reduction: $<$ 2 min./CCD.

TSF:
Mode– Stare N=10
TSF– OCM_img_cal_bias

Inputs:
Raw data bias frames
CalFile– 541 *Master Bias frame* reference frame
Laboratory values of bias levels.

Outputs:
CalFile– 541 *Master Bias frame* to be used by **seq. 632** *de-bias flat field*

Recipe:

```
Bias -i <raw_bias_list> -r READNOISE [-oc OVERSCAN_CORRECTION]

raw_bias_list       : list of raw bias exposures
READNOISE           : readnoise value in ADU (float).
                      Range of allowed values: 1.0 - 5.0.
OVERSCAN_CORRECTION : overscan correction mode (integer).
                      Description of allowed values:
                        0: apply no overscan correction (default)
                        1: use median of the prescan in the
                           x-direction
                        2: use median of the overscan in the
                           x-direction
                        3: use median of the prescan in the
                           y-direction
                        4: use median of the overscan in the
                           y-direction
                        5: use the per-row value of the prescan in
                           the x-direction
                        6: use the per-row value of the overscan in
                           the x-direction
```

Before applying this recipe, use **Recipe– Split**—which is documented in **seq.– 631**—with the `-t bias` option to split the raw multi-extension FITS input files.

CA:
Processing (make):
1. Trim the input raw bias frames, and do an overscan correction on the resulting images.
2. Combine the trimmed images into an average bias frame, using 5σ rejection.
3. Measure the statisics of the combined image.

CAP:
`Constants:`

```
  SIGMA_CLIP          : clipping threshold for bad pixels in raw
                        input data (float). Value: 3.0.
  REJECTION_THRESHOLD : clipping threshold for bad pixels in the
                        resultant bias image (float). Value: 4.0.
  MAXIMUM_ITERATIONS  : number of iterations in bias statistics
                        measurement (integer). Value: 3.

reduced_bias_list = []
for raw_bias in raw_bias_list:
    reduced_bias = eclipse.trim_and_overscan(raw_bias)
    reduced_bias_list.append(reduced_bias)

# first estimate of mean
median = eclipse.cube_median(reduced_bias_list)

# Find outliers in first reduced bias image
dev = eclipse.image_sub(median, reduced_bias_list[0])
good = eclipse.threshold2pixelmap(dev, -SIGMA_CLIP * READNOISE,
                                  SIGMA_CLIP * READNOISE)
good = eclipse.pixelmap_2_image(good)

# record these values
sum_data   = eclipse.image_mul(reduced_bias_list[0], good)
sum_pixels = good

# Find outliers in remaining images, and reject these too
for reduced_bias in reduced_bias_list[1:]:
    dev = eclipse.image_sub(median, reduced_bias)
    good = eclipse.threshold2pixelmap(dev, -SIGMA_CLIP * READNOISE,
                                      SIGMA_CLIP * READNOISE)
    good = eclipse.pixelmap_2_image(good)
    eclipse.image_add_local(sum_data,
                            eclipse.image_mul(reduced_bias, good))
    eclipse.image_add_local(sum_pixels, good)

# new mean
bias_image = eclipse.image_div(sum_data, sum_pixels)
bias_statistics = eclipse.iter_stat(bias_image, MAXIMUM_ITERATIONS,
                                    REJECTION_THRESHOLD)
```

QC Parameters:

```
mean   : mean pixel value of the bias image
median : median pixel value of the bias image
stdev  : standard deviation of the bias image pixel values
```

5.4.2 Req. — *Cat I: Flat-field - dome key bands + user bands - doit*

Objective:

Determine master dome flat frame for both **Keybands** and **Userbands**.

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, remove cosmic rays.

During the lifetime of **OmegaCAM** the dome flatfields shall be measured for the 4 keybands and the key-composite filter at least once/week. Thus at least within 3 days of the taking of science data a dome flatfield in the key passband will be available.

The sequence of master dome flatfields in the keybands, acquired over periods of months to years will be used to perform a trend analysis on the long term stability of the instrument and lamp which illuminates the dome flatfield. With the exception of the effects of the unstability of the lamp, this trend analysis is redundant with that obtained from both **req. 563** *Photometric Calibration - zeropoint key bands - doit* and that of **req. 562** *Photometric Calibration - monitoring*. Thus, by combining the results of these **req.**'s an accurate description of the behaviour of the lamp is feasible. The prediction of the behaviour of the lamp is a result, which will be used as an input for **req. 547** *Quick detector responsivity check*.

The dome flatfields will be used on an individual CCD chip level. The relative variations of the quantum efficiciency between individual CCD chips will be measured by **req. 563** *Photometric Calibration - zeropoint key bands - doit*.

The redundancy between various measurements of **req. 563** *Photometric Calibration - zeropoint key bands - doit* and **req. 542** *Master Domeflat* at keybands will be used to evaluate the relative chip-to-chip gain variations, and in due time, when advanced insight in this item is achieved, this knowledge might be used to further optimize observing scenarios.

The cross-calibration/handshaking of two lamp sets will be done during a transition period of about one week. During that week observations will be taken with both lamp sets. Template TSF–OCM_img_cal_domeflat has a parameter which one of the two lamp sets will be used It is anticipated that about every 3 months a new set will become operational. During this transition period of about a week both lamps sets will be observed sequentially. Also TSF-OCM_img_cal_quick can be used for the crosscalibration.

For the userbands the dome flatfield will only be taken in the period of the week that that particular passband has been used for science observations. No trend analysis will be done on the data taken in the User pass bands.

It will be checked whether the dome flatfields are reproducable and whether artifacts appear on the same pixels. This will also serve as a check on the reproducibility of the filter exchange mechanism.

Fulfilling or fulfilled by:

Selfstanding.

Output is used by:
req. 563 *Photometric Calibration - zeropoint key bands - doit*
req. 547 *Quick detector responsivity* through the characterization of the dome flat field lamp.
req. 546 *Master flatfield*

When performed/frequency:

Daytime, daily. For the keybands the dome flatfields will be measured at least once/week. For the Userbands the dome flatfields will be measured at least within 3 days that the Userband has been used for science observations. When filters have been changed in the cassette the presence of dust and/or scratches might require new flat field exposures. During Comm. the reproducibility of the filter exchange mechanism will be checked.

Sources, observations, instrument configurations:

Domeflats with the four keyfilters and the single key composite filter. 5 observations per filter, with approximately 20000 ADU. Adding 5 exposures of 20,000 counts satisfies the accuracy requirement. Each science/standard observation preferably has an associated domeflat observed within 3 days. About 4 dome flats will be measured per day. The calibration lamps will be used and during a transition period of about one week both sets will be used.

Required accuracy, constraints:

Accurately measuring pixel-to-pixel gain variations as small as 1%. Re-insertion of the filter shall not alter the flat field stucture by more than 0.3% (rms, measured over the full detector area).

Estimated time needed:

Observation: 10 min. Reduction: 3 min./CCD.

TSF:

Mode– Stare N=5
TSF– OCM_img_cal_domeflat

Inputs:

Raw dome flatfields
CalFile– 541 *Master Bias frame*
CalFile– 522 *Hot pixel map*
CalFile– 523 *Gain*
CalFile– 535 *Cold pixel map*

Outputs:

CalFile– 542 *Master Domeflat frame*
CalFile– 542L *Dome Lamp*

Recipe:

```
Dome_Flat -i <raw_domeflats> -b bias -g GAIN [-c cold] [-h hot]
          [-oc OVERSCAN_CORRECTION]
```

```
raw_domeflats       : raw dome flatfields

bias                : master bias frame

cold                : the cold pixel map

hot                 : the hot pixel map

GAIN                : the gain value in e/ADU (float).

                      Range of allowed values: 0.1 - 10

OVERSCAN_CORRECTION : overscan correction mode (integer).

                      Description of allowed values:

                        0: apply no overscan correction (default)

                        1: use median of the prescan in the

                           x-direction

                        2: use median of the overscan in the

                           x-direction

                        3: use median of the prescan in the

                           y-direction

                        4: use median of the overscan in the

                           y-direction

                        5: use the per-row value of the prescan in

                           the x-direction

                        6: use the per-row value of the overscan in

                           the x-direction
```

Before applying this recipe, use **Recipe– Split**—which is documented in **seq.– 631**—with the `-t dome` option to split the raw multi-extension FITS input files.

CA:

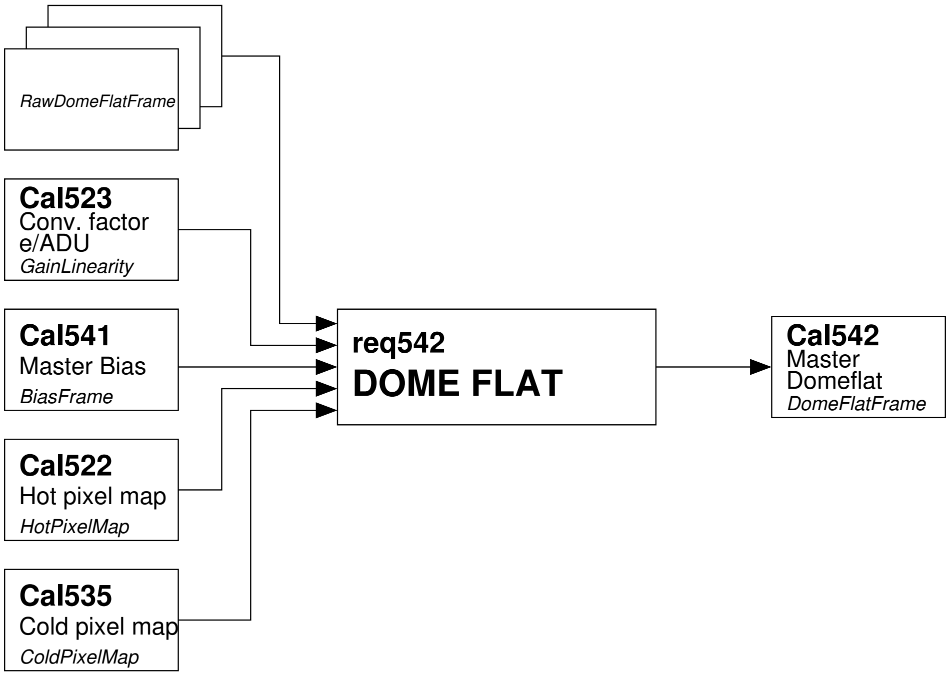


Fig 5.4.2 Dataflow and object class names (in small italic font) for req542

Process (make):

1. Construct a mask from input hot and cold pixel maps.
2. Trim, overscan correct and de-bias the raw dome frames.
3. Normalize frames to one illumination level (median pixel value 1.0).
4. Build a mean frame, rejecting 5σ outliers.
4.1 Compute the median frame of the stack of input frames.
4.2 Compute the error in each input frame ($\sigma = \sqrt{gain * counts}$)
4.3 Reject 5σ deviations in each input frame
4.3 Compute the mean in the remaining data

CAP:

```
Constants:
  SIGMA_CLIP      : the sigma clipping threshold

mask = eclipse.pixelmap_binary_AND(hot, cold)

scales = []
```

```
scaled_images = []

# Trim, overscan correct, de-bias and scale each input image
for raw_dome in raw_domeflats:
    reduced_dome = eclipse.trim_and_overscan(raw_dome)
    eclipse.image_sub_local(reduced_dome, bias)
    stat = eclipse.image_stat_opts(reduced_dome, pixelmap=mask)
    scaled_images.append(eclipse.image_mul(reduced_dome, 1.0/stat.median))
    scales.append(1.0/stat.median)

median_image = eclipse.cube_median(scaled_images)

# Record the good values
good_data  = eclipse.make_image(xsize, ysize)
good_count = eclipse.make_image(xsize, ysize)

# Loop over remaining data
i=0
for dome in scaled_images:
    dev_image = eclipse.image_sub(median_image, dome)
    err_image = eclipse.image_pow(dome, 0.5))
    eclipse.image_div_local(dev_image, err_image)

    threshold = SIGMA_CLIP * sqrt(GAIN * scales[i])
    good = eclipse.threshold2pixelmap(dev_image, -threshold, threshold)
    good = eclipse.pixelmap_2_image(good)

    eclipse.image_add_local(good_data,
                            eclipse.image_mul(dome, good))
    eclipse.image_add_local(good_count, good)
    i=i+1

#
domeflat = eclipse.image_div(good_data, good_count)
```

QC Parameters:

```
mean   : mean pixel value of the master dome flat
median : median pixel value of the master dome flat
stdev  : standard deviation of the master dome flat pixel values
```

5.4.3 Req. — *Cat I: Flat-field - twilight*

Objective:

Determine master twilightflat frame, using observations of the twilight sky.

Twilightflat observations will be attempted for each passband that is observed during that night. If insufficient twilight time is available then the twilightflat observations are taken preferably in the previous or subsequent night. In addition twilightflats of the 4 keybands will be taken at least once a week.

In order to minimize the spatial gradient in the sky brightness, the observations need to be made on the solar circle, i.e. the great circle through the zenith and the sun's position, at a zenith distance of about 20° in the solar antidirection. Preferably, the field of view does not include stars brighter than TBD magnitude.

The consortium will deliver a list a targets for twilight flats.

Master twilight flats are obtained through an average with sigma rejection procedure on a stack of raw twilight flats, intended to remove any contamination (including stars) present on individual raw twilight flats and reduce photon shot noise.

Fulfilling or fulfilled by:

Selfstanding

When performed/frequency:

Evening twilight.

An attempt will be made to observe twilightflats for all bands observed during the night. For all observed bands, twilightflats will be observed within a maximum of 2 nights. In addition, twilightflats for the keybands will be obtained at least once a week, irrespective of whether keybands were used for science observations during that week.

Sources, observations, instrument configurations:

Twilightflat with 5 observations per filter at 'empty' sky, near 20° from zenith in solar antidirection. It may be advisable to determine a standard target list of approximately 10-20 empty fields, equally spaced in right ascension, at 20° from zenith. A 'standard' target list will be provided by consortium at commissioning.

The telescope should be tracking. The maximum jitter offset allowed by the TSF should be used in order to be able to remove stars. Observations should approximate 20000 ADU. Exposure time should be based on skybrightness, typically it may vary from 0.1 sec to 300 sec, changing rapidly during the twilight period. No effects of shutter are expected.

The Tyson and Gal technique/formula for adapting exposure time to the variations of the twilight brightness is implemented in the observing template (similar to WFI@2.2m). A trend analysis of twilightflat can be used to calibrate this formula.

Each science/standard observation preferably has an associated twilightflat observed within 2 days. We expect a total of 10 twilightflats to be observed during each twilight period.

Required accuracy, constraints:

Mean levels should be approximately 20000 ADU.

Estimated time needed:
Observation: in total 25 min./night. Reduction: 3 min./CCD.

TSF:
Mode– Jitter N=5
TSF– OCM_img_cal_skyflat
Use an appropriate (empty) field, exposure time determined by Tyson-Gal formula.

Inputs:
Raw twilight flatfields
CalFile– 541 *Master Bias frame*
CalFile– 522 *Hot pixel map*
CalFile– 523 *Gain*
CalFile– 535 *Cold pixel map*

Outputs:
CalFile– 543 *Master Twilightflat frame*

Recipe:
```
Twilight_Flat -i <raw_twilightflats> -b bias -g GAIN [-c cold] [-h hot]
              [-oc OVERSCAN_CORRECTION]

raw_twilightflats   : raw twilight flatfields
bias                : master bias frame
cold                : cold pixel map
hot                 : hot pixel map
GAIN                : the gain value in e/ADU (float)
                      Range of allowed values: 0.1 - 10.0
OVERSCAN_CORRECTION : overscan correction mode (integer).
                      Description of allowed values:
                        0: apply no overscan correction (default)
                        1: use median of the prescan in the
                           x-direction
                        2: use median of the overscan in the
                           x-direction
                        3: use median of the prescan in the
                           y-direction
                        4: use median of the overscan in the
                           y-direction
                        5: use the per-row value of the prescan in
                           the x-direction
                        6: use the per-row value of the overscan in
                           the x-direction
```

Before applying this recipe, use **Recipe– Split**—which is documented in **seq.– 631**—with the `-t twilight` option to split the raw multi-extension FITS input files.

CA:

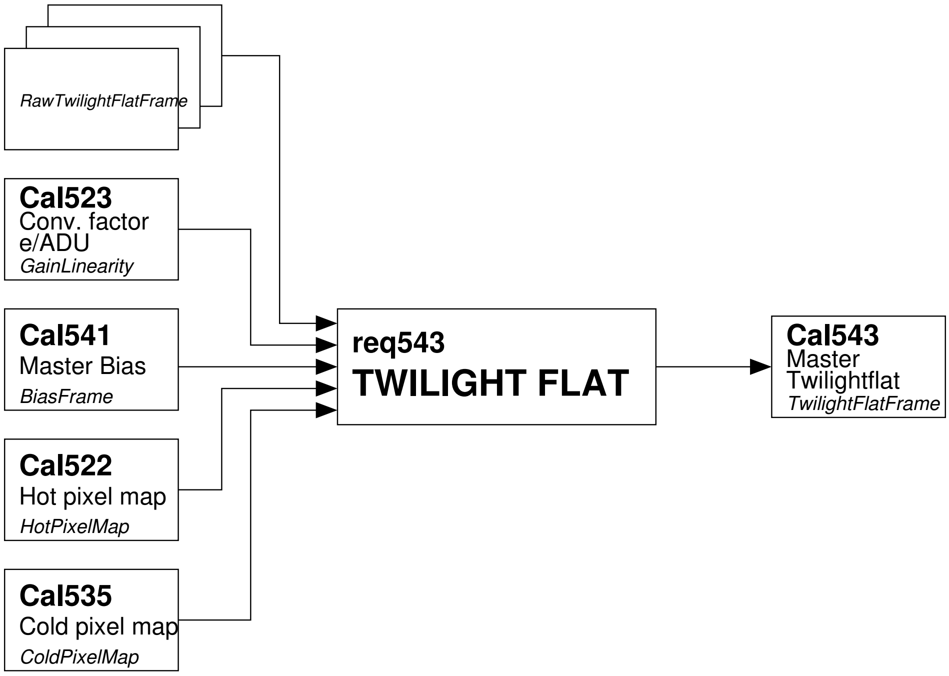


Fig 5.4.3 Dataflow and object class names (in small italic font) for req543

NOTE: The data reduction procedure is essentially the same as for dome flats (**req.**542).

Process (make):

1. Construct a mask from input hot and cold pixel maps.
2. Trim, overscan correct and de-bias the raw twilight frames.
3. Normalize frames to one illumination level (median pixel value 1.0).
4. Build a mean frame from the stack of input frames, rejecting 5σ outliers.

4.1 Compute the median frame of the stack of input frames.
4.2 Compute the error in each input frame ($\sigma = \sqrt{gain * counts}$).
4.3 Reject 5σ deviations in each input frame.
4.3 Compute the mean in the remaining data.

CAP:

`Constants:`

```
  SIGMA_CLIP    : the sigma clipping threshold

mask = eclipse.pixelmap_binary_AND(hot, cold)

scales = []
scaled_images = []

# Trim, overscan correct, de-bias and scale each input image
for raw_twilight in raw_twilightflats:
    reduced_twilight = eclipse.trim_and_overscan(raw_twilight)
    eclipse.image_sub_local(reduced_twilight, bias)
    stat = eclipse.image_stat_opts(reduced_twilight, pixelmap=mask)
    scaled_images.append(eclipse.image_mul(reduced_twilight, 1.0/stat.median))
    scales.append(1.0/stat.median)

median_image = eclipse.cube_median(scaled_images)

# Record the good values
good_data  = eclipse.make_image(xsize, ysize)
good_count = eclipse.make_image(xsize, ysize)

# Loop over remaining data
i=0
for twilight in scaled_images:
    dev_image = eclipse.image_sub(median_image, twilight)
    err_image = eclipse.image_pow(twilight, 0.5))
    eclipse.image_div_local(dev_image, err_image)

    threshold = SIGMA_CLIP * sqrt(GAIN * scales[i])
    good = eclipse.threshold2pixelmap(dev_image, -threshold, threshold)
    good = eclipse.pixelmap_2_image(good)

    eclipse.image_add_local(good_data,
                            eclipse.image_mul(twilight, good))
    eclipse.image_add_local(good_count, good)
    i=i+1

#
twilightflat = eclipse.image_div(good_data, good_count)
```

QC Parameters:

```
mean   : mean pixel value of the master twilight flat
median : median pixel value of the master twilight flat
stdev  : standard deviation of the master twilight flat pixel values
```

5.4.4 Req. — *Cat I: Flat-field - night sky*

Objective:

Create Night Sky flat frame.

The flatfield that most closely reproduces the actual gain variation of the science and standard observation, can be obtained by averaging a large number of *flat-fielded* science and standard observations, taking care of properly masking the contaminating objects. While such a night-sky flat (aka superflat) could, in principle, improve on the quality of the twilight flat, the procedure to obtain this flat can be computationally very expensive. On the other hand night-sky flats may also be suitable for fringe removal.

It remains to be seen to what extent the problem of sky concentration, i.e. nonuniform illumination due to stray light/reflection affects the quality of the night-sky flat.

Because these night-sky flats can only be obtained from actual observations, we cannot guarantee their availability for bands for which only standards were obtained. It is, therefore, not clear how routine building of the skyframes should be incorporated into our photometric calibration scheme.

A minimum of 5 images in a night in a given band is required to optimally fulfill this requirement. Images are stacked and a median average is calculated, intended to remove any non-systematic effects (objects, cosmic rays, satellite tracks, etc.).

Check reproducibility of artifacts on flat field images in order to inspect filter exchange mechanism reproducibility and flexure of telescope.

Fulfilling or fulfilled by:

Data reduction of raw data from science and photometric standard observations
The night sky flat is used in **req.**546 Master flat.

When performed/frequency:

daytime, daily (flexure, reproducibility during Comm)

Sources, observations, instrument configurations:

Science and standard observations

Required accuracy, constraints:

This procedure would benefit from a prior detection and masking of bright objects. If no masks of bright objects are available, then a minimum of 15 frames should be included.

Estimated time needed:

Observation: None. Reduction: 5 min./CCD/filter for 15 science frames.

TSF:

Raw data of science and standard star observations in jitter and dither mode is used. Avoid using multiple exposures with the same pointing in Stare mode.

This procedure benefits from first flatfielding the data, and determining the night-sky flat as a modification of this flatfield. First de-bias the science/standard frames and apply the flats, then normalize the frames and build a median average frame.

Inputs:

Raw science images
CalFile– 541 *Master Bias frame*
CalFile– 546 *Master flatfield*
CalFile– 522 *Hot pixel map*
CalFile– 535 *Cold pixel map*

Outputs:

CalFile– 544 *Nightsky flat frame*

Recipe:

```
Nightsky_Flat -i <raw_science_frames> -b bias -f flat [-c cold] [-h hot]
              [-oc OVERSCAN_CORRECTION]

raw_science_frames  : the raw science images
bias                : the master bias image
flat                : the master flat image
cold                : the cold pixel map
hot                 : the hot pixel map
OVERSCAN_CORRECTION : overscan correction mode (integer).
                      Description of allowed values:
                        0: apply no overscan correction (default)
                        1: use median of the prescan in the
                           x-direction
                        2: use median of the overscan in the
                           x-direction
                        3: use median of the prescan in the
                           y-direction
                        4: use median of the overscan in the
                           y-direction
                        5: use the per-row value of the prescan in
                           the x-direction
                        6: use the per-row value of the overscan in
                           the x-direction
```

Before applying this recipe, use **Recipe– Split**—which is documented in **seq.– 631**—with the `-t science` option to split the raw multi-extension FITS input files.

CA:

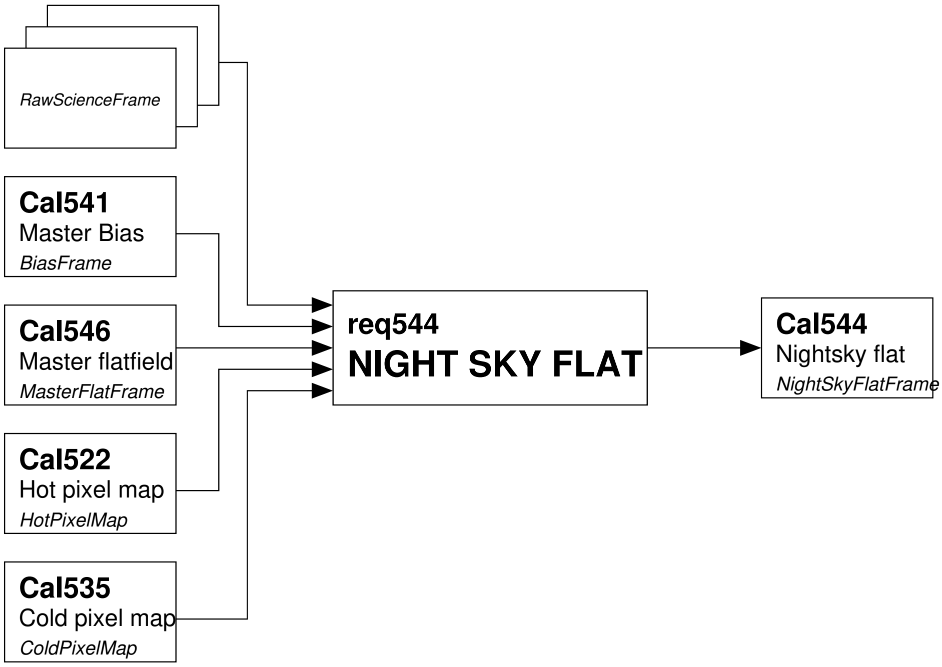


Fig 5.4.4 Dataflow and object class names (in small italic font) for req544

Process (make):

1. Check that at least three input science images are given; this is necessary to calculate a median average of a cube of these.
2. Construct a mask from input hot and cold pixel maps.
3. Trim, overscan correct, de-bias and flat-field the input science images.
4. Iteratively estimate the statistics of the resulting science images.
5. Normalize each image by dividing by its median pixel value as determined above.
6. Stack these science images in a cube.
7. Determine the median average of the cube (this is an image).
8. Normalize the night sky flat obtained in 7. (use mask obtained in 2.).
9. Assign bad pixels a value of 0 (use mask obtained in 2.).

CAP:

```
mask = eclipse.pixelmap_binary_AND(hot, cold)

cube = []
for frame in raw_science_frames:
```

```
    frame = eclipse.trim_and_overscan(frame)
    frame = eclipse.image_sub(frame, bias)
    frame = eclipse.image_div(frame, flat)
    stats = eclipse.iter_stat(frame)
    frame = eclipse.image_div(frame, stats.median)
    cube.append(frame)

nightsky = eclipse.cube_avg_median(nightsky)
stats = eclipse.stat(nightsky, pixmap=mask)
nightsky = eclipse.image_div(nightsky, stats.avg_pix)
nightsky = eclipse.image_mul(nightsky, mask)
```

QC Parameters:

```
mean   : mean pixel value of the night sky flat
median : median pixel value of the night sky flat
stdev  : standard deviation of the night sky flat pixel values
```

5.4.5 Req. — *Cat I: Flat-field - fringing*

Objective:

Determine the fringe pattern of the background.

Fringing due to variable strength of several skylines, mostly apparent at the long wavelengths, requires a different approach to background subtraction. Normally, after flatfielding, the background can be expected to be flat over the entire image, and a median of the image, excluding 5σ outliers, would in principle be sufficient to subtract the background.

In images that suffer from fringing we have to deal with a background that is variable on small ($<< 1'$) scales within the image, and can not be distinguished from sources. The image itself can, therefore, not be used to determine the background. However, given the fact that most observations are taken in jitter or dither mode, the information of several images can be combined to determine a background. This average should include enough observations to properly exclude contamination from sources, and, because the standard jitter/dither patterns only include 5 pointings, one background computation per jitter/dither is probably not sufficiently accurate. On the other hand, because the fringing pattern may vary with time and telescope position, a straight mean (the supersky) over an entire nights worth of data may also not be usable.

A suitable strategy to construct a fringed background image, usable for subtraction, thereby removing the fringe pattern, remains to be determined. If the fringe pattern is stable over the night, a decomposition of the night-sky flat in an additive and multiplicative term is feasable. The assumption that the high-frequency spatial component in the night-sky flat are fringes, while the lowest frequency components represent gain variations has been used with reasonable success.

Fulfilling or fulfilled by:

Data reduction of science and photometric standard observations

When performed/frequency:

Commissioning and when long wave science frames are taken.

Sources, observations, instrument configurations:

Use science and standard data to determine background

Estimated time needed:

Observation: None. Reduction: 5 min./CCD/filter for 15 science frames.

TSF:

Use same data as for night sky flat (**req.**544)

Inputs:

Raw science images
CalFile– 541 *Master Bias frame*
CalFile– 546 *Master flatfield*
CalFile– 522 *Hot pixel map*
CalFile– 535 *Cold pixel map*

Outputs:

CalFile– 545 *ff-fringe*

Recipe:

```
Fringe_Flat -i <raw_science_frames> -b bias -f flat [-c cold] [-h hot]
            [-oc OVERSCAN_CORRECTION]

raw_science_frames  : the raw science images
bias                : the master bias image
flat                : the master flat image
cold                : the cold pixel map
hot                 : the hot pixel map
OVERSCAN_CORRECTION : overscan correction mode (integer).
                      Description of allowed values:
                        0: apply no overscan correction (default)
                        1: use median of the prescan in the
                           x-direction
                        2: use median of the overscan in the
                           x-direction
                        3: use median of the prescan in the
                           y-direction
                        4: use median of the overscan in the
                           y-direction
                        5: use the per-row value of the prescan in
                           the x-direction
                        6: use the per-row value of the overscan in
                           the x-direction
```

Before applying this recipe, use **Recipe– Split**—which is documented in **seq.– 631**—with the `-t science` option to split the raw multi-extension FITS input files.

CA:

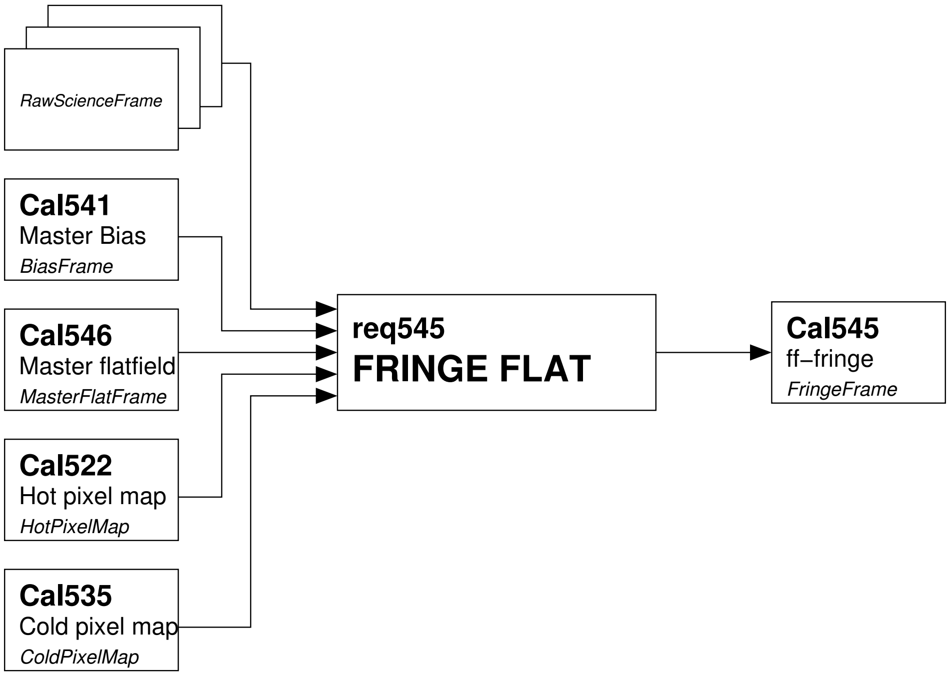


Fig 5.4.5 Dataflow and object class names (in small italic font) for req545

Process (make):

1. Check that at least three input science images are given; this is necessary to calculate a median average of a cube of these.
2. Construct a mask from input hot and cold pixel maps.
3. Trim, overscan correct, de-bias and flat-field the input science images.
4. Iteratively estimate the statistics of the resulting science images.
5. Normalize each image by dividing by its median pixel value as determined above.
6. Stack these science images in a cube.
7. Determine the median average of the cube (this is an image).
8. Normalize the fringe map obtained in 7. (use mask obtained in 2.).
9. Subtract 1.0 from the normalized fringe map.
10. Assign bad pixels a value of 0 (use mask obtained in 2.).

CAP:

```
mask = eclipse.pixelmap_binary_AND(hot, cold)

cube = []
```

```
for frame in raw_science_frames:
    frame = eclipse.trim_and_overscan(frame)
    frame = eclipse.image_sub(frame, bias)
    frame = eclipse.image_div(frame, flat)
    stats = eclipse.iter_stat(frame)
    frame = eclipse.image_div(frame, stats.median)
    cube.append(frame)

fringemap = eclipse.cube_avg_median(fringemap)
stats = eclipse.stat(fringemap, pixmap=mask)
fringemap = eclipse.image_div(fringemap, stats.avg_pix)
fringemap = eclipse.image_sub(fringemap, 1.0)
fringemap = eclipse.image_mul(fringemap, mask)
```

QC Parameters:

```
mean   : mean pixel value of the fringe map
median : median pixel value of the fringe map
stdev  : standard deviation of the fringe map pixel values
```

5.4.6 Req. — *Cat I: Flat-field - master flat*

Objective:

Determine the master flatfield, to be used to correct for the pixel-to-pixel gain variation in the raw image data.
Three different measures of the variation in the gain are available: the dome flat (**req.**542), the twilight flat (**req.**543) and the night-sky flat (**req.**544).
A suitable choice of the final master flatfield, based on a combination of one or more of these flatfields, will be implemented.
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 flatfield. 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 flatfield and can be used to improve the quality of the master flat.
This master flatfield could then be used to flatfield the science and standard images in the image pipeline.
Experience at FORS has shown that a suitable combination of twilight and night-sky flats provided the best determination of the gain variation.
The master flatfield is proportional to the inverse variance in the flatfielded data and can therefore be used to build a weight image. Individual weight images (**seq.– 633**) assign a weight of zero to bad pixels (**req.**522, **req.**535), saturated pixels, pixels that have a relative gain outside a user defined range, cosmic-ray events and satellite tracks.

Fulfilling or fulfilled by:

Additional data reduction of flat fields obtained by **req.**542 543 544.

When performed/frequency:

daytime, daily

Estimated time needed:

Observation: None. Reduction: 3 min./CCD.

Inputs:

CalFile– 522 *Hot pixel map*
CalFile– 535 *Cold pixel map*
CalFile– 542 *Master Domeflat frame*
CalFile– 543 *Master Twilightflat frame*
CalFile– 544 *Nightsky flat frame*

Outputs:

CalFile– 546 *Master flatfield*

The master flatfield is used in **seq. 632** *Trim, debias and flatfield* and **seq. 633** *Construct individual weights.*

Recipe:

```
Master_Flat -d domeflat -t twilightflat [-n nightskyflat] [-c cold] [-h hot]

domeflat     : the master domeflat
twilightflat : the master twilightflat
nightskyflat : the nightskyflat
cold         : the cold pixel map
hot          : the hot pixel map
```

CA:

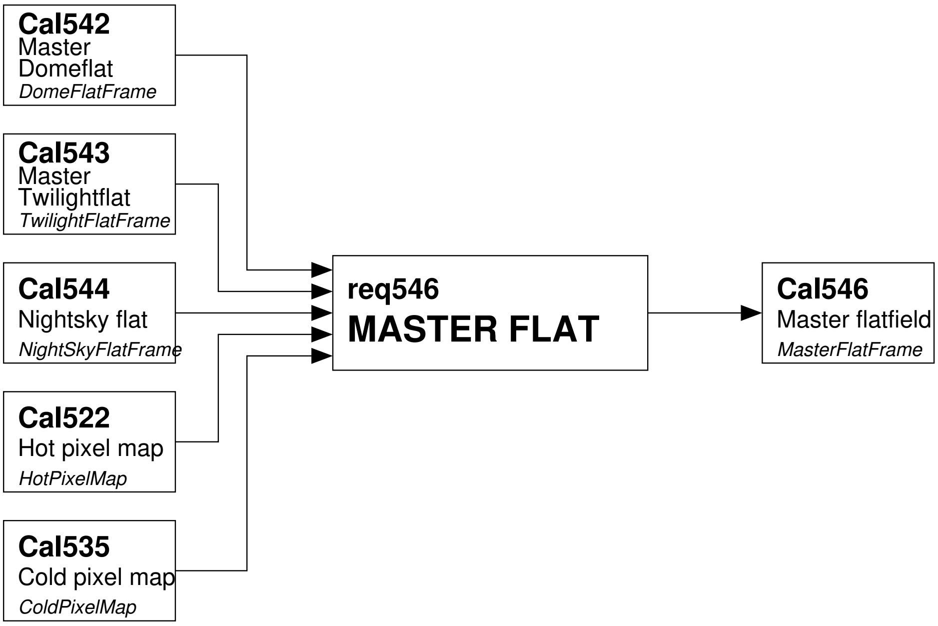


Fig 5.4.6 Dataflow and object class names (in small italic font) for req546

Process (make):

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 components. The low spatial frequencies of the twilight flat are then multiplied by the high spatial frequencies of the dome flat.

Low spatial frequencies are extracted as follows:
1. All bad pixels in input images are replaced by the median value of the pixels in a box around the bad pixel.
2. To reduce problems with Fourier filtering near image edges the size of the image is increased by mirroring the edges and corners.

3. 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).
4. The Fourier transform of the image is multiplied by the Gaussian filter.
5. The image is transformed back, and the mirrored regions removed.
6. The resulting image is normalized, excluding bad pixel values.

CAP:

```
Constants:
  DIG_FILTER_SIZE     : standard deviation of Gaussian used to filter
                        value: 9.0
  MIRROR_XPIX         : width of region in x to mirror
                        value: 75
  MIRROR_YPIX         : width of region in y to mirror
                      : value: 150
  MEDIAN_FILTER_SIZE  : size of box around bad pixels to median filter
                        value: 36

filter = eclipse.image_gen_lowpass(DIG_FILTER_SIZE)

twilightflat = eclipse.image_clean_deadpix_median(twilightflat, MEDIAN_FILTER_SIZE)
twilightflat = eclipse.image_mirror_edges(twilightflat, MIRROR_XPIX, MIRROR_YPIX)
twlt_comp = eclipse.image_fftw(twilightflat, direction=-1)
twlt_comp = eclipse.image_mul(filter, twlt_comp)
twlt_comp = eclipse.image_fftw(twlt_comp, direction=1)
stats = eclipse.stat(twlt_comp, pixmap=mask)
twlt_comp = eclipse.image_div(twlt_comp, stats.avg_pix)

domeflat = eclipse.image_clean_deadpix_median(domeflat, MEDIAN_FILTER_SIZE)
domeflat = eclipse.image_mirror_edges(domeflat, MIRROR_XPIX, MIRROR_YPIX)
dome_comp = eclipse.image_fftw(domeflat, direction=-1)
dome_comp = eclipse.image_mul(filter, dome_comp)
dome_comp = eclipse.image_fftw(dome_comp, direction=1)

dome_comp = eclipse.image_div(domeflat, dome_comp)
masterflat = eclipse.image_mul(twlt_comp, dome_comp)
stats = eclipse.stat(masterflat, pixmap=mask)
masterflat = eclipse.image_div(masterflat, stats.avg_pix)
if nightskyflat:
    masterflat = eclipse.image_mul(masterflat, nightskyflat)
```

QC Parameters:

```
mean   : mean pixel value of the master flat
median : median pixel value of the master flat
stdev  : standard deviation of the master flat pixel values
```

5.4.7 Req. — *Cat I: Quick detector responsivity check-doit*

Objective:

Quickly check the overall health in terms of responsivity by observing the dome screen with the composite filter.
Together with **req. 521** *Read-noise* this item forms the most important day-to-day health check. The expected lamp intensity is characterized in **req. 542** *Dome flat*. This measurement will lead to a go/non-conformance flag and day report. The results will have to be inspected on the site, as this is a daytime health check of the instrument.
Trend analysis on the raw data will be redundant with that of **req. 542** *Dome flat.*
The equivalent of this **req.** on the sky is provided by **req. 562** *Photometric Calibration - monitoring*

Fulfilling or fulfilled by:

Selfstanding

When performed/frequency:

Commissioning, daytime, every day of operations both during CP and RP.

Sources, observations, instrument configurations:

Dome flat with composite key filter

Required accuracy, constraints:

1%

Estimated time needed:

Observation: 3 min/day. Reduction: 1 min./CCD.

TSF:

Mode– Stare N=1
(**TSF– OCM_img_cal_domeflat,** N=1, filter= composite)
= **TSF– OCM_img_cal_quick**

Inputs:

2 raw dome flatfields **CalFile– 541** *Master Bias frame* **CalFile– 542L** *Dome Lamp*

Outputs:

CalFile– 547 *Quick check* **CalFile– 547r** *Quick check - day report*

Recipe:

```
Quick_Check -i raw_domeflat_1 raw_domeflat_2 -b bias [-oc OVERSCAN_CORRECTION]

raw_domeflat_1 raw_domeflat_2 : two raw dome flats
bias                          : master bias frame
OVERSCAN_CORRECTION : overscan correction mode (integer).
                      Description of allowed values:
                        0: apply no overscan correction (default)
```

```
1: use median of the prescan in the
   x-direction
2: use median of the overscan in the
   x-direction
3: use median of the prescan in the
   y-direction
4: use median of the overscan in the
   y-direction
5: use the per-row value of the prescan in
   the x-direction
6: use the per-row value of the overscan in
   the x-direction
```

Before applying this recipe, use **Recipe– Split**—which is documented in **seq.– 631**—with the `-t dome` option to split the raw multi-extension FITS input files.
also run on-site

CA:

Process (make):
1. trim, overscan-correct the raw dome flat fields.
2. de-bias the trimmed dome flat fields.
3. divide the de-biased images.
4. compute the image statistics.

CAP:

```
# trim, overscan, de-bias
dome_1 = eclipse.trim_and_overscan(raw_domeflat_1)
eclipse.image_sub_local(dome1, bias)
dome_2 = eclipse.trim_and_overscan(raw_domeflat_2)
eclipse.image_sub_local(dome2, bias)

quick_check_image = eclipse.image_div(dome1, dome2)
quick_check_statisics = eclipse.iter_stat(quick_check_image)
```

QC Parameters:

```
mean   : mean pixel value of the quick check image
median : median pixel value of the quick check image
stdev  : standard deviation of the quick check image pixel values
```

5.4.8 Req. — *Cat II: Illumination correction*

Objective:

Characterize the illumination correction.

The zeropoint is determined individually for each CCD in **req.** 563. The gain variation over individual chips is characterized by the twilight and sky flatfields (**req.** 543 and **req.** 544) under the assumption of an ideal flat illumination over the field of view. In practice this ideal flat illumination can be affected by stray light (sky concentration) and a correction for this effect has to be made.

An initial verification during commissioning whether this effect is indeed present will be obtained by quickly deriving many zeropoints for individual stars in a standard field (eg from the INT-La Palma preparatory programme). This technique, which is fully automatic, could also be used to quickly verify the effect during RP.

In case the effect is found to have an amplitude over a single chip larger than 1% , it has to be characterized using series of measurements of a standard field with 32 pointings, such as done in **req. 569** *Secondary Standards*.

The map that results from the (interactive) characterization of the illumination correction is used to make an illumination correction frame for every individual chip (**CalFile– 548**).

Fulfilling or fulfilled by:

Initial verification of effect during Commissioning : selfstanding
Characterization of the effect : additional data reduction of **req. 569** *Secondary Standards*

When performed/frequency:

First verification of effect during early commissioning. Characterization at end of Commissioning. During RP, once/month to be determined by experience.

Sources, observations, instrument configurations:

Standard equatorial field

Required accuracy, constraints:

better than 1% for the amplitude over a single CCD.

Estimated time needed:

Verification of effect during Commissioning : 0.5 hour.

Inputs:

The inputs for deriving the illumination correction frame:
CalFile– 548F *Illumination correction fit coefficients*
The name of the chip for which to derive the frame

The inputs for deriving a quick-fit to the effects of stray light (Category III recipe):
Fully calibrated image of a equatorial standard field
CalFile– 569E *Primary Standard stars*

Outputs:

The output from deriving the illumination correction frame:
CalFile– 548 *Illumination correction*
The output from a full, interactive characterization of the effects of stray light:
CalFile– 548F *Illumination correction fit coefficients*

Recipe:

For deriving an illumination correction frame (**CalFile– 548** *Illumination correction*) out of fit coefficients:

```
Illumination_Correction -i illumination_correction_fit -c chip_name

illumination_correction_fit : the map containing the fit made to the
                              illumination variation.
chip_name                   : the name of the chip for which the frame
                              must be derived.
```

For a quick evaluation of the effects of stray light - CATIII :

```
Illumination_Correction_Verify -i <science_frames> -s standard_catalog

science_frames   : list of de-biased, flatfielded, and astrometrically
                   calibrated frames of a standard field.
standard_catalog : standard star catalog.
```

CA:

The description given in the following paragraph refers to the procedure used in the creation of the illumination correction frame (**CalFile– 548**). The steps followed are:
1. Retrieve fit parameters from the illumination correction map.
2. Generate a frame with pixel values determined from these fit parameters.

The description given in the following paragraphs refers to the procedure used in deriving a quick-fit to the effects of stray light. This check is done as an initial verification of the effect during Commissioning (see **req.** 569). The steps to follow are:

1. De-bias, flatfield and astrometrically calibrate the input chips.

The following steps are performed on every separate input chip as reduced in step 1:
2. Derive a source catalog from the input chip (Sextractor).
3. Remove saturated sources from the source catalog.

4. Associate the source catalog with the standard star catalog.
5. Remove all the sources from the source catalog that have not been associated with a source from the standard star catalog.
6. Subtract the measured magnitude of the identified standard stars from their magnitude as known from the standard star catalog. The numbers resulting from the subtraction will further be referred to as zeropoints. Store for every standard star its zeropoint together with its x-y pixel position on the chip in a list.
7. Visualise the variation of the zeropoint as a function of the position on the chip.

The full characterization of the illumination correction needs a preliminary reduction of the data for **req.** 569 that preceeds the data reduction for **req.** 569 that is used to derive **CalFile– 569** *Secondary Standards*. For the purpose of characterization, the data must be de-biased, flatfielded and astrometrically calibrated. The full characterization of the illumination correction is done in an interactive analysis. Note that the characterization of the illumination correction and the derivation of **CalFile– 569** *Secondary Standards* are caught in a bootstrap. This analysis must be performed for every filter separately. Obtaining the full characterization of the effects of stray light involves the following steps:

1. Extract catalogs from processed data. Note that many sources in these catalogs will have been observed multiple times.
2. Remove the saturated sources from the catalogs.
3. Associate the catalogs with the standard star catalog.
4. For every source that has been observed more than once, collect its different positions in the focal plane and combine these with its different measurements of the zeropoint. This produces for every such source a set of differential zeropoints as a function of its position in the focal plane. This information combined for all sources traces the variation of the zeropoint (note that in the end, the illumination correction is charaterized using relative photometry of the stars, since we are determining the variation of the zeropoint over the FoV).
5. From the variation of the zeropoints over the FoV, derive a map of the illumination correction as a function of the position in the focal plane (**CalFile– 548F**).

CAP:

```
The description given in this section refers to the procedure used in the
creation of the illumination correction frame.

fit_coeffs = retrieve_fit_coefficients_from(map)
image_generator = eclipse.image_generator.image_generator(xsize, ysize)
illum = image_generator.generate_poly2d(fit_coeffs)
```

QC Parameters:

TBD

5.5 Astrometric Calibration

The aim of the astrometric calibration is to accurately determine the projection matrix for converting pixel positions to sky coordinates. This will be done automatically in the data reduction pipeline.

For **Mode– SSO** N= the astrometric solution can be derived provided some observational limits, restricting the curvature of the stellar track, are met. When stellar tracks become more curved than the limits, all astrometric quality checks will have to be relaxed. All source positions will be derived using the first-order moments of the pixel distribution of the detected objects. The resulting positions, in such cases, will be the weighted averages of the stellar tracks and that of the (than not) moving targets. The exposure time limitation is defined by the curvature of the stellar tracks in the field, which should be less than 1% over the full track. Estimates for limitations on tracking for **OmegaCAM** are defined by requiring the curvature of the stellar tracks to be less than 0.1 FWHM of the PSF ($< 0''6$). Assuming tracking in Ra, a track length of less than 0.1 degree at any declination is allowed. For a track length of 0.2 degree only declinations between 0 and +/-20 deg or +/-70 to +/-90 deg are within the above limits. For the other modes there are no such restrictions.

The astrometric calibration can be derived using two fundamentally different methods.

The first, traditional, method is to derive the full projection matrix separately for each pointing, ignoring any detailed prior knowledge (apart from some rough initial estimates such as pointing, orientation and platescale). This method has great freedom and allows for instrument independent determination of the projection matrix. It can thus be used on a variety of instruments, but requires more input data elements to achieve an accuracy equal to that of the next method.

The second method allows for a separate determination—and use—of instrument specific characteristics of the geometry that do not change (much) among different pointings and the characteristics that do change with pointing. Fixed characteristics are, e.g., position of the chips relative to each other, the position of the rotator axis, the optical deformation at the position of the focal plane and perhaps pointing accuracy. Variable characteristics include items like the flexure of the telecope and its instrument. The latter items need to be determined for each pointing separately. This method can be robust and has less degrees of freedom facilitating accurate astrometry with a small amount of data points. It is, however, tightly fixed to the instrumental geometry. A geometric model must be obtained and associated parameter values must be determined before standard astrometric reduction can be done.

Both methods will make use of an astrometric reference catalog. Current catalogs have positional accuracies that are not extremely high on the scale of the field of view of **OmegaCAM**. To achieve higher positional accuracy within a given pointing set and to allow accurate co-addition without degrading the PSF the astrometric solution will make use of the overlap among the pointing set.

The methods can be applied both to the main camera CCD chips and to the guide CCDs, as the latter can be viewed as a auxiliary CCDs of the main camera.

The requirements outlined here determine the calibration data necessary to perform an astrometric solution using prior knowledge of instrument specific characteristics.

The algorithm for the derivation of the astrometric solution is detailed in Section 7.3, while its use is detailed in **seq.– 634.**

5.5.1 Req. — *Cat III: Position of Camera in focal plane*

Objective:

Determine the position of the chips with respect to the optical axis of the telescope. This is part of the static astrometric calibration of the camera. It involves the determination of the chip position, scale, and orientation with respect to a perfect pixel plane. This has to be done with the ADC in and out.

This procedure produces the astrometric pre-solution, allowing for a determination of an approximate pixel scale, distance of the reference pixel with respect to the optical axis and location and rotation of each CCD on the detector plate. In fact, the expected pointing error and other a-priori positional offsets are expected to be small; hence the standard astrometric solution can already be obtained without a pre-solution. However, a first inspection and verification of the pre-solution is a task to initialize the system.

The parameters to be derived are obtained in the standard image pipeline astrometric calibration procedure. These are in turn communicated to the Instrument Software, which writes these values as Fits headers (e.g. CDELT(2x), CRPIX (2x), CDMATRIX (4x). These FITS header keywords are used in the image pipeline as pre-solution/input parameters for the automated astrometric calibration (**seq.– 634**).

In fact, there are two important axes, and they must agree. If there is field distortion in a symmetric pattern about a point that is not the mechanical rotator axis then the instrument rotator will introduce image distortions in the outskirts of the field. The instrument rotator is verified in **req. 566** *Dependency on angle - ADC/rotator reproducability.*

The axes determined from this req and that of req 566 will be compared.

Fulfilling or fulfilled by:

Selfstanding

When performed/frequency:

Each mechanical change of the camera. Each user supplied filter, once a year. Commissioning.

Sources, observations, instrument configurations:

High object density (but unconfused) areas such as open clusters. Standard areas, possibly overlapping with standard star fields. All filters and the two optical configurations ADC in and out should be exercised.

Required accuracy, constraints:

Internal precision: 0.3 pixel. External precision limited by reference catalog

Estimated time needed:

Observation: 2 hours

TSF:

Mode– dither N=5
TSF– OCM_img_obs_dither, each filter, ADC In/Out

Inputs:

CalFile– in1 *Astrometrical reference catalogue*, e.g. US Naval, or GSC2

Outputs:

seq.– 634 Communicate relevant parameter values of FITS keywords to Instrument software.

CA:

Run standard image pipeline. Run standard image pipeline, but co-addition is not required. **SeqFile– 634C** *Astrometrically calibrated catalogue* produced by the standard pipeline contains the relative positions of all the chips. Print offsets etc in **SeqFile– 634C** *Astrometrically calibrated catalogue* and compare with focal plane design values (previous versions of 634C).

5.5.2 Req. — *Cat III: Telescope Pointing and offsetting*

Objective:

Verify the pointing and the offsetting of the telescope for both optical configurations (ADC in and out).
The pointing model is provided independent of the **OmegaCAM** S/W, but a verification of both the pointing and the offsetting accuracy is required.
Perform on-site spot checks of the pointing model. The data from the Guider CCD is also used for this- visual inspection.
Also in the data reduction pipeline, as a standard check in the astrometric solution, the pointing error is determined.

Fulfilling or fulfilled by:

Selfstanding.

When performed/frequency:

Commissioning, after each change of the pointing model, and to be determined by experience.

Sources, observations, instrument configurations:

Standard field

Required accuracy, constraints:

1 arc second

Estimated time needed:

10 min

TSF:

Stare mode, dither and jitter mode. In fact all observations can be used.

Inputs:

Offset Guide CCD

Outputs:

Conformance flag.

CA:

Run *standard pipeline* and inspect parameter "reference position with respect to targetted pointing" of the astrometric solution. The parameters to be derived are generated by the standard astrometric calibration procedure. The Interactive Analysis here concentrates on the collection and interpretation of the automatically generated parameters (see **seq.– 634**).

5.5.3 Req. — *Cat III: Telescope and Field Rotator tracking*

Objective:

Verify that the rotator performs properly and simultaneously that the telescope is tracking correctly.

Up to Zenith distances of 60 degrees and wind speed of 18 m/s with a dynamical component of 30%, the free tracking of the telescope shall be better than 0.2 arcsec rms. With closed-loop autoguiding, the rms deviation shall not exceed 0.05 arcsec

Two kinds of image deformations will be inspected.
First, check at various telescope positions the global performance of the rotator plate which is driven by the pointing model. When the rotator plate is not performing optimal, the objects are *elongated* in a circular pattern (concentric rings) with the rotator plate axis at the center. This inspection is done using **req. 554** *PSF Anisotropy*.

Second, check for the tracking of the telescope to find functional dependency with telescope position. This is purely a verification. As an internal check, do this for each **OmegaCAM** observing mode. This inspection is also done using **req. 554** *PSF Anisotropy*. When the telescope is not tracking correctly the shapes of stellar objects are systematically elongated. The amount of elongation may not exceed a certain value, corresponding to the basic tracking requirements given above.

The offset information from the Guider CCD's is another element in the functional check. When offsets for guide stars are becoming too large, rotator plate errors or telescope tracking errors are apparent as different patterns of offsets during an exposure time.

Positional information on single objects can be determined at 0.02 arcsec rms. Thus for a minimum of 100 objects the rms deviation of 0.2 arcsec rms for open-loop tracking and 0.05 arcsec for closed-loop autoguiding is readily available. A series of short exposures will give at least such numbers of objects for analysis, but longer exposures can yield as many as 100.000 objects for analysis allowing a higher signal to noise ratio for the determination of rotator plate and tracking errors.

Rotator plate determination should be carried out near zenith to allow for the largest field rotation possible during an observation. For moderate exposure times of a few minutes the field rotation can be sufficient (several tens of degrees equals one tenth of a full rotation) to produce detectable errors to verify the required accuracy.

Fulfilling or fulfilled by:

Data reduction of standard science observations (almost all will do).

When performed/frequency:

Commissioning, at each change of the pointing model, in RP to be determined by experience. In CP check once with the ADC in and out.

Sources, observations, instrument configurations:

Most observations can be used.

Required accuracy, constraints:
VST requirements:
free tracking better than 0.2 arcsec r.m.s.
autoguiding tracking better 0.05 arcsec

Estimated time needed:
First method: 2 min, Second method: 0.01 sec.

Outputs:
Conformance flags
Source extraction files
Source extraction calfiles (configuration, parameters files)

CA:
Run Recipe of **req. 554** *PSF Anisotropy*
Do an interactive analysis on the output **CalFile– 554**.
Note:
Inconsistent Rotator Plate motion can be detected by inspecting the systematic behaviour of stellar object elongation and position angle. The pattern of elongation/position angle reveals a concentric orientation when the rotator plate is not revolving correctly during an exposure. Furthermore, the elongation of stellar objects should progressively increase from the rotation axis of the plate outward. When this pattern is recognized at a significance level set by the instrument (PSF anisotropy) a non-conformance flag should be raised.
solve f(Alt,Az) = h(Theta) : Theta < 1'
Inconsistent telescope tracking can be detected by inspecting the systematic behaviour of stellar object elongation. Therefore the objects shape parameters are inspected and the systematic behaviour of elongation parameters, e.g. position angle and elongation of the bounding ellipse are inspected. Whenever the ensemble of objects with a significant ($> 3\sigma$ elongation not equal 1 and a $> 3\sigma$ position angle not equal 0, where the two values are dictated by the known performance of the instrument (PSF anisotropy) a non-conformance flag should be raised.

5.5.4 Req. — *Cat III: PSF Anisotropy*

Objective:

Determine the PSF anisotropy.
Detailed characterization of the Point Spread Function at various positions in the focal plane shall be provided. Monitor optical defects and possible variations in time. Do this for both optical configurations of the telescope, ADC in and out.

Fulfilling or fulfilled by:

Data reduction of observations of high density field, such as employed in **req.**551.

When performed/frequency:

Commissioning
Each optical change to the telescope,
Remount of the detector assembly. Once per three months.

Sources, observations, instrument configurations:

High density fields observed with **Mode– Stare** N=1.

Required accuracy, constraints:

better than 1%

Estimated time needed:

Few minutes per pointing. Reduction: < 3 min.

TSF:

Mode– Stare N=1
TSF– OCM_img_obs_stare

Inputs:

raw science images
or
processed science images

Outputs:

CalFile– 554 *PSF anisotropy*

Recipe:

```
PSF_Anistropy -i <filenames> [-s] [-d DETECT_THRESH] [-f FLAG_MAX]
               [-m FLUX_MAX] [-r MIN_FLUX_RADIUS] [-rm MAX_FLUX_RADIUS]

filenames        : list of raw or processed science images
-s               : switch to not run sextractor if catalogs already exist
DETECT_THRESH    : sextractor detection threshold (float)
                   Range of allowed values: 0.0 - 100.0. Default 10.0.
FLAG_MAX         : maximum flag value for sources (integer)
                   Range of allowed values: 0 - 255. Default 0.
FLUX_MAX         : maximum maximum pixel value for sources (integer)
                   Range of allowed values: 0 - 70000. Default 60000.
```

```
MIN_FLUX_RADIUS : minimum half-light radius for sources (float)
                  Range of allowed values: 0.0 - 50.0. Default 0.0.
MAX_FLUX_RADIUS : maximum half-light radius for sources (float)
                  Range of allowed values: 0.0 - 50.0. Default 50.0.
```

CA:

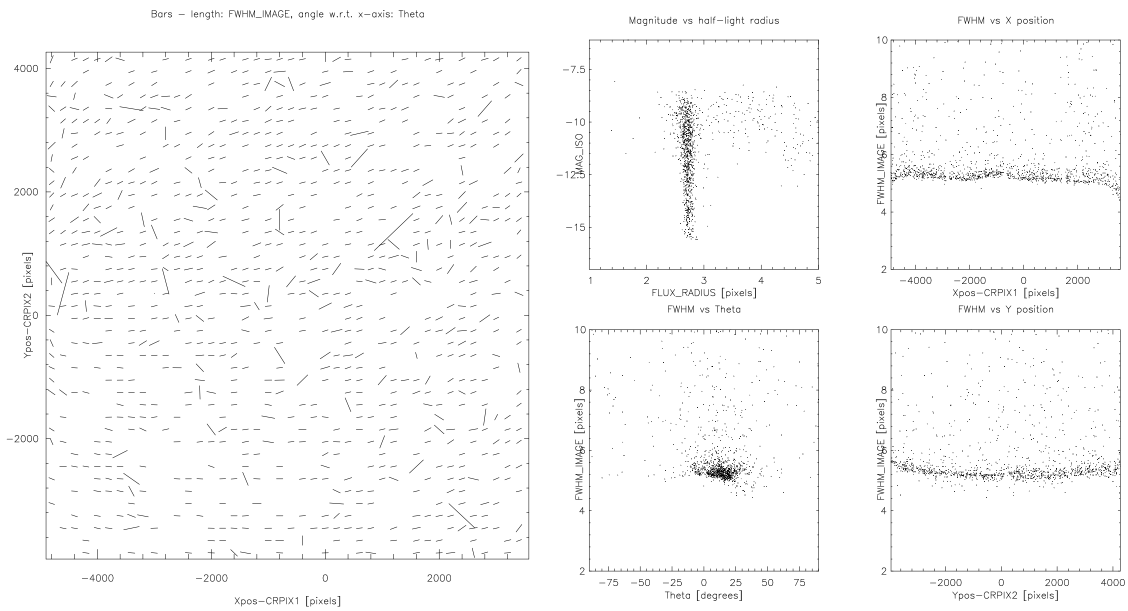


Fig 5.5.4 Example of default output plot of req554

Run standard pipeline reduction.
Do source extraction on output image, using Sextractor.
During the beginning of Commissioning source extraction can be done on raw data. Use the FWHM of stellar objects to search for systematic effects across the detector area. The distribution of the position angles and FWHM of the objects ellipse has to be evaluated. This is an interactive procedure, nearly identical to the one used for the determination of the tilt in **req. 571** *Camera Focus/tilt*

CAP:

```
# Create catalogs of images
catalogs = []
for filename in filenames:
    cat = Catalog()
    cat.frame = BaseFrame(pathname=filename)
    cat.threshold = DETECT_THRESH
    cat.sexparam = ['Xpos', 'Ypos', 'A', 'B', 'THETA', 'FWHM_IMAGE',
                    'FLUX_RADIUS', 'FLUX_MAX', 'MAG_ISO', 'ERRTHETA_IMAGE']
```

```
    # Run SExtractor
    cat.make()
    catalogs.append(cat)

# Filter and merge catalogs
for cat in catalogs:
    LDAC.filter(incat=cat.filename, outcat=cat.filename, table='OBJECTS',
                sel='(A > 1.0 AND B > 1.0 AND Flag < %d AND MaxVal < %d AND
                     FLUX_RADIUS > %d AND FLUX_RADIUS < %d')' % (FLAG_MAX+1,
                     FLUX_MAX, MIN_FLUX_RADIUS, MAX_FLUX_RADIUS))
LDAC.ldacpaste(' ',join([cat.filename for cat in catalogs]), 'cal554.cat')

# Create plot of the source parameters
...

this produces a catalog, which can be visualised in several ways. By default
a plot is produced.
```

5.5.5 Req. — *Cat I: The astrometric solution for templates - doit -see 6.3.4*

Recipe:

See Section 6.3 (**seq.– 634**).

5.5.6 Req. — *Cat III: The astrometric solution for Guide CCD's*

Objective:

Perform astrometric solutions for the Guide CCD's and hand over the solution to the Instrument S/W for locating Guide stars.
Note, the Guide CCDs can be read out separately and 'stand alone'. For this task INS should stream the FITS images of the Guide CCD's to the hard disk.

Fulfilling or fulfilled by:

Selfstanding

When performed/frequency:

Commissioning.

Sources, observations, instrument configurations:

Special readout (manual command) of the Guide CCD's into FITS files

Required accuracy, constraints:

1 arcsec rms for the accuracy with respect to the external standard;
External precision is driven by the position reference catalog. This is in the case of the USNO-A2 catalog of the order 0.3" with possible systematic excursions to 1".

Estimated time needed:

The images used are the Guide CCD images. For this the exposure time of the Guide CCD needs to be comparable to the Science CCD to ensure enough (gt 40) stellar images for astrometric calibration. Total time during Comm is about 1 hour.

Inputs:

Reference position catalog (see A4)

Outputs:

CalFile– 556 *Guide CCD guide star signal and offset*
astrometry solutions inserted into the descriptors; handed over manually to instrument S/W responsible

CA:

The parameters to be derived are extracted using the standard astrometric calibration procedure. The images that are input to this procedure are nowthe Guide CCD images instead of the usual Science CCD images. For this the exposure time of the Guide CCD needs to be comparable to the Science CCD to ensure enough (gt 40) stellar images for astrometric calibration (see **seq.– 634**).
See **req.**555.
Here, the required accuracy is not as high as for the science images. Therefore it suffices to restrict the degrees of freedom to incorporate linear effects per CCD only.

CAP:

```
input flat_fielded_images, reference_cat
foreach image in flat_fielded_images
  extract_cat = image.source_extract()
```

```
  pairs_cat   = extract_cat.pattern_match(reference_cat)
  astrom_solv = pairs_cat.solve(PDeg=n, Iter=m)
  calib_image = image.head(astrom_solv)
done
output calib_images
```

5.6 Photometric Calibration

The basic requirement for the photometric calibration of the broad-band filters is to achieve an accuracy of better than 5% on the photometric scale in 'instrumental magnitudes' as assigned to the units of the resultant output image of the "image pipeline". The accuracy of the colour transformation terms of instrumental to standard systems should be better than 10% on the photometric scale.

In order to maintain this accuracy on a routine basis over years of operation, a set of requirements are specified in this section.

The descriptions of these requirements involve the following **OmegaCAM** specific ingredients:

- **key passbands** ($X =$ u', g', r' and i' in Sloan system)
- **two lens correctors** (near Zenith, the baseline, key configuration) and an **atmospheric dispersion corrector -ADC** for operations in User mode at larger Zenith angles.
- a **composite key filter** ($X =$ u', g', r' and i' in each quadrant),
- a standard **polar field**, observable throughout the year,
- **8 equatorial fields**, containing both primary and secondary standard stars (Landolt fields - see section 7.1)
- a **dome lamp** and a **fixed dome screen** equipped with a stabilized current supply,
- **32 CCD's** are operated simultaneously, with the exception of the composite filter which 'feeds' 8 CCDs simultaneously in one passband.
- data rates should stay within limits that allow processing and storing of the data with the currently anticipated technology.
- A **standard atmospheric extinction curve** is adopted and all atmospheric extinction in various passbands is taken as a scaling of this curve.
- The photometric monitoring employs observing **strategy – freq** which has overriding priority on the scheduling and which employs observing **mode – stare** and its associated **trend analysis**.

The prime concept of the **OmegaCAM** photometric calibration is to *continuously* maintain the photometric scale in the keybands, even when the science programme does not require the usage of these passbands during a particular night or period. This continuity is used by the data reduction (calibration and its trend analysis) and is meant to ease the maintenance of the photometric system on a routine basis.

The usage of a standard extinction curve results into a high rigidity of the pipeline processing, and provides a tool for error estimates, quality checks, recognizing non-conform data and provides a framework for successful pipeline processing of incomplete data.

We model the characterization of the photometric system in terms of a series of gains, where for each aspect of the calibration we distinguish a gain g_0 at a pre-determined fixed moment and the variation of that gain as function of time $g(t)$, the latter being mostly analyzed by a **trend analysis**.

All photometry is determined on an individual CCD chip basis ($N =$ number of chip, 1...32), apart from the atmospheric extinction which is common for all chips.

Most gains depend on passband (X), but not the variation of the atmospheric extinction, which is assumed to scale with the standard extinction curve.

In Figure 5.6 an overview is sketched of the various requirements which form the photometric calibration.

The fixed and variable gains of the various calibrations are defined as follows:

Atmospheric extinction: use scaling of standard extinction curve represented by $g_{sel.e}(X)$:
middle of the night: $g_e(0) \times g_{sel.e}(X)$ **req.**563
during the night : $g_e(t)$, t in hours **req.**562

Zeropoint effective DQE - **req.**563
middle of the night: $g_{DQE}(0, N, X)$
on different nights: $g_{DQE}(t, N, X)$, t in nights

Flat field - **req.**542 and others
for period of 7 days: $g_{ff}(week, N, X)$
different weeks: $g_{ff}(t, N, X)$ t in weeks ; every week a new ff is assigned, replacing $g_{ff}(week, N, X)$

For a given star observed in passband X at a given position in the field of view, at a given moment (t) the relation between the output (I_{obs} of the detectors) and the zero-airmass intensity $I_{ref,X}$ is given by the general expression:

$$I_{obs,X} = g_e(0)g_e(t)g_{sel.e}(X) \times g_{dqe}(0, N, X)g_{dqe}(t, N, X) \times g_{ff}(week, N, X) \times I_{ref,X}$$

Colour term:

The primary standard stars have been measured with presumably the same filter passbands, but with CCD detectors which have a different relative spectral responsivity. This implies that the effective $g_{DQE}(0, N, X)$ when observing these primary stars depends on the colour of these stars: $g_{DQE}(0, N, X, X - x)$.

The photometric calibration involves the solution of the general equation above along a different path, with different unknowns, and different knowns.

The initialization of photometric calibration is to carefully process backwards and forwards through the basic equation. Particularly the settling of the secondary standards, which cover a larger field of view than the primary standards is tricky. On one hand the preparatory programme will provide this information, on the other hand **OmegaCAM** calibrations can self-calibrate the secondary standards, which has the advantage that it avoids the extra bootstrapping via another telescope and detector system.

Normalizations:
$g_{ff}(week, N)$ unity at central pixel of each chip N
$g_e(0)$ and $g_e(t)$ multiplication factor of standard extinction curve represented by $g_{sel.e}(X)$

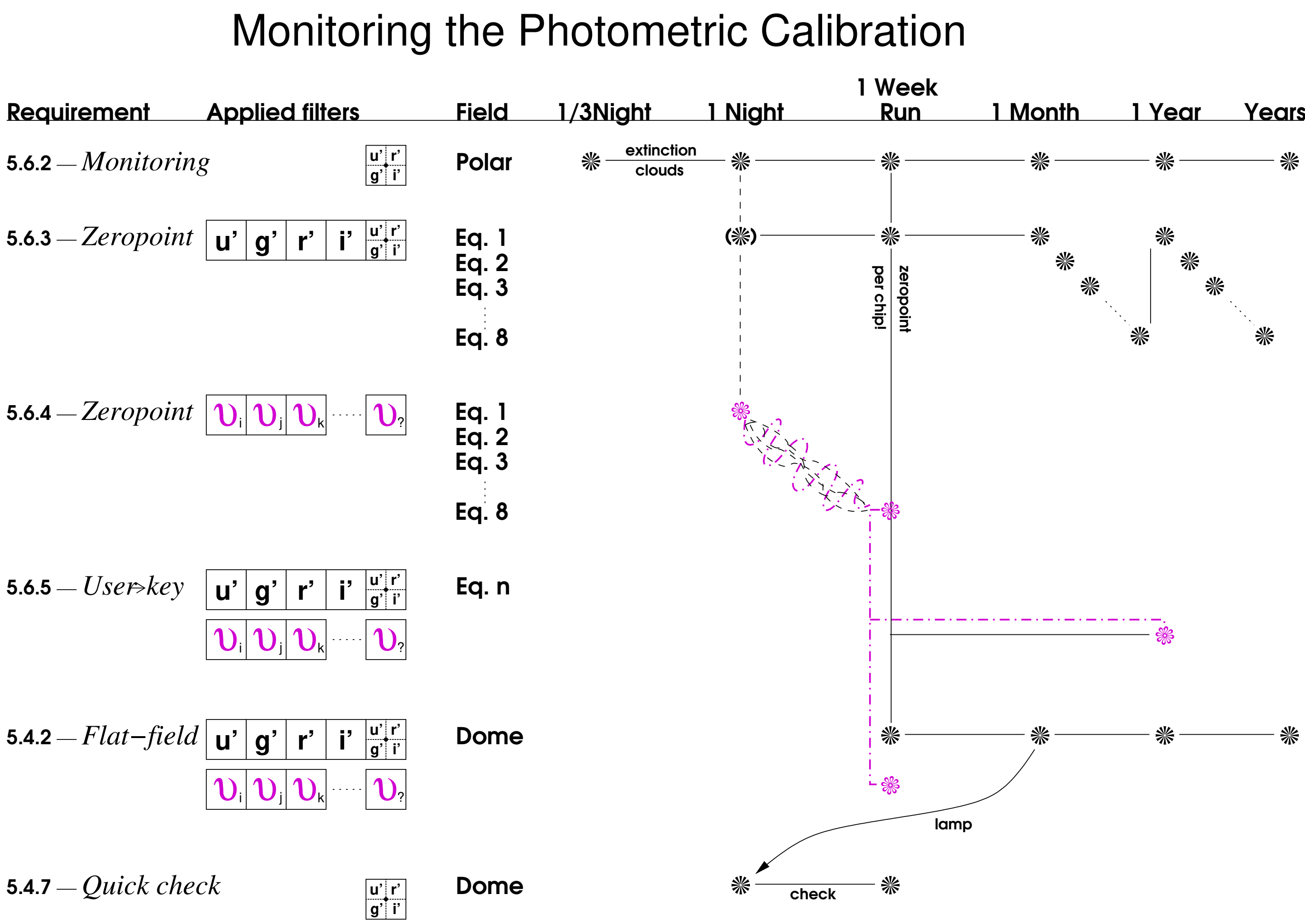


Fig 5.6. *Overview of tracking the photometric calibration at various time scales. From left to right, the figure indicates the requirement number, the requirement name, the used filters and standard fields. The stars indicate at which frequency the measurement is done. For further details, see text.*

5.6.1 Req. — *Cat III: Shutter Timing*

Objective:

Verify the actual timing of the shutter, monitor position dependent delays of the shutter motion in both directions.
Exposure timing has to be accurate to $\pm 0.2\%$ at 1 sec exposure time at any position of the focal plane (**OmegaCAM** Technical Specifications). The exposure timing signal is provided by PULPO. For the following considerations we assume that the PULPO timing signal is accurate to better than 0.1 msec (i.e. it is not the dominating source of inaccuracy).
The shutter mechanism consists of a pair of chasing carbon fibre blades. Their movement is controlled by the Shutter Control Unit (Shutter CU) such that it results in an identical effective exposure time all over the frame. These two movements may or may not overlap in time depending on the exposure time and the blade traveling time.
The opening blade starts moving immediately (μsec's) after the falling edge of the TTL signal (provided by PULPO). This is the beginning of the exposure procedure. The closing blade starts moving immediately (μsec's) after the rising edge signal was detected and ends (about 1 sec later) when the closing blade completely covers the aperture, which marks the end of the exposure procedure. Therefore, the duration of an exposure procedure is always: `exposure time + blade travel time (ca. 1 sec)`
Two types of delays affect the effective exposure time: The delays of the start of the blade movements after the opening/closing TTL signal edge (i.e. absolute exposure time) and position dependent delays during blade movement (i.e. exposure homogeneity).
The open/close delays are up to 0.05 msecs due to signal polling of the Shutter Control Unit software. These values are well within the requirements (shutter open time error: $\pm 0.2\%$ at 1 sec corresponding to ± 2 msecs). Deviations from this occur only in case of a severe shutter failure which is detected by the Shutter CU and PULPO independently followed by operator actions.
Position dependent delays (requirement: 0.2% at 1 sec exposure time) will be monitored in regular intervals of 3 months. This will be done for both shutter movement directions.
Dome flatfields of 10 sec and 0.1 sec exposure time will be taken for both shutter blade movement directions. Illumination level shall be such that the CCD's are at about 60% to 80% full well for the 10 sec exposure. Exposure times will have to be evaluated during Commissioning.

Fulfilling or fulfilled by:

selfstanding

When performed/frequency:

Commissioning, once per 3 month, further to determined by experience. Daytime

Sources, observations, instrument configurations:

Dome flat field with a level of about 40,000 - 50,000 ADU's

Required accuracy, constraints:

Timing error less than 0.2%.

Estimated time needed:
1 hour

TSF:
Mode– Stare N=4
(**TSF– OCM_img_cal_domeflat,** $N = 4$, $t_{exp} = 10.0, 0.1, 0.1, 10.0$)
= **TSF– OCM_img_cal_shutter**

Inputs:
Raw dome flatfields
CalFile– 541 *Master Bias frame*

Recipe:
Recipe– Shutter_Delay

CA:
Interactive analysis with Eclipse After bias correction divide (0.1 sec exp.) / (10 sec exp.) for each shutter direction and average columns (or rows, which is appropriate) to improve S/N (the effect is 1-dimensional).
The result will be inspected for deviations from homogeneity ($\pm 2\%$ at 0.1 sec, i.e. $\pm 0.2\%$ at 1 sec) for both shutter directions. Shutter direction is indicated in header.

CAP:
```
input bias, low_level_flat, high_level_flat
for ShutterDirection up,down
tmp = (low_level_flat-bias)/(high_level_flat-bias)

line = tmp.colapse()

output ShutterDirection,line
```

5.6.2 Req. — *Cat I: Photometric Calibration - monitoring*

Objective:

Monitor any **short term** variability related to the transparency of the atmosphere (atmospheric extinction) or due to instrumental instabilities (e.g. effective DQE) with a minimum sampling of at least 3 times/night. This provides a **daily overall health check** of the instrument and detectors. A further trend analysis has to provide information on **long term** stability.

The variation (r.m.s.) of the flux detected by the autoguider shall also be used as an indicator (put in the FITS header) of the sky conditions. This is to be done for each science observation.

This monitoring is done on a standard **polar field**, which will be repeatedly observed at the beginning, middle and end of the night with the **composite key filter** (u', g', r' and i' band), irrespective of the passbands used for the science observations. The observations are done in the standard configuration, with the two-lens corrector. A comparison of the measured intensity of the stars with reference values is used to qualify the overal conditions of instrument and atmosphere, the actual zeropoint (both unit airmass extinction and instrument DQE) being determined by **req.**563.

The comparison of the observed signal with the expected signal from standard stars in each of the four quadrants will lead to the determination of the product of the atmospheric extinction and the overall effective DQE of the detector system including the optics, i.e. $g_e(t) \times g_e(0) \times g_{st.e}(X) \times g_{DQE}$, with $g_{st.e}(X)$ the gain of the standard extinction curve at passband X, and g_{DQE} the overall effective DQE. As **req.**563 solves for g_{DQE}, a comparison with these measures gives $g_e(t) \times g_e(0) \times g_{st.e}(X)$, the variation of the gain during the night. The thus derived values of $g_e(t) \times g_e(0) \times g_{st.e}(X)$ at $t =$ beginning, middle, and end of the night (could be more if the observer so wishes) provides the required monitoring. Note that in this scheme, $g_e(t) = 1$ by definition for the middle of the night ($t = 0$). Excursions from the standard extinction curve due to extraordinary meteorological conditions, can be traced by computing the standard deviation of observed minus standard curve values in the various bands.

The repetitive measurements on the same field and with the same filter will also be used in trend analysis to monitor the overall long term stability of the instrumentation and atmosphere. The redundancy of these measurements with **req. 563** *zeropoint* and **req. 542** *Flat field -dome* will be used as a cross-check on the validity of the photometric system.

The data taken through the composite filter will provide a simultaneous measurement of the sky brightness in the four different **key bands**; this data will thus give a spectrum of the sky with a resolution R of roughly 5. This sky spectrum shall be derived on-line from the data. This serves as a quality check on the health of the instrument and on the clearness of the atmosphere, as clouds or cirrus will be immediately noticable in the spectral shape. Reference tables containing the expected sky brightness (and thus colour) as function of lunar phase will be used in the evaluation of the data.

Note that small differences exist between the monolithic key-band filters and the composite filter (see **req.**565). This might lead to small color terms that should be taken into account if needed.

Fulfilling or fulfilled by:
Selfstanding; a corresponding requirement on detector level is
req. 547 *Quick detector responsivity check-doit*

When performed/frequency:
beginning, middle and end of each night; any moment atmospheric conditions are suspect.

Sources, observations, instrument configurations:
The **OmegaCAM** polar field, key composite filter, two lens corrector; short (about 20 sec) integration **Strategy– freq**

Required accuracy, constraints:
all photometry better than 1-2% on the photometric scale.

Estimated time needed:
Observation: 3 times 4 min. (100 sec preset + 100 sec integration + 42 sec readout) totalling 12 min/night each night. Reduction: 10 min./CCD.

Detailed estimate of required integration time:
The secondary standard stars in the standard field have a limiting magnitude g'~ 20. The internal accuracy of the present task is set on 1-2% level in order to achieve an overall end-to-end accuracy of better than 5% on the photometric scale. Clearly, for the composite filter the desired exposure time will be dominated by the response in the u' band. Using the WFI2.2m ETC, corrected for the VST optics and estimated CCD spectral responsivity, we estimate a S/N= 20 after 100 sec integration for a V=20 F0V star at airmass 2 (South Pole) with nominal 1.0 arcsec seeing. Thus, most secondary standard stars will have a S/N better than 20 in the u' band in a 100 sec integration. This would just match to the wanted 5% overall photometric accuracy in the u' band, but would lead simultaneously to higher S/N in the other bands. We compute for a V=20 GoV star S/N=90 in i' new moon, S/N=70 in r' new moon, S/N=45 in r' full moon and S/N =80 in g' new moon. Thus an integration time of 100 sec for the composite filter appears reasonable, but clearly the u' band data give lower photometric accuracy, and the final value will depend on the achieved responsivity in the u' band. The final number has to be determined by experience and depends on the distribution and colours of the stars in the u' quadrant of the field of view.

TSF:
Strategy– freq
Mode– Stare N=1
(**TSF– OCM_img_obs_stare,**N=1, filter=composite)
= **TSF– OCM_img_obs_monit**

Inputs:

The inputs for deriving the photometry result tables are:
Fully calibrated image of the polar standard field
CalFile– 569 *Secondary Standard Catalog*
CalFile– 565C *composite -> monolithic*

The inputs for deriving the transparency of the atmosphere are (Category I recipe):
List of photometry result tables
CalFile– 563 *Zeropoint + extinction - keybands* (g_{DQE} and $g_e(0) \times g_{st.e}(X)$)
CalFile– 564E *Standard extinction curve*

The inputs for the **on-line** monitoring of the sky spectrum are (Category III recipe):
List of input raw images of the polar standard field
List of input masterbias frames
List of input masterflat frames
CalFile– 562Su *Sky brightness-u'* Reference values u'
CalFile– 562Sg *Sky brightness-g'* Reference values g'
CalFile– 562Sr *Sky brightness-r'* Reference values r'
CalFile– 562Si *Sky brightness-i'* Reference values i'

Outputs:

The output from deriving a photometry result table is:
CalFile– 562T *Photom + Sky*

The output from deriving the transparency of the atmosphere is:
CalFile– 562 *Extinction - night report* ($g_e(t) \times g_e(0) \times g_{st.e}(X)$ at var. t)

Recipe:

For deriving a photometry result table:

```
PhotCal_Extract_Resulttable -i science_frame -s standard_catalog
                            [-t transformation_table]

science_frame        : input frame that has been de-biased, flatfielded
                       and astrometrically calibrated. An illumination
                       correction (if needed) should also have been
                       applied.
standard_catalog     : standard star catalog.
transformation_table : transformation table containing color-terms.
```

For deriving the transparency of the atmosphere:

```
PhotCal_Monitoring -i <photometry_result_tables> -z <zeropoints>
                   -c extinction_curve

photometry_result_tables : list of photometry result tables as derived
                           from a list of fully reduced images of a
                           standard field.
zeropoints               : list of zeropoints, one for every combination
                           of photometric band and chip.
extinction_curve         : a standard extinction curve.
```

For the on-line monitoring of the sky spectrum:

```
PhotCal_Monitoring_Sky -iu <raw_science_frames_sloanu_kwadrant>
                       -ig <raw_science_frames_sloang_kwadrant>
                       -ir <raw_science_frames_sloanr_kwadrant>
                       -ii <raw_science_frames_sloani_kwadrant>
                       -fu <masterflat_frames_sloanu_kwadrant>
                       -fg <masterflat_frames_sloang_kwadrant>
                       -fr <masterflat_frames_sloanr_kwadrant>
                       -fi <masterflat_frames_sloani_kwadrant>
                       -b  <masterbias_frames>
                       -s  <skybrightness_tables>
                       -lp lunar_phase
                       [-zu zeropoint_sloanu]
                       [-zg zeropoint_sloang]
                       [-zr zeropoint_sloanr]
                       [-zi zeropoint_sloani]
                       [-oc OVERSCAN_CORRECTION]

raw_science_frames_sloanu_kwadrant : list of raw science frames from the
                                     Sloan u kwadrant.
raw_science_frames_sloang_kwadrant : list of raw science frames from the
                                     Sloan g kwadrant.
raw_science_frames_sloanr_kwadrant : list of raw science frames from the
                                     Sloan r kwadrant.
raw_science_frames_sloani_kwadrant : list of raw science frames from the
                                     Sloan i kwadrant.
masterflat_frames_sloanu_kwadrant  : list of masterflat frames for the
                                     chips of the Sloan u kwadrant.
```

```
masterflat_frames_sloang_kwadrant  : list of masterflat frames for the
                                     chips of the Sloan g kwadrant.
masterflat_frames_sloanr_kwadrant  : list of masterflat frames for the
                                     chips of the Sloan r kwadrant.
masterflat_frames_sloani_kwadrant  : list of masterflat frames for the
                                     chips of the Sloan i kwadrant.
masterbias_frames                  : list of masterbias frames.
skybrightness_tables               : list of skybrightness tables.
lunar_phase                        : lunar phase in days after new moon
                                     (integer). Default: 0
zeropoint_sloanu                   : overall zeropoint of the detector
                                     block for Sloan u (float).
                                     Default: 0.00.
zeropoint_sloang                   : overall zeropoint of the detector
                                     block for Sloan g (float).
                                     Default: 0.00.
zeropoint_sloanr                   : overall zeropoint of the detector
                                     block for Sloan r (float).
                                     Default: 0.00.
zeropoint_sloani                   : overall zeropoint of the detector
                                     block for Sloan i (float).
                                     Default: 0.00.
OVERSCAN_CORRECTION                : overscan correction mode (integer).
                                     Description of allowed values:
                                       0: apply no overscan correction
                                          (this is the default)
                                       1: use median of the prescan in
                                          the x-direction
                                       2: use median of the overscan in
                                          the x-direction
                                       3: use median of the prescan in
                                          the y-direction
                                       4: use median of the overscan in
                                          the y-direction
                                       5: use the per-row value of the
                                          prescan in the x-direction
                                       6: use the per-row value of the
                                          overscan in the x-direction
```

CA:

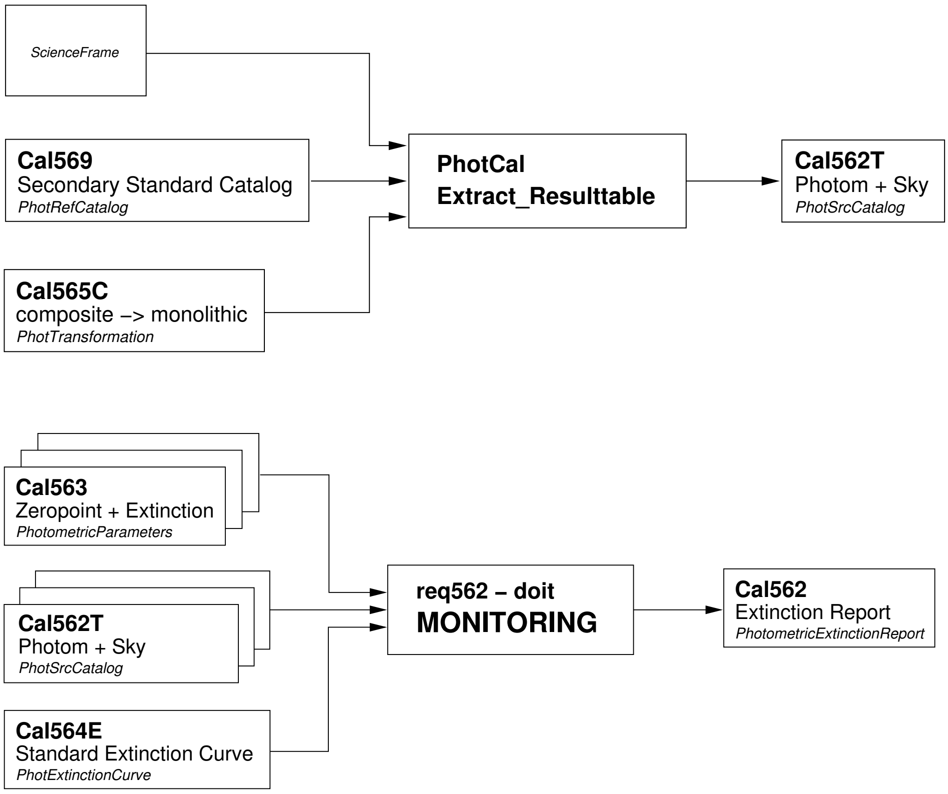


Fig 5.6.2 Dataflow and object class names for req562, monitoring the atmosphere.

The description given in the following paragraphs refers to the procedure used in deriving **photometry result tables**. The procedure can be run on the complete set of monitoring data, but also on a subset of the pointings for the night. The creation of the photometry result tables requires the secondary standard star catalog (**CalFile– 569,** see **req.** 569 for accuracy and constraints). In the creation of the photometry result tables, great care is taken to remove sources from the catalog unsuitable for the photometric calibration (see the steps outlined below).

1. De-bias, flatfield and astrometrically calibrate the input chips. Apply illumination correction (if needed).

The following steps are performed on every separate input chip as reduced in step 1:
2. Derive a source catalog from the input chip (Sextractor).
3. Remove saturated sources and blended sources from the source catalog. Remove sources from the catalog with clipped or otherwise deformed apertures.

4. Associate the source catalog with the standard star catalog (LDAC).
5. Remove all the sources from the source catalog that have not been associated with a source from the standard star catalog.
6. Subtract the measured magnitude of the identified standard stars from their magnitude as known from the standard star catalog. If a correction for color-terms is needed, the latter magnitudes have to be transformed first. The numbers resulting from the subtraction will further be referred to as raw zeropoints.
7. Measure the skybackground from the input chip.
8. Store in a photometry result table (**CalFile– 562T**): 1) the list of raw zeropoints, 2) the skybackground measured from the chip, 3) the airmass of the chip, 4) the observation date of the chip from which the sources where extracted, 5) the name of the key-band, 6) the central wavelength of the filter, 7) the name of the chip.

The description given in the following paragraph refers to the procedure used in **monitoring the transparency** of the atmosphere. The monitoring report for the night is obtained from all the photometry result tables derived previously in the steps 1-8 described above.

1. Sort the set of photometry result tables by the observation time. Sort the tables in these subsets further by key-band. Every combination of time/key-band is one subset of photometry result tables.
2. Sort the set of input zeropoints by key-band.
3. Derive the atmospheric extinction for every single combination of time/key-band. These extinctions are all converted to a value per unit airmass.
4. Sort the derived atmospheric extinctions for the separate key-bands into subsets based on the observation time.
5. Fit the standard extinction curve to each subset of derived atmospheric extinctions.
6. Derive the difference between the fit of the extinction curve to the measured extinctions and the extinction curve itself. This is the shift of the curve.
7. Store for every observation time : 1) the observation time, 2) the calculated shift of the standard extinction curve and its uncertainty, 3) the derived extinction for every key-band and their uncertainties, 4) the difference between the derived extinction and the standard extinction curve for every key-band.

The description in the following paragraph refers to the procedure used in the on-line **monitoring of the sky spectrum**. The procedure is run on one single image of the polar field at a time.

1. De-bias and flatfield every chip.
2. Derive the median sky background from the input chips. This will give 32 values, 8 for each kwadrant of the composite filter.
3. Determine the colors from the median skybackground values, and visualize these colors

together with the values derived from the input tables of skybrightness.

CAP:

```
The description given in this section refers to the procedure used in
monitoring the transparency of the atmosphere. Any small differences
between the composite filter and the monolithic key-band filters
(if any) are not taken into account in this description.

Constants:
  DETECT_THRESH  : Sextractor processing parameter (float). Value: 10.0.
  SIGCLIP_LEVEL  : for sigma-clipping in deriving the atmospheric
                   extinctions (float). Value: 3.0.
  MAX_REL_ERROR  : maximum allowed relative error in the extinction

/* Step 1 : make photometry result tables

(1)      for every frame in the list of science frames:
(1.1)      derive_source_catalog_from_frame:
             Sextractor.sextractor(frame_name)
(1.2)      remove_saturated_stars_from_the_catalog:
             LDAC.filter(catalog_name, filter_criterium_1)
(1.3)      associate_source_catalog_with_standard_star_catalog:
             LDAC.prephotom(catalog_name, refcat)
(1.4)      remove_not_associated_sources_from_catalog:
             LDAC.filter(catalog_name, filter_criterium_2)
(1.5)      create_empty_list_for_raw_zeropoints
(1.6)      for every source in the source catalog:
(1.6.1)      mag = take_magnitude_of_source_from_standard_catalog
(1.6.2)      instmag = take_measured_magnitude_of_source
(1.6.3)      raw_zeropoint = mag - instmag
(1.6.4)      raw_zeropoint_err = sqrt(mag_err**2 + instmag_err**2)
(1.6.5)      add_raw_zeropoint_and_error_to_list
(1.7)      measure_skybackground_from_frame
(1.8)      save_the_photometry_result_table

/* Step 2 : make monitoring report

(1)      result_table_subsets_date = group_result_tables_by_observing_date
(2)      zeropoint_subsets_key_band = group_zeropoints_by_key_band

(3)      create_empty_list_for_report_entries
```

```
(4)       for every result_table_subset in result_table_subsets_date:
(4.1)       result_table_subsets_key_band = group_result_table_subset_by_key_band
(4.2)       create_empty_list_for_atmospheric_extinctions
(4.3)       for every result_table_subset in result_table_subsets_key_band:
(4.3.1)       get_zeropoint_subset_for_this_key_band
(4.3.2)       derive_the_atmospheric_extinction_for_this_keyband:
                create_empty_list_for_separate_extinctions
                for every raw zeropoint in result_tables:
                    subtract_zeropoint_from_raw_zeropoint
                    add_result_to_list
                sigma_clip_the_separate_atmospheric_extinctions
                derive_average_atmospheric_extinction
(4.3.3)       add_atmospheric_extinction_to_list
(4.4)       fit_extinction_curve_to_atmospheric_extinctions:
              least_squares_fits(atmospheric_extinctions, extcurve)
(4.5)       make_entry_in_extinction_report_for_this_catalog_subset
(4.6)       add_entry_to_list
(5)       save_the_monitoring_report
```

QC Parameters:

TBD

5.6.3 Req. — *Cat I: Photometric Calibration - zeropoint keybands - doit*

Objective:

Determine the zeropoint of the overall detector chain, separately for each CCD chip, in all four keybands (no composite filter) and the true atmospheric extinction at midnight by measuring standard stars in the 4 key passbands in one of the eight equatorial fields and the polar field. Do this every night whatever the science programme on the telescope may be. Optionally, add one observation of another equatorial standard field with higher airmass to obtain a redundant, classical, measurement of the atmospheric extinction.

The keybands plus the two-lens corrector form the standard for this requirement, the use of the ADC is considered as User mode (see **req.**564).

The zeropoint corresponding to the DQE of each of the 32 CCD chips will be determined on an individual chip basis $g_{DQE}(0, N, X)$. Thus the composite filter can not be used for this. However, additional data will be acquired with the composite filter for redundancy. In case the relative gain variations of individual CCD chips appear small and well characterized by the overall flatfield (which is not really expected) then the **req.** might be fulfilled with only the composite filter, substantially releaving the data rate and workload.

The combination of the data of **req. 562** *Monitoring* taken at the middle of the night with the present zeropoint data will be used to solve separately for the effect of the extinction and DQE at the middle of the night. A standard extinction curve will be used as a reference, for error analysis, and to support the derivation of the solutions in the pipeline processing.

As **primary standards** the Landolt equatorial fields will be used, possibly extended with preparatory programme results (Section 7 of CP). For each of the 8 equatorial fields a solution for **secondary standard** stars will be made for a larger, one degree, field of view. The acquisition of catalogues of secondary standards is discussed in **req.**569. These secondary standards data will be used for the nightly determination of the zeropoint.

During commissioning the reproducibility of the zeropoint determination should be verified for the different observing modes **Mode– jitter** N=5 and **Mode– dither** N=5. This also serves as an end-to-end-test.

In the case that composite filter data is used for the analysis, the transformation between composite and monolithic key band filters should be taken into acount (see **req. 565** *Filter band passes*).

Fulfilling or fulfilled by:

selfstanding, but uses data of **req.**562 at the middle of the night.

When performed/frequency:

Once in the middle of each night. The linking of external (primary) to internal (secondary) standards will be done once during commissioning, and the first year of operation for each standard field; after that to be determined by experience.

It is to be determined by experience (stability of the system) whether the nightly zeropoint measurements can be relaxed as follows: in case the keyfilters are not used for science observations in a particular night, it is sufficient to only take an exposure with the composite filter.

During commissioning the **Mode– dither** N= and **Mode– jitter** N= have to be verified.

Sources, observations, instrument configurations:

Strategy– freq Mode– Stare N=1 **OmegaCAM** equatorial fields; exposures in u', g', r' and i' and composite. Two lens corrector.

Required accuracy, constraints:

1% on the photometric scale

Estimated time needed:

Observation: 12 min. each night Reduction: 3 min/CCD/filter.

Detailed estimate of required integration time:

The secondary standard stars in the standard fields have a limiting magnitude g'~ 20. The internal accuracy of the present task is set on the 2% level in order to achieve an overall end-to-end accuracy of better than 5% on the photometric scale. Given an airmass of 1.3 and an S/N=50, we expect an integration time of:

u' band 110 sec but S/N =45 for F0V
g' band 60 sec
r' band 75 sec
i' band 115 sec

As there will be many much brighter stars (than the g'=20 limiting mag for which this calculation is made) in the field of view, leading to much better S/N these integration times will be sufficient for the goals, also at other moonphases (actually checked with ETC). The u' band observations are on the limit and should be carefully checked during Commissioning.
Thus a total sequence would last including readout of 42 sec:
preset (100), u'(110 + 42), g'(60 + 42) , r'(75 + 42), i' (115 + 42) = 628 sec
In addition, a composite filter image should be taken (100 sec) which in the long run might replace the u', g', r', i' sequence with monolithic filters.

TSF:

Strategy– freq
Mode– Stare N=1
(**TSF– OCM_img_obs_stare,** N=1, filter=key band)
= **TSF– OCM_img_obs_zpkey**

Inputs:

In pipeline data reduction, the **req.**563 reductions of a given night should preceed the **req. 562** *Monitoring* reductions.

The inputs for deriving the photometry result tables are:
Fully calibrated image of equatorial standard field
CalFile– 569 *Secondary Standard Catalog*

The inputs for deriving the zeropoints for the night are:
List of photometry result tables
CalFile– 562 *Extinction night report*
CalFile– 564E *Standard extinction curve*

Outputs:

The output from deriving a photometry result table is the same as for **req.** 562.

The outputs from deriving the zeropoints for the night are:
CalFile– 563 *Zeropoint + extinction - keybands* (always) used by **seq.– 636.**

Recipe:

For deriving a photometry result table:

```
PhotCal_Extract_Resulttable -i science_frame -s standard_catalog
                             [-t transformation_table]

science_frame        : input frame that has been de-biased, flatfielded
                       and astrometrically calibrated. An illumination
                       correction (if needed) should also have been
                       applied.
standard_catalog     : standard star catalog.
transformation_table : transformation table containing color-terms.
```

For deriving the zeropoints for the night:

```
PhotCal_Extract_Zeropoint -e <photometry_result_tables>
                          -m monitoring_report
                          -c extinction_curve

photometry_result_tables : list of photometry result tables as derived
                           from a list of fully reduced images of a
                           standard field.
monitoring_report        : extinction night report.
extinction_curve         : a standard extinction curve.
```

CA:

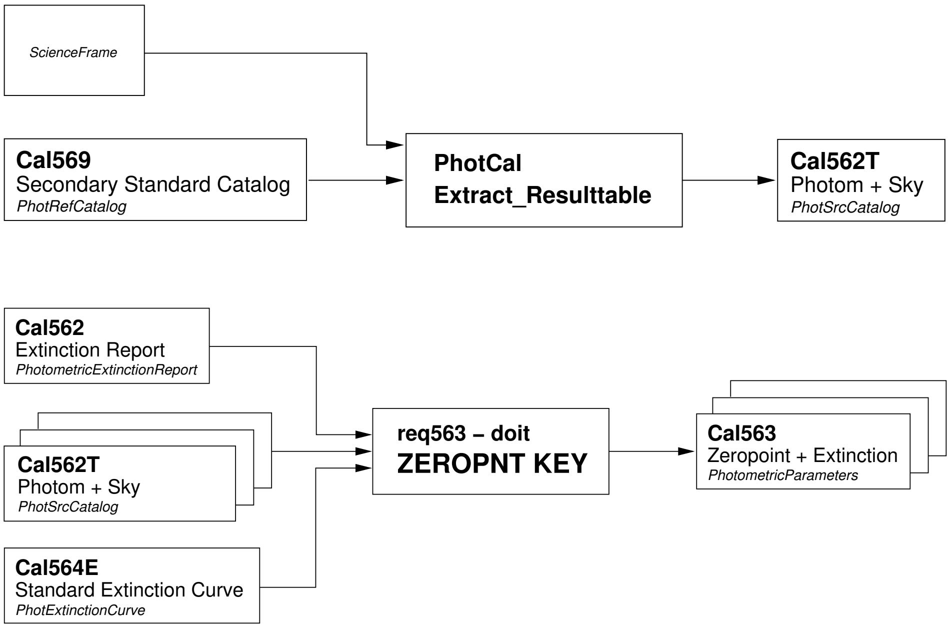


Fig 5.6.3 Dataflow and object class names for req563

The processing in the photometric part of the pipeline must be done for every **key band** separately and has two phases: 1) derive a photometry result table for every input chip, 2) use the photometry result tables in deriving the zeropoints for the night. The photometry result tables are derived in the same way as described in steps 1-8 of the relevant section of the **CA** from **req.** 562; these have to be derived first before the zeropoints for the night. The zeropoints for the night are derived from these photometric result tables as described in the steps 1-5 below. These steps are repeated for every separate input photometry result table.

1. Retrieve from the input extinction report (**CalFile– 562**) the atmospheric extinction for the middle of the night (see **req.** 562).
2. Multiply the retrieved extinction with the airmass contained in the input photometry result table.
3. Subtract the resultant value for the extinction from every raw zeropoint contained in the input photometry result table.
4. Derive the zeropoint from the resultant raw zeropoints by deriving the weighted average.
5. Store the derived zeropoint and the input extinction per unit airmass, together with the name of the chip for which the zeropoint was derived and the name of the key band.

CAP:

```
Constant:
  DETECT_THRESH  : Sextractor processing parameter (float). Value: 10.0.

/* Step 1 : make photometry result tables

(1)       create_empty_list_for_photometry_result_tables
(2)       for every frame in the list of science frames:
(2.1)       derive_source_catalog_from_frame:
              Sextractor.sextractor(frame_name)
(2.2)       remove_saturated_stars_from_the_catalog:
              LDAC.filter(catalog_name, filter_criterium_1)
(2.3)       associate_source_catalog_with_standard_star_catalog:
              LDAC.prephotom(catalog_name, refcat)
(2.4)       remove_not_associated_sources_from_catalog:
              LDAC.filter(catalog_name, filter_criterium_2)
(2.5)       create_empty_list_for_raw_zeropoints
(2.6)       for every source in the source catalog:
              mag = take_magnitude_of_source_from_standard_catalog
(2.6.1)         if composite_filter_data:
                  apply_transformation_to_mag
              instmag = take_measured_magnitude_of_source
              raw_zeropoint = mag - instmag
              raw_zeropoint_err = sqrt(mag_err**2 + instmag_err**2)
              add_raw_zeropoint_and_error_to_list
(2.7)       save_the_photometry_result_table
(2.8)       add_photometry_result_table_to_list

/* Step 2 : make extinction

cwl = central_wavelength_of_key_band
extinction  = value_from_extinction_curve_at_cwl
extinction += shift_from_input_report_at_middle_of_the_night

/* Step 3 : make zeropoints

(1)     for every table in list of photometry_result_tables:
(1.1)     ext = extinction * airmass_from_table
(1.2)     ext_err = extinction_error * airmass_from_table
(1.3)     create_empty_list_for_zeropoints
(1.4)     for every raw_zeropoint in table:
```

```
          zeropoint = raw_zeropoint + ext
          zeropoint_err = sqrt(raw_zeropoint_err**2 + ext_err**2)
          add_zeropoint_and_error_to_list
(1.5)   get_weighted_mean_and_error_of_zeropoint
(1.6)   save_zeropoint_and_extinction
```

QC Parameters:

TBD

5.6.4 Req. — *Cat I: Photometric Calibration - zeropoint user bands*

Objective:

Determine the zeropoint of the overall detector chain and the atmosphere by measuring standard stars in the **User passbands**.
The zeropoints of the photometric calibration in the User bands will only be determined for the nights that the User bands are actually used for scientific observations. The measurements will be done on one of the eight equatorial fields.
The atmospheric extinction will be determined in the keybands through **req. 562** *monitoring* on the polar field. The extinction results from **req. 562** *monitoring* are appended here, like in **req.**563, with a monolithic exposure at the equatorial field at midnight. The extinction curve will be used to transform the measured atmospheric extinction at the keybands to the User bands.
The transformation (colour term) of the user passband to the key passbands is determined for a limited set of filters and the ADC with the keybands in **req.**565.
By combining the extinction results in the keybands, the passband transformation coefficients and the direct zeropoint measurements in the User bands, the zeropoint corresponding to the DQE of each of the 32 CCD chips will be determined on an individual chip basis.
Trend analysis on these data is not required. The instrumental magnitudes of standard stars in each of the userbands will not be solved.

Fulfilling or fulfilled by:

Selfstanding, but uses data of **req.**562 at the middle of the night.

When performed/frequency:

Once in the middle of each night.

Sources, observations, instrument configurations:

All observations done with **Mode– Stare** N=1. **OmegaCAM** equatorial fields; always one exposure in composite key filter and additional exposures in User bands.

Required accuracy, constraints:

2% on the photometric scale for broad bands and 5% for narrow band filters.

Estimated time needed:

2-3 minutes per User band. Reduction: 3 min./CCD/filter.

TSF:

Mode– Stare N=1
(**TSF– OCM_img_obs_stare,** N=1, filter=userband)
= **TSF– OCM_img_obs_zpuser**

Inputs:

The inputs for deriving the photometry result tables are:
Fully calibrated image of equatorial standard field
CalFile– 569 *Secondary Standard Catalog*
CalFile– 565 *User -> key*

The inputs for deriving the zeropoints for the night are:
List of photometry result tables
CalFile– 562 *Extinction night report*
CalFile– 564E *Standard extinction curve*

Outputs:

The output from deriving a photometry result table is the same as for **req.** 562.

The outputs from deriving the zeropoints for the night are:
CalFile– 564 *Zeropoint - extinction - Userbands* (always)

Recipe:

The same recipes are used as for **req.**563. Only difference is the requirement of **CalFile– 565** *User -> key* as an extra input for deriving the photometry result tables.

CA:

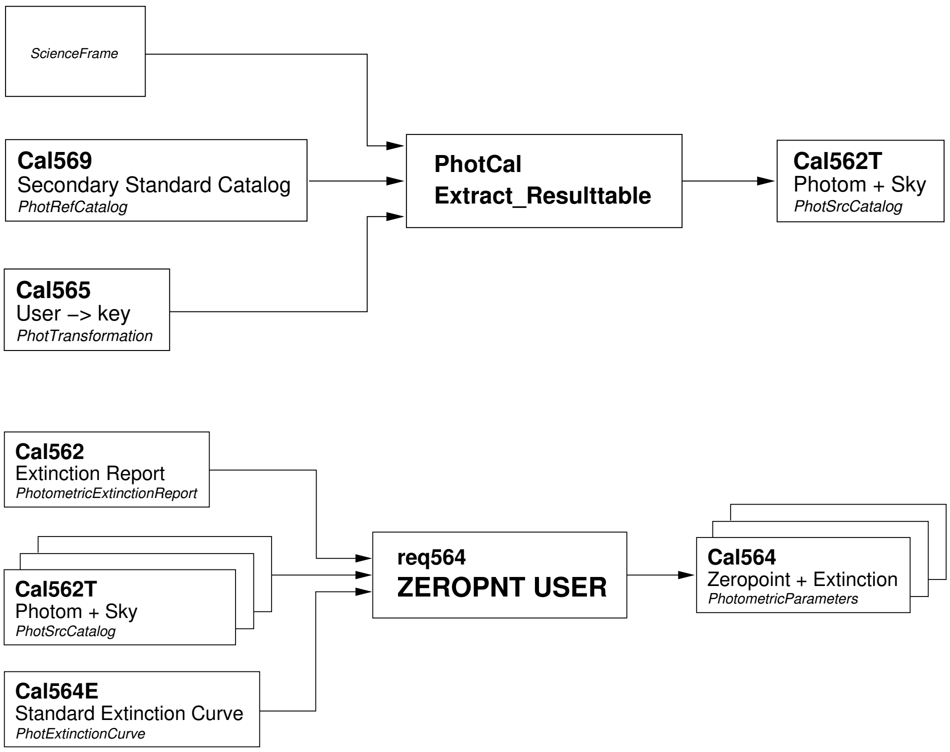


Fig 5.6.4 Dataflow and object class names for req564

The processing in the photometric part of the pipeline must be done for every **User band** separately and has two phases: 1) derive a photometry result table for every input chip, 2) calculate the extinction, and derive the zeropoint for every separate chip (i.e. from the information contained in every separate photometry result table). The photometry result tables are derived in the same way as described in steps 1-8 of the relevant section of the **CA** from **req.** 562; these have to be derived first before the zeropoints for the night. The zeropoints for the night are derived from these photometry result tables as described in the steps 1-6 below. These steps are repeated for every separate input photometry result table.

1. Retrieve from the input extinction report (**CalFile– 562**) the shift on the extinction curve as determined for the middle of the night (see **req.** 562).
2. Retrieve from the standard extinction curve the atmospheric extinction evaluated at the central wavelength of the User-band filter under consideration. Apply the shift as retrieved from the extinction report to this extinction. Note that this extinction is per unit airmass.
3. Multiply the extinction with the airmass from the photometry result table, and subtract the resultant value from the raw zeropoints contained in the photometry result table.
5. Derive the zeropoint from the extinction corrected raw zeropoints by deriving the weighted average.
6. Store the derived zeropoint and the input extinction per unit airmass, together with the name of the chip for which the zeropoint was derived and the name of the User-band.

CAP:

```
Constants:
  DETECT_THRESH  : Sextractor processing parameter

/* Step 1 : make photometry result tables

(1)    create_empty_list_for_photometry_result_tables
(2)    for every frame in the list of science frames:
(2.1)    derive_source_catalog_from_frame:
           Sextractor.sextractor(frame_name)
(2.2)    remove_saturated_stars_from_the_catalog:
           LDAC.filter(catalog_name, filter_criterium_1)
(2.3)    associate_source_catalog_with_standard_star_catalog:
           LDAC.prephotom(catalog_name, refcat)
(2.4)    remove_not_associated_sources_from_catalog:
           LDAC.filter(catalog_name, filter_criterium_2)
(2.5)    create_empty_list_for_raw_zeropoints
(2.6)    for every source in the source catalog:
           mag = take_magnitude_of_source_from_standard_catalog
           apply_transformation_to_mag
           instmag = take_measured_magnitude_of_source
           raw_zeropoint = mag - instmag
           raw_zeropoint_err = sqrt(mag_err**2 + instmag_err**2)
```

```
          add_raw_zeropoint_and_error_to_list
(2.7)   save_the_photometry_result_table
(2.8)   add_photometry_result_table_to_list

/* Step 2 : make extinction

cwl = central_wavelength_of_user_band
extinction  = value_from_extinction_curve_at_cwl
extinction += shift_from_input_report_at_middle_of_the_night

/* Step 3 : make zeropoints

(1)   for every table in list of photometry_result_tables:
(1.1)   ext = extinction * airmass_from_table
(1.2)   ext_err = extinction_error * airmass_from_table
(1.3)   create_empty_list_for_zeropoints
(1.4)   for every raw_zeropoint in catalog:
          zeropoint = raw_zeropoint + ext
          zeropoint_err = sqrt(raw_zeropoint_err**2 + ext_err**2)
          add_zeropoint_and_error_to_list
(1.5)   get_weighted_mean_and_error_of_zeropoint
(1.6)   save_zeropoint_and_extinction
```

QC Parameters:

TBD

5.6.5 Req. — *Cat I: Filter band passes - user bands vs key bands*

Objective:

Characterize the transformation coefficients, including the colour term for the **OmegaCAM** user passbands to the **OmegaCAM** key passband.
In addition, determine the expected small colour terms between the four bands of the *composite* filter and the *monolithic* filters for all the **key bands**. The monolithic filters for the key bands are produced by a different company than the composite filter. The colour terms of the transformation monolithic-composite filter are put in a Calfile which is used as input in the determination of the zeropoint of the night **req.**563.
The standard keybands are calibrated in **req.**563 with the two lens corrector; the characterization of the ADC at the keybands and its transformation to the standard configuration is part of the present requirement.

Fulfilling or fulfilled by:

Selfstanding.

When performed/frequency:

Once commissioning for user bands, once/year for composite filter

Sources, observations, instrument configurations:

OmegaCAM equatorial fields; **Mode– Stare** N=1; ADC for key passbands.

Required accuracy, constraints:

10% on the photometric scale (formal spec) and 2% (goal) for broad band filters
1% for composite -> key

Estimated time needed:

20 min/run

Inputs:

Reference magnitudes and transformations:
CalFile– 569 *Secondary Standard Catalog*
CalFile– 564E *Standard extinction curve*

each night:
CalFile– 562 *Extinction night report*

Outputs:

CalFile– 565 *User -> key - monolithic*
CalFile– 565C *composite -> monolithic*

5.6.6 Req. — *Cat III: Dependency on angle - ADC, rotator/reproducibility*

Objective:

Verify the dependency of the photometric calibration on the angle of the field rotator. Do this for both optical configurations of the telescope, ADC in and out.
Measure dome flatfield at 12 field rotator angles.
Measure the polar field with 12 field rotator angles.
This also verifies the reproducibility and provides an end-to-end test.

Fulfilling or fulfilled by:

Selfstanding

When performed/frequency:

Commissioning

Sources, observations, instrument configurations:

Flat field on dome with the 4 key passbands.

Required accuracy, constraints:

1% on the photometric scale

Estimated time needed:

Dome: day time, one day
polar field 6 hours.

TSF:

Mode– Stare N=12
TSF– OCM_img_cal_domeflat
and
Mode– Stare N=12
TSF– OCM_img_obs_stare

Outputs:

Flat fields - internal

CA:

Standard image pipeline with source extraction of polar field. Compare the source flux for the various rotator angles in an interactive analysis.

5.6.7 Req. — *Cat III: Linearity (as a function of flux)*

Objective:

Verify the linearity (ratio of input over output) of the overall detector amplification- data reduction chain for the three different observing modes as an end-to-end test.
Compare the resultant magnitudes derived by the image pipeline by taking short and long exposures of the same standard field.

Fulfilling or fulfilled by:

Selfstanding

When performed/frequency:

Commissioning

Sources, observations, instrument configurations:

Equatorial field in all the four key passbands; short 10 sec, 100 sec, 400 sec and 800 sec exposure. Two lens corrector.

Required accuracy, constraints:

Better than 1% on the photometric scale

Estimated time needed:

Commissioning 6 hour

TSF:

Mode– Stare N=2
TSF– OCM_img_obs_stare
and
Mode– jitter N=5
TSF– OCM_img_obs_jitter
and
Mode– dither N=5
TSF– OCM_img_obs_dither

Outputs:

OK flag

CA:

Run standard image pipeline and do source extraction on the calibrated image. Compare magnitudes of source lists for different exposure times in an interactive analysis.

5.6.8 Req. — *Cat III: Detection limit and ETC calibrations*

Objective:
Verify the effective detection limit/ overall througput of VST + camera and subsequently the parameter values used for the Exposure Time Calculator.

Fulfilling or fulfilled by:
Data analysis of **req. 563** *Zeropoint*, **req. 567** *Linearity end-to-end-test*

When performed/frequency:
Commissioning

Required accuracy, constraints:
10 % in detection limit

Outputs:
Values for ETC

CA:
Interactive analysis

5.6.9 Req. — *Cat I: Secondary Standards*

Objective:

Build catalogs of secondary photometric standards by observing Landolt fields, centered on each individual CCD.

A fundamental concept in the calibration of **OmegaCAM** data is the photometric calibration for each separate CCD. To obtain this calibration with a single observation requires photometric standards covering the entire FoV of the instrument. There are currently no catalogs of photometric standard stars satisfying this requirement. Hence, obtaining such catalogs of **secondary standards** for the equatorial fields and the polar field will be part of the calibration observations to be performed during operations.

Obtaining the necessary observations of secondary standards is a time-consuming operation (see below). Moreover, because these observations should cover all 8 equatorial fields (ref **seq.– 563),**approximately two (bright) nights each month will have to be reserved for these observations, at least in the first year of operations.

The set of secondary standards observations will also be used to determine the illumination correction (**req.**548). Note, that this constitutes a bootstrap problem, because determining accurate zeropoints of the secondary standards requires that the illumination correction is already known.

Because of this bootstrap problem, a straightforward determination of the zeropoint requires that for each CCD the zeropoint and illumination correction are determined simultaneously using primary standards. Therefore, obtaining measurements of secondary standards requires a sequence of observations that positions the primary standards in each CCD, i.e. 32 pointings per field, per filter.

An initial determination of the illumination correction will serve as a check on the magnitude of the effects of stray light. To do this in a quick and efficient way, a preliminary catalog of standard stars that cover the FoV of **OmegaCAM** is required. The data for constructing such a catalog (**CalFile– 569E**) has been obtained. In case the effects of stray light turn out not to be negligible, the illumination correction will have to be characterized (see **req.** 548).

The order of priority of determining standards is first the **key bands** followed by the composite filter.

Fulfilling or fulfilled by:

Selfstanding.
Initial determination of the illumination correction is used to verify presence/absence of this effect (see **req.**548).

When performed/frequency:

Commissioning, 2 nights of bright time, each month in the first year.

Sources, observations, instrument configurations:

Strategy– freq Mode– Stare N=1 **OmegaCAM** equatorial fields and polar field.

Required accuracy, constraints:

0.02 mag in individual secondary standard stars. The sources in **CalFile– 569** cover a FoV of approximately 1.2 square degree for every standard field. The density of the standard stars in a given field is such that the number of standard stars per chip is typically a few hundred.

Estimated time needed:

Two standard fields are observed each month, with an observing time of 5.8h each. Each standard field is observed with 32 pointings, in each of the **key bands**. Using an observing strategy in which the filter is changed after every set of 32 pointings, the total observing time T_{obs} for one field is estimated by:

$$T_{obs} = \sum_{i=1}^{N_{filter}} [(T_{int}(X_i) + T_{read} + T_{mov}) \times 32 + T_{filter}],$$

with $T_{int}(X_i)$ the integration time in band X_i, T_{read} the readout time of the full detector block, T_{mov} the movement time of the telescope, and T_{filter} the filter exchange time.
With a T_{read}=25s, a T_{mov}=40s, and a T_{filter}=75s, the total integration time for the four **key bands** and one field will be ≈ 21000s (5.8h). The integration times for each of the four **key bands** are: $T_{int}(u')$=110s, $T_{int}(g')$=60s, $T_{int}(r')$=75s, $T_{int}(i')$=115s. The reduction time of the data taken for one field is 2 hours per CCD/filter.

TSF:

Strategy– mosaic
Mode– Stare N=1
TSF– OCM_img_obs_stare

Inputs:

CalFile– 569E *Primary Standard stars*

Outputs:

CalFile– 569 *Secondary Standard Catalog*
CalFile– 548 *Illumination Correction*

CA:

Process those chips from the total set of standard field data that contain the Landolt stars. The following steps are performed on every separate chip in this subset (step 1):
1. De-bias, flatfield and astrometrically calibrate the input chip.
2. Derive a source catalog from the input chip (Sextractor).
3. Remove saturated sources from the source catalog.
4. Associate the source catalog with the standard star catalog.
5. Remove all the sources from the source catalog that have not been associated with a source from the standard star catalog.
6. Subtract the measured magnitude of the identified standard stars from their magnitude as known from the standard star catalog. The resultant numbers will further be referred to

as raw zeropoints.
7. Derive a zeropoint for the input chip from the list of raw zeropoints by deriving the weighted average, and store this zeropoint.

This procedure will result in 32 values for the zeropoint, one for every chip. The remaining standard field data that does not contain Landolt start must only be de-biased, flatfielded, and astrometrically calibrated.

Obtain catalogs (step 2). The following steps are performed on every separate input chip of the previously de-biased, flatfielded and astrometrically calibrated standard field data :
1. Derive a source catalog from the input chip (Sextractor).
2. Remove the sources from the catalog with an error in the magnitude equal to or larger than 0.03 magnitudes.
3. For the remaining sources, subtract their measured magnitudes from the zeropoint of the input chip as it has been derived in step 1.
4. Store in a catalog : the zeropoint corrected magnitudes of the sources together with their positional information.

Obtain secondary standards (step 3):
1. Associate the source catalogs derived in processing step 2.
2. Select from the association result those sources which have been observed at least 7 times. For every source in the resulting list, average the magnitude.
3. Select the sources with an averaged magnitude accurate within 0.02 mags.

Note that the procedures above only describe the most straightforward way of deriving a standard star catalog. In a more refined procedure with relative photometry, the non-standard stars can also be used. This will lead to more accurate results.

5.7 Internal alignments, optics etc

5.7.1 Req. — *Cat III: Camera focus/tilt*

Objective:

Determine and verify the camera focus.
In a series of exposures both M2 will be moved and charge will be moved over a single chip, so that only one read-out is nessecary (TS 3.3.12).
Verify once for each filter that they have the same optical thickness (15mm physical thickness). Do this by measuring the "filter focus offset".
The tilt of the detector plane with respect to the focal plane and its dependency on the orientation of the telescope shall be determined both from:
CalFile– 554 *PSF anisotropy*
and from the matrix of best focus positions provided by the present requirement.

Fulfilling or fulfilled by:

Selfstanding

When performed/frequency:

verify focus: CP and RP. Filter thickness and tilt only once during comissioning.

Estimated time needed:

Observation:
- Focus offset and tilt: 2 hours during CP
- Verification filter focus offset 10-30 min/filter (Commissioning)
Reduction: 1 min/CCD.

TSF:

TSF– OCM_img_tec_focusseqwhich stores data for one chip. For the special occasion of the tilt determination data of all chips will be stored (a trivial mod).

Outputs:

for filter offsets:
-focus offset values to be transferred to INS data base
-Conformance flag for optical thickness of all filters
For tilt during CP:
-Tilt value

Recipe:

Recipe– Focus Standard Paranal focus MIDAS procedure

CA:

Verification of focus:
this is an on-line activity which is performed using the RTD and the standard Paranal MIDAS focus procedure tplTFocus.prg. This relatively simple MIDAS procedure asks the user to point on the screen at a non-saturated, well exposed star. It will then compute for a sequence of stellar images the optimally focussed images and returns in keywords outputr(1) = sigmax outputr(2) = sigmay outputr(3) = focus. Plots are made and tables are created for further interactive analysis (if wanted).

Standard RTD allows to make cross cut plots and to "pick" an object interactively for which position, FWHM in X, Y, and other parameters are computed. It can be used to inspect any CCD, e.g., by reading in RTD the complete file that is created after merging of the data from both mosaic halves.

The MIDAS procedure forms the core of the data reduction of all focus determinations. In "stand alone" mode it provides the optimal focus values, and the values for focus offsets for other filters.

For the tilt determination, which is to be done only once during commissioning, tplTFocus.prg will be put in a loop in a supervising MIDAS command procedure. While running this superviser loop, well exposed stars will be selected over a large number of CCDs (if not all) and this output (best focus value) will be merged into one big MIDAS table. Table will be converted into an image and a big x-y contour plot of best focus values will directly show the tilt. An regression analysis on the table will result in the finally adopted orientation angle and amplitude. Since this procedure is run only once, it will not be automized but entirely interactively processed with the help of a handful of simple MIDAS commands.

CAP:

```
! "@(#) $Id: req571.tex,v 1.29 2004/06/03 21:11:14 valentyn Exp $"
!# -*- tcl -*-
!#************************************************************************
!# E.S.O. - VLT project
!#
!# tplTFocus.prg
!#
!# who        when      what
!# ---------- --------  ---------------------------------------------
!# alongino   25/10/00  renamed from TC
!#
!#************************************************************************
!#   NAME
!#   tplTFocus
!#
!#   SYNOPSIS
!#
!#
!#   DESCRIPTION
!#
!#
!#   FILES
!#
!#   ENVIRONMENT
!#
```

```
!#   RETURN VALUES
!#
!#   CAUTIONS
!#
!#   EXAMPLES
!#
!#   SEE ALSO
!#
!#   BUGS
!#
!#--------------------------------------------------------------------------
!#
!.KEYWORDS
! ------------------------------------------------------
!
! PARAMETERS:
!    P1: start-x (default = 0.)
!    P2: step-x  (default = 1.)
!
!define/parameter p1 0. N "Enter start-x : "
!define/parameter p2 1. N "Enter step-x : "
!
define/local centin/c/1/6 CURSOR
!
! initialization
define/local focx/d/1/1 0.0
define/local focy/d/1/1 0.0
define/local sigmax/d/1/1 0.0
define/local sigmay/d/1/1 0.0
define/local focus/d/1/1 0.0
define/local aux/d/1/1 0.0
define/local fox/d/1/1 0.0
define/local sign/d/1/1 0.0
create/tab tfocus 15 0 null
!
! setup the graphics window
create/graph
set/graphics default
set/graphics pmode=1
!
! getting the coordinates...
```

```
center/gauss {centin} tfocus
if outputi(1) .eq. 0 then
    write/out
    write/out Invalid number of coordinates.
    return
endif
!
set/midas output=no ! we don't want output on the screen...
!
!
! fit-x computation
compute/table tfocus :step = SEQ
regression/polynomial tfocus :xfwhm :step 2
save/regression tfocus test
comp/regression tfocus :fitx = test(:step)
focx = {p1} + {p2}*({outputd(2)}/(-2.*{outputd(3)})) - {p2}
!
! fit-y computation
regression/polynomial tfocus :yfwhm :step 2
save/regression tfocus test
comp/regression tfocus :fity = test(:step)
focy = {p1} + {p2}*({outputd(2)}/(-2.*{outputd(3)})) - {p2}
!
! compute the scaling for plots (big job ;{ )
!
compute/table tfocus :maxim = MAX(:xfwhm,:yfwhm)
sort/table tfocus :maxim(-)
define/local maxim/r/1/1 {tfocus,:maxim,1}
maxim = {maxim} + 0.2  ! let's give a margin value
compute/table tfocus :minim = MIN(:xfwhm,:yfwhm)
sort/table tfocus :minim
define/local minim/r/1/1 {tfocus,:minim,1}
minim = {minim} - 0.2  ! let's give a margin value
set/graphics yaxis={minim},{maxim}
!
! plot (x,y)-value versus (x,y)-sigma
sort/table tfocus :step
set/graphics stype=6 colour=1 ! cross(x), black
plot/table tfocus :step :xfwhm
set/graphics stype=5 colour=4 ! cross(+), blue
overplot/table tfocus :step :yfwhm
```

```
!
! plot of the fit
set/graphics stype=0 ltype=1 ! no symbol, solid line
set/graphics colour=1 ! black
overplot/table tfocus :step :fitx
set/graphics colour=4 ! blue
overplot/table tfocus :step :fity
!
! focus computation
aux  = {focx} * {focy}
aux  = M$ABS({aux})
fox  = M$ABS({focx})
sign = {focx}/{fox}
focus = {sign} * M$SQRT({aux})
!
! residual computation
compute/table tfocus :residx2 = (:xfwhm - :fitx) ** 2
compute/table tfocus :residy2 = (:yfwhm - :fity) ** 2
statis/table tfocus :fitx
sigmax = outputr(1)
statis/table tfocus :fity
sigmay = outputr(1)
!
set/midas output=yes ! enable the screen output...
write/out
write/out
if mid$sess(16:20) .eq. "94NOV" then
   set/format f12.2
else
   set/format ,f12.2 !for double keywords
endif
write/out "X: BLACK    y: BLUE"
write/out
write/out "FOCUS:  X =  {focx} " "Y =  {focy}  " "MEAN =  {focus}"
write/out
if mid$sess(16:20) .eq. "94NOV" then
   set/format f12.3
else
   set/format ,f12.3 !for double keywords
endif
write/out "sigma-x = {sigmax}   sigma-y = {sigmay}"
```

```
set/graphics default

delete/tab tfocus no

!
outputr(1) = sigmax
outputr(2) = sigmay
outputr(3) = focus

return
```

5.7.2 Req. — *Cat III: Ghosts - ADC*

Objective:

Verify the absence/presence of ghosts.
For each available filter inspect the presence of ghosts by making several exposures near a very bright star at various angular distances from the field center. Do this for both correctors. Bright stars should also be positioned on the edges of the baffles, filters and the dewar entrance window and on the dividing lines between segments of composite filters
Various studies on expected behavour of ghosts are posted on **OmegaCAM**website. Bottom line of these studies is that ghosts are expected to be very faint and very large, pre-cluding their suppresion by dithered observations and subsequent masking.
The inspection will be done in first instance visually on the RTD.
Off-line the images will be fed to the standard pipeline for closer inspections.

Fulfilling or fulfilled by:

Selfstanding

When performed/frequency:

CP

Sources, observations, instrument configurations:

Very bright star at several angular distances from the field center (0.7, 1, 2, 3 degree, to be determined by experience). All available filters, or when a new filter is installed.
To be done for both correctors, to be determined by experience.

Required accuracy, constraints:

Estimated time needed:

1 hour/filter

TSF:

Mode– Stare N=1
TSF– OCM_img_obs_stare

CA:

Standard image pipeline, visual inspection of images on RTD

5.8 Effect of Telescope

The various effects of the telescope on the quality of the images produced by the camera have been addressed by the following **req.**'s:

req. 524 *Electromagnetic compatibility*
req. 551 *Position of camera in focal plane*
req. 552 *Telescope pointing*
req. 553 *Telescope and rotator tracking*
req. 554 *PSF anisotropy*
req. 566 *Dependency on rotation angle - field rotator, ADC*
req. 572 *Ghosts*

This list appears rather complete and no additional requirements are specified.

5.9 Workhorses and End to end tests

WORKHORSES

The following ‘work horses’ or ‘doit’ requirements are specified:

req. 521 *CCD read noise - doit*
req. 531 *CCD Dark Current - doit*
req. 541 *Bias - doit*
req. 547 *Quick detector responsivity check - doit*
req. 555 *The astrometric solution for templates - doit*
req. 563 *Photometric Calibration - zeropoint keybands - doit*

The following **end-to-end tests** (i.e. observational data which employ many different aspects of the system and which can be used to trace reproducibility) have been specified:

req. 566 *Dependency on rotation angle - ADC, rotator/ reproducibility*
req. 567 *Linearity (as a function of flux)*
req. 562 *Photometric calibration - Monitoring/ Health check*

5.10 On site quick look analysis

Requirements for the Real Time Display, essentially requirements on how to perform visual health checks on the acquired data, are given in the Instrument S/W User requirement document.

Here **req.**'s are listed which require analysis on the site.

The first list gives the requirements for which on-site analysis is essential (listed in order of priority).

req. 562 **Photometric calibration- Monitoring/ Skyspectrum*
req. 547 **Quick detector responsivity check - doit*
req. 521 **CCD read noise - doit*
req. 531 **CCD Dark Current - doit*

The second list gives the requirements for which on-site analysis is desirable/most practical (listed in order of priority). On-site, these activities will output go/no-go flags.

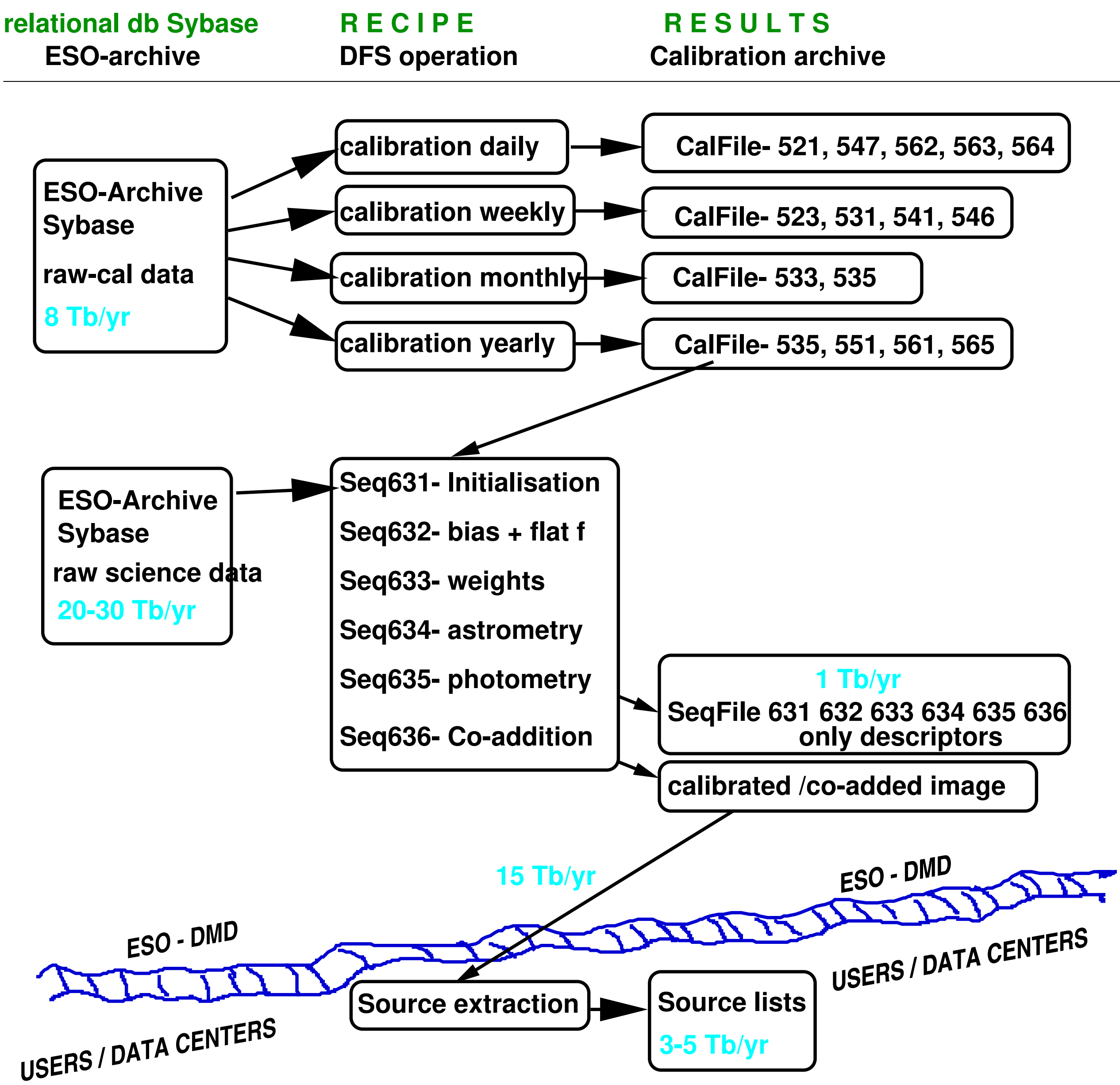
req. 571 *Camera focus/tilt*
req. 552 *Telescope pointing*
req. 553 *Telescope and rotator tracking*
req. 566 *Dependency on rotation angle - field rotator, ADC*
req. 524 *Electromagnetic compatibility*
req. 551 **Position of camera in focal plane*
req. 554 **PSF anisotropy*
req. 572 *Ghosts*

In the lists above the **req.**'s which produce a **CalFile–** are marked with a *****. These **req.**'s have also to be processed off-line at ESO HQ (e.g. DFS pipeline, DFS operations, calibration pipeline, QC1, Quality control, trendanalysis).

DFS-pipeline modules, extracted from the off-line ESO HQ version, (including thosee used for calibration pipeline and QC1) could fulfill these task on-site with relative limited extra effort. Such modules will run with a stripped Calfile-date base As a desirable side effect, this creates the possibility to also quickly verify any other **req.**'s with extracted DFS-pipeline modules in the case of un-expected events.

The filling of the calibration database (**CalFile–)**should however be exclusively handled by the off-line pipeline at HQ.

Fig 6. Data Flow of Science and Calibration data



6 DATA REDUCTION SOFTWARE SPECIFICS

6.1 Data reduction software requirements- see DRS and URD v1.2

6.2 Estimate of data volumes - see DRS and URD v1.2 at FDR

6.3 The Image Pipeline

As noted previously, it is useful to distinguish between the **calibration part of the Pipeline** and the **image part of the Pipeline**. Here we provide more detail on the operation of the image part of the Pipeline

The image part of the pipeline applies calfiles to raw science data, to produce astrometrically and photometrically calibrated science data. The Pipeline applies different *recipes* to raw science data, based on the observing template used. There are currently three observing templates for science data envisioned for **OmegaCAM**. These templates (see also Section 4 CP) process data by executing a number of recipes

- **TSF– OCM_img_obs_stare**: **Recipe– Split**, **Recipe– Reduce**
- **TSF– OCM_img_obs_jitter**: **Recipe– Split**, **Recipe– Reduce**, **Recipe– Coadd**
- **TSF– OCM_img_obs_dither**: **Recipe– Split**, **Recipe– Reduce**, **Recipe– Coadd**

The recipes (Split, Reduce, Coadd) combine a series of processing steps that are specified by their (intermediate) data product in **seq.– 631**-**seq.– 636**. These can be summarized by:

- **Initialization and image statistics** Split the multi-extension FITS image and describe (image statistics) each individual CCD. (**seq.– 631**, **Recipe– Split**).
- **de-biasing and flatfielding** This step removes the gain-variation within images (**seq.– 632**, **Recipe– Reduce**).
- **Constructing weights**, incorporating the detection of cosmic rays, and satellite tracks (**seq.– 633**, **Recipe– Reduce**).
- **Astrometry**, measuring the astrometric distortions and offsets (**seq.– 634**, **Recipe– Reduce**).
- **Apply photometry** (**seq.– 635**, **Recipe– Reduce**)
- **Dedithering**, combining the images to construct a clean, maximally uniform, image of the FOV of **OmegaCAM** and perform a quality check (**seq.– 636**, **Recipe– Coadd**).

The detailed specification of each **seq** is analogous to the specification of the **req**'s in Section 5. It should, however, be understood that within the context of the image part of the Pipeline, the recipes for the individual **seq**'s are combined into three recipes; the **Recipe– Split** covers **seq.– 631**, the **Recipe– Reduce** incorporates **seq.– 632**–**seq.– 635** and the regridding part of **seq.– 636**, **Recipe– Coadd** does the co-addition part of **seq.– 636**. The recipe for processing the data for each Observing Template is composed of these three recipes. An overview of the data flow is given in Fig. 6.3.

For **TSF– OCM_img_obs_dither** observations **Recipe– Coadd** is used to produce one co-added image by combining data from all CCD's. For **TSF– OCM_img_obs_jither** observations **Recipe– Coadd** can be applied per CCD to create a co-added image for each CCD.

Operations **seq.– 632**-**seq.– 635** and the regridding of **seq.– 636** are performed on individual chips. The format of the raw data will be one FITS file per observation including the data for 32 chips in 32 extensions. **seq.– 631** separates these raw data in FITS files for individual chips. Only in **seq.– 636** the data from individual chips are combined.

Note that **seq.– 631**-**seq.– 636** can also be used within various recipes for the calibration pipeline.

In **seq.– 632**-**seq.– 635** (**Recipe– Reduce**), the image pipeline will produce transient calibration data that is *specific* for individual CCDs. These data are recorded in **SeqFile–** 's, e.g. **SeqFile– 633** *individual weight map*. Most of this transient calibration data is uninteresting or could be easily re-constructed, while other data have persistent scientific value (such as seeing, PSF, astrometric solution, zeropoint, sky-brightness, etc.), while other data have potential, but unspecified, value for trend analysis or trouble-shooting. Although the data recorded in a **SeqFile–** may be discarded when the data is no longer needed, it is advisable to store the calibrated data for individual CCDs (**seq.– 635**) and the associated weight (**seq.– 633**) as these contain all potentially interesting (meta-)data for each CCD.

A complete overview of all the data products (expressed as pipeline objects) produced in the image pipeline is given in Fig. 6.4. The aim of this figure is to illustrate the various dependencies between various data products.

Both the image part of the Pipeline and the calibration part of the Pipeline obey the accuracies and errors as listed in Section 5. Propagation of errors shall be computed, where applicable.

The command-line usage for **Recipe– Split** is given in **seq.– 631**, for **Recipe– Reduce** in **seq.– 632** and for **Recipe– Coadd** in **seq.– 636**.

Dataflow for the image part of the Pipeline

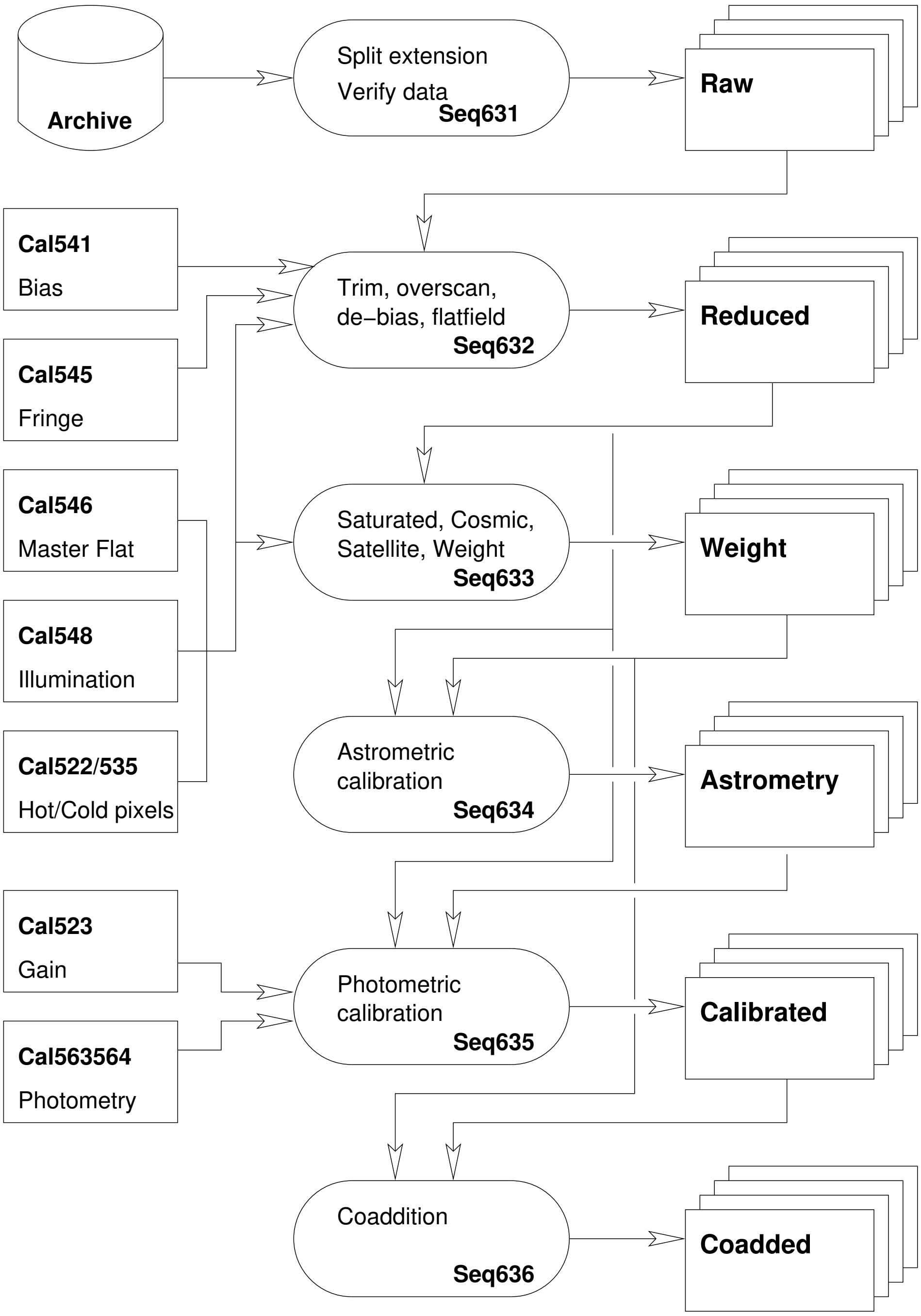


figure 6.3.

Data dependencies for the image part of the Pipeline

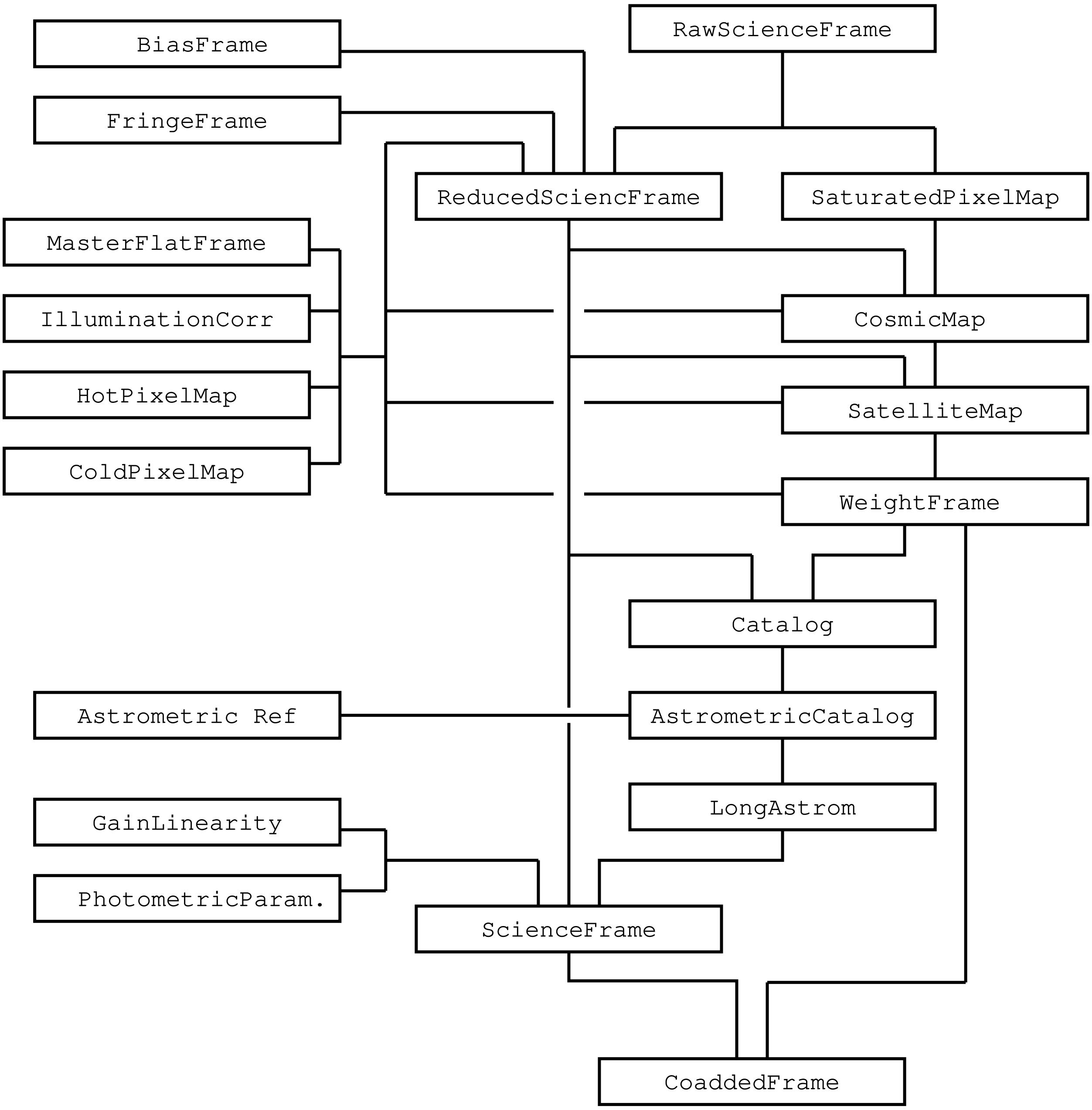


figure 6.4.

6.3.1 Seq. — *Cat I: Ingestion and image statistics*

Objective:

Prepare the data for pipeline processing and provide image statistics and qualifications.

The image part of the Pipeline processing is based on the processing of individual chips. The image data arrives from the telescope in the form of FITS files with the data for each chip stored into a single FITS extension. For the purpose of efficient parallel processing this **seq.–** splits the FITS files (and their extensions) into individual FITS files for individual chips. This functionality is provided generically by the pipeline environment, and is assumed to be used prior to any data processing (including all calibration data processing)

It is assumed that incoming raw data are valid FITS files. Further qualification of the raw data by means of image statistics should determine whether the data can be meaningfully processed by the pipeline.

Separate statistics of pre- and overscan regions are recorded for trend-analysis of detector stability.

Estimated time needed:

5 sec/CCD

Inputs:

Raw science frames.

Outputs:

SeqFile– 631 *Raw science frame*

Recipe:

```
Split -i <raw_unsplit_images> -t image_type [-n number_of_extensions]

raw_unsplit_images   : raw multi-extension FITS data
image_type           : Type of the supplied raw unsplit images (string)
                       Description of allowed values:
                       bias     - Bias image - see req.541
                       dark     - Dark current image - see req.531
                       dome     - Domeflat image - see req.542
                       science  - Science image - this req.631
                       twilight - Twilightflat image - see req.543
number_of_extensions : Number of extensions (i.e. ccd's) to be found in the
                       unsplit data (integer)
                       Range of allowed values: 1 - 64. Default: 32
```

CA:

Process (make):

1. Extract header from given extension and merge with primary header
2. Extract image data from given extension.
3. Determine image statistics in the trim, pre- and overscan regions

4. Write out image with merged header.

CAP:

```
main_header = eclipse.header(raw_unsplit_image)
extension_header = eclipse.header(raw_unsplit_image, extension)
header = main_header.merge(extension_header)

# NAXIS1, NAXIS2, PRSCX, PRSCY, OVSCX, OVSCY from header
# Assumes read-out from left-hand side
trim       = [PRSCX+1, PRSCY+1, NAXIS1-OVSCX, NAXIS2-OVSCY]
prescan_x  = [1, PRSCY+1, PRSCX, NAXIS2-OVSCY]
overscan_x = [NAXIS1-OVSCX+1, PRSCY+1, NAXIS1, NAXIS2-OVSCY]
overscan_y = [PRSCX+1, NAXIS2-OVSCY+1, NAXIS1-OVSCX, NAXIS2]

image           = eclipse.image(raw_unsplit_image, extension)
stat_trim       = eclipse.stat(image, region=trim)
stat_prescan_X  = eclipse.stat(image, region=prescan_X)
stat_overscan_X = eclipse.stat(image, region=overscan_X)
stat_prescan_Y  = eclipse.stat(image, region=prescan_X)

eclipse.image_save(image, header)
```

QC Parameters:

```
stat_trim       -- Image statistics - trim region
stat_prescan_X  -- Image statistics - prescan X  (right)
stat_overscan_X -- Image statistics - overscan X (left)
stat_overscan_Y -- Image statistics - overscan Y (top)
```

6.3.2 Seq. — *Cat I: Trim, de-bias, flatfield*

Objective:

Trim, de-bias, and flatfield the raw data.

This second step of the image part of the Pipeline applies a number of calibration files to a raw science image after trimming and overscan correcting the image. This includes bias subtraction and flat-fielding, as well as fringe removal.

Several methods for overscan correction are possible, provided the calibration files have been derived using the same method (see Section 7.1 for further details).

Fringe removal is done using an automatic fringe scaling algorithm. This method assumes the standard deviation of the pixel values in the reduced science image is a function of a fringe map scale factor. Pixel statistics can then be used to calculate a minimum for this function. Masks and sigma thresholding are used to exclude bad pixels, stars and focus on pixels in fringe minima and maxima.

Due to problems with non-uniform illumination, as described in **req.**548, the zeropoint is not uniform over the image. Therefore an illumination correction must be applied. Because this corrects for a non-uniform illumination, the background would not be flat anymore after this correction. Therefore, the background is subtracted prior to multiplying the data with the illumination correction image.

Estimated time needed:

Approximately 30 seconds/CCD

Inputs:

SeqFile– 631 *Raw science frame*
CalFile– 541 *Master Bias frame*
CalFile– 546 *Master flatfield*
CalFile– 545 *ff-fringe*
CalFile– 548 *Illumination correction*

Outputs:

SeqFile– 632 *Reduced science frame*
Updated image statistics (before and after background removal)

Recipe:

Note that the recipe below performs step **seq.– 632** through **seq.– 635** and the regridding part of **seq.– 636.**The target grid, defined by the input parameters ra, dec and pixelscale, has to be the same for all images that are to be coadded later by **Recipe– Coadd.**

```
Reduce -i raw -b bias -f flat -p photom -g GAIN -ra RA -dec DEC
      -ps PIXELSCALE [-fr fringe] [-c cold] [-h hot]
      [-il illumination] [-sl THRESHOLD_LOW]
      [-sh THRESHOLD_HIGH] [-oc OVERSCAN_CORRECTION]
      [-ft IMAGE_THRESHOLD] [-fl FRINGE_THRESHOLD_LOW]
      [-fh FRINGE_THRESHOLD_HIGH] [-ct DETECTION_THRESHOLD]
```

```
    [-sit DETECTION_THRESHOLD] [-sht HOUGH_THRESHOLD]
    [-dt ASTROM_DETECTION_THRESHOLD]

raw          : raw science frame
bias         : master bias frame
flat         : master flat frame
photom       : photometric parameters
GAIN         : gain (float)
               Range of allowed values: 0.1 - 10.0.
RA           : right ascension of the center of the output image
               (decimal degrees, float)
               Range of allowed vales: 0.0 - 360.0
DEC          : declination of the center of the output image
               (decimal degrees, float)
               Range of allowed values: -90.0 - 90.0
PIXELSCALE   : pixel scale of output image (arcsec, float)
               Range of allowed values: 0.05 - 1.0
fringe       : fringe map
cold         : cold pixel map
hot          : hot pixel map
illumination : illumination correction frame
THRESHOLD_LOW         : saturation cutoff low (float)
                        Range of allowed values: 0.0 - 200.0. Default 50.0.
THRESHOLD_HIGH        : saturation cutoff high (float)
                        Range of allowed values: 40000.0 - 70000.0.
                        Default 50000.0.
OVERSCAN_CORRECTION   : overscan correction mode (integer).
                        Description of allowed values:
                          0: apply no overscan correction (default)
                          1: use median of the prescan in the
                             x-direction
                          2: use median of the overscan in the
                             x-direction
                          3: use median of the prescan in the
                             y-direction
                          4: use median of the overscan in the
                             y-direction
                          5: use the per-row value of the prescan in
                             the x-direction
                          6: use the per-row value of the overscan in
                             the x-direction
```

```
IMAGE_THRESHOLD         : (for fringe map scaling) do not use pixels from
                          science image outside this threshold (float)
                          Range of allowed values: 2.0 - 10.0. Default: 5.0
FRINGE_THRESHOLD_LOW    : do not use pixels from fringe map below this
                          threshold (float)
                          Range of allowed values: 1.0 - 3.0. Default: 1.5.
FRINGE_THRESHOLD_HIGH   : do not use pixels from fringe map above this
                          threshold (float)
                          Range of allowed values: 3.0 - 10.0. Default: 5.0.
DETECTION_THRESHOLD     : Sextractor detection threshold used for cosmic ray
                          detection (float)
                          Range of allowed values: 2.0 - 20.0. Default: 5.0.
DETECTION_THRESHOLD(2)   : Sextractor detection threshold used for satellite
                           track detection (float)
                           Range of allowed values:  Default: 5.0.
HOUGH_THRESHOLD          : Theshold for satellite tracks in Hough image
                           (float)
                           Range of allowed values: TBD. Default 1000.0.
ASTROM_DETECTION_THRESHOLD : Sextractor detection threshold used for
                             extracting astrometric catalog (float)
                             Range of allowed values: 5.0 - 15.0.
                             Default: 10.0.
```

CA:

1. Perform an overscan correction and trim the science image. 2. Subtract the master bias.
3. Divide by the master flatfield.
4. Check for filters that may require fringe subtraction. Scale and subtract a fringemap if so. The scaling is done 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 assume $\frac{d\sigma}{da} = 0$ 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 $\times$ j, which is not equal to NAXIS1 $\times$ NAXIS2 if there are any masked pixels), and μ_I and μ_F 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.
5. Compute image statistics.

6. Subtract the background.
7. Multiply the image with the illumination correction frame.
8. Compute the image statistics again.

CAP:

```
Constants:
  BACKGROUND_CONFIG : SExtractor configuration parameters and their values
                      in case they differ from the SExtractor defaults.
          CHECKIMAGE_TYPE = 'BACKGROUND'
          DETECT_THRESH   = 1000
          ANALYSIS_THRESH = 3.0
          BACKPHOTO_THICK = 24
          BACK_TYPE       = 'AUTO'
          BACK_VALUE      = 0.0, 0.0
          FILTER          = 'N'
          GAIN            = 1
          MEMORY_PIXSTACK = 100000

ima = eclipse.trim_and_overscan(raw)
ima = (ima-bias)/flat

if filter.has_fringes():
  FUNCTION make_mask():
    mask = ~(hotpixelmap & coldpixelmap & saturatedpixelmap)
    stat = ima.iter_stat()
    m, s = stat.median, stat.stdev
    mask &= ima.thresh_to_pixmap(m-IMAGE_THRESH*s, m+IMAGE_THRESH*s)
    fstat = fringe.iter_stat()
    fm, fs = fstat.median, fstat.stdev
    mask &= ( fringe.thresh_to_pixmap(fm-FRINGE_THRESH_HIGH*fs,
                                      fm-FRINGE_THRESH_LOW*fs) |
              fringe.thresh_to_pixmap(fm+FRINGE_THRESH_LOW*fs,
                                      fm+FRINGE_THRESH_HIGH*fs) )
    return mask

  # calculate fringe map scale factor, scale and subtract from science
  mask = make_mask()
  I = ima
  F = fringe
  mean_I = I.Stat(pixelmap=mask).mean
  mean_F = F.Stat(pixelmap=mask).mean
```

```
N = mask.count()
a = (sum(mask,I*F)-N*mean_I*mean_F) / (sum(mask,F*F)-N*mean_F*mean_F)
ima = ima - a * fringe

stats1 = ima.iter_stat()

# compute background, and subtract
sextractor(ima, BACKGROUND_CONFIG,
                CHECKIMAGE_NAME=tmp_ima)
ima = ima-tmp_ima

# apply illumination correction
ima = ima * illumination
stats2 = ima.iter_stat()
```

6.3.3 Seq. — *Cat I: Construct weigths*

Objective:

Construct individual weight maps.

In addition to the effects of hot and cold pixels, individual images are contaminated by saturated pixels, cosmic ray events, and may be affected by satellite tracks. For purposes of subsequent analysis and image combination, pixels affected by these effects need to be assigned a weight of zero in weight maps that are unique to each image.

Saturated pixels are pixels whose counts exceed a certain threshold. In addition, saturation of a pixel may lead to 'dead' neighbouring pixels, whose counts lie below a lower threshold.

Cosmic-ray events can be detected using special source detection filters (retina filters), with Sextractor. These are essentially neural networks, trained to recognize cosmic rays, taking a set of neighbouring pixels as input (see Section 7.1 for further details).

Satellite tracks can be discovered by a line-detection algorithm such as the Hough transform, where significant signal along a line produces a 'peak' in the transformed image. This peak can be clipped, and transformed back into a pixelmap that masks the track (see Section 7.1 for further details).

Since the variance is inversely proportional to the Gain, which is proportional to the flatfield, the weight is given by:

$$W_{ij} = G_{ij} P_{hot} P_{cold} P_{saturated} P_{cosmic} P_{satellite}$$

where P are binary maps where good pixels have a value of 1 and bad pixels have a value of 0.

To take into account the illumination correction, as described in **req.**548, the weight should be divided by the illumination image.

Estimated time needed:

Approximately 1 min/CCD

Inputs:

SeqFile– 631 *Raw science frame* (for saturated pixelmap)
SeqFile– 632 *Reduced science frame* (for cosmic and satellite)
CalFile– 522 *Hot pixel map*
CalFile– 535 *Cold pixel map*
CalFile– 546 *Master flatfield*
CalFile– 548 *Illumination correction*

Outputs:

SeqFile– 633A *Saturated pixel map*
SeqFile– 633C *Cosmic pixel map*
SeqFile– 633S *Satellite track map*
SeqFile– 633 *Weight image*

Recipe:

This is covered by **Recipe– Reduce** which is documented in **seq.– 632**.

CA:

Saturated Pixel Map:
Process (make):
1. Create a pixelmap by applying a threshold to the raw data

Cosmic Map:
Process (make):
1. Construct a weight map from the flatfield, saturated, hot and cold pixelmaps.
2. Subtract the background from the reduced science image.
3. Multiply by the square-root of the weight to produce a SNR image.
4. Run Sextractor with a cosmic retina to produce a segmentation image and a catalogue of of cosmic-ray detections.
5. Use the segmentation image to produce a cosmic pixelmap.
6. Use the catalogue to count the number of cosmic ray events.

Satellite Map:
Process (make):
1. Construct a weight map from the flatfield, saturated, hot and cold pixelmaps.
2. Subtract the background from the reduced science image.
3. Multiply by the square-root of the weight to produce a SNR image.
4. Create a Hough-transformed image for pixels $> 5\sigma$.
5. Create a pixelmap by inverting the clipped Hough image.

Weight Image:
Process (make):
1. Combine the hot, cold, saturated, cosmic and satellite pixelmaps (logical AND).
2. Multiply the combined pixelmap with the master flatfield.
3. Divide the weight by the illumination correction.

CAP:

Saturated Pixel Map:
```
saturated = eclipse.threshold2pixelmap(raw, THRESHOLD_LOW, THRESHOLD_HIGH)
```

Cosmic Pixel Map:
```
Constants:
  COSMIC_CONFIG : SExtractor configuration parameters and their values
                  in case they differ from the SExtractor defaults.
          CHECKIMAGE_TYPE = 'SEGMENTATION'
          ANALYSIS_THRESH = 3.0
```

```
          CATALOG_TYPE    = 'ASCII'
          DETECT_MINAREA  = 1
          DETECT_THRESH   = 5.0
          GAIN            = 1
          MEMORY_PIXSTACK = 100000
          BACKPHOTO_THICK = 24
          BACK_TYPE       = 'AUTO'
          BACK_VALUE      = 0.0, 0.0
          FILTER          = 'Y'
          FILTER_NAME     = 'cosmic.ret'    # Retina filter for SExtractor
          CHECKIMAGE_NAME = 'cosmic.fits'   # Image with cosmic ray events

mask   = eclipse.pixelmap_2_image(hot & cold & saturated)
weight = eclipse.image_mul(flat, mask)
weight = eclipse.image_div_local(weight, illumination)
eclipse.image_pow_local(weight, 0.5)
snr_image = eclipse.image_mul(reduced, weight)

sextractor(snr_image, COSMIC_CONFIG)

cosmic = eclipse.threshold2pixelmap(cosmic, -0.5, 0.5)
```

Satellite Pixel Map:

```
mask   = eclipse.pixelmap_2_image(hot & cold & saturated)
weight = eclipse.image_mul(flat, mask)
weight = eclipse.image_div_local(weight, illumination)
eclipse.image_pow_local(weight, 0.5)
snr_image = eclipse.image_mul(reduced, weight)

stat = eclipse.stat_iter(snr_image)

hough = eclipse.hough_transform(snr_image,
                                stat.stdev*DETECTION_THRESHOLD)

satellite = eclipse.inverse_hough_transform(hough, HOUGH_THRESHOLD)
```

Weight image

```
mask   = eclipse.pixelmap_2_image(hot & cold & saturated & cosmic & satellite)
weight = eclipse.image_mul(flat, mask)
weight = eclipse.image_div_local(weight, illumination)
```

6.3.4 Seq. — *Cat I: Astrometric calibration, determine seeing*

Objective:

Determine the astrometric solution for the science image. Also determine the seeing.

Produce a catalog of the science image. Use the catalog to determine the astrometric solution and the seeing, and record the solution in the header of the image. The astrometrically calibrated image is used when building the dedithered image (**seq.– 636**).

Next to the basic scientific requirement of an accuracy of the absolute astrometry better than 0.1 arcsec rms, an upper limit on the astrometric uncertainties is determined by the requirement that astrometric inaccuracies do not degrade the seeing of the dedithered image (**seq.– 636**).

The full astrometric process solves simultaneously the focal plane assembly parameters and the telescope light-path parameters. A series of these solutions should result in a definition of the telescopes current focal plane structure, which in normal circumstances should be constant, and a definition of the light path deformation, which in normal circumstances can be variable. The empirical determination of the full telescope plate deformation can be used as a-priori knowledge in the pairing phase for another determination.

The default astrometric solution is obtained on a 'per chip' basis. Alternatively, one can choose to use the overlap that results from a repeated observation of the same area of sky in a jittered or dithered observation to improve on this solution. Improved accuracy is obtained if one uses the overlap information for a single chip. Maximum accuracy is obtained if one also uses the overlap between different chips in dithered observations. The latter method breaks the inherent parallelism of processing individual chips.

The US-NAVAL Observatory A2.0 catalog provides astrometric reference objects with a position accuracy of 0.3 arcsec rms. Since there are > 10 reference objects per CCD, even the default astrometric solution will give a position error $< 0.3/\sqrt(10) < 0.1$ arcsec rms.

The large CCD's are created by stitching smaller masks together. Between the stitches errors in the position of pixels with respect to a truly regular grid may occur. The method described above may be applied to individual stitches for each CCD, therefore increasing the number of free parameters of the solution. The small stitches are generally too small to yield enough pairs for high resolution solutions, but an ensemble of observations would. Determination of the stitching topology is a do-once operation.

Preliminary inspection of **OmegaCAM** CCDs has shown that the position error introduced by stitching is approximately 0.02 arcsec. Correction for stitching is therefore currently not considered.

Estimated time needed:

Approximately 30 sec./CCD.

Inputs:

SeqFile– 632 *Reduced science frame*
SeqFile– 633 *Individual weight image*
CalFile– in1 *US-NAVAL Observatory A2.0*

Outputs:

SeqFile– 634 *Astrometrically calibrated science image* with astrometry solution inserted into the descriptors
Astrometric quality parameters
Seeing **SeqFile– 634C** *Astrometrically calibrated catalog* - not further used
SeqFile– 634R *Astrometric residual map* - not further used

Recipe:

This is covered by **Recipe– Reduce** which is documented in **seq.– 632**.

CA:

Process (make):
1. Run sextractor to obtain input catalog.
2. Run LDAC-preastrom to produce preliminary calibrated catalog and preliminary local reference catalog.
3. Run LDAC-aplastrom, LDAC-associate and LDAC-make_ssc to create a pairs catalog of the preliminary calibrated catalog and the local reference catalog.
4. Run LDAC.astrom to determine the full astrometric solution from the preliminary calibrated catalog and the pairs catalog.
5. Apply the astrometric calibration and determine the wcs-parameters that describe the distortion.
6. Select stellar objects (no flag, bright but not too bright, and approximately the right size) from the catalog.
7. Use the median of the half-light radius (FLUX_RADIUS) as a measure of the seeing.

CAP:

```
Constants:
  APLASTROM_CONFIG : LDAC.aplastrom configuration parameters and their values
                     do NOT differ from the LDAC defaults.
  ASSOCIATE_CONFIG : LDAC.associate configuration parameters and their values
                     in case they differ from the LDAC defaults.
          INTER_COLOR_TOL = 2.5
          ISO_COLOR_TOL   = 1.5
          PAIR_COLS       = 'NO'
  ASTROM_CONFIG : LDAC.astrom configuration parameters and their values
                  in case they differ from the LDAC defaults.
          NITER           = 5
          XMIN            = 0.0
          XMAX            = 32767.0
          YMIN            = 0.0
          YMAX            = 32767.0
  MAKE_DISTORT_CONFIG : LDAC.make_distort configuration parameters and their
                        values do NOT differ from the LDAC defaults.
```

```
  MAKE_SSC_CONFIG : LDAC.make_ssc configuration parameters and their values
                    in case they differ from the LDAC defaults.
        MAKE_PAIRS      = 'YES'
  PREASTROM_CONFIG : LDAC.preastrom configuration parameters and their values
                     in case they differ from the LDAC defaults.
        MAX_OFFSET      = 150.0
        PHOT            = 'FLUX_ISO'
        RMS_TOL         = 2.57
        XMIN            = -32768.0
        YMIN            = -32768.0
  SEXTRACTOR_CONFIG : SExtractor configuration parameters and their values
                      in case they differ from the SExtractor defaults.
        ANALYSIS_THRESH = 3.0
        BACKPHOTO_THICK = 24
        BACK_TYPE       = 'AUTO'
        BACK_VALUE      = 0.0, 0.0
        GAIN            = 1
        MEMORY_PIXSTACK = 100000
        FILTER          = 'N'
        WEIGHT_TYPE     = 'MAP_WEIGHT'

incat = sextractor(image, SEXTRACTOR_CONF)

LDAC.preastrom(incat, outcat1, inref, outref1, PREASTROM_CONFIG)
LDAC.aplastrom(outcat1, outcat2)
LDAC.associate([outcat2, outref1], [outcat3, outref2], ASSOCIATE_CONFIG)
LDAC.make_ssc([outcat3, outref2], outref3, MAKE_SSC_CONFIG)

LDAC.astrom(outcat1, astcat1, outref3, ASTROM_CONFIG)
LDAC.aplastrom(astcat1, astcat2)
LDAC.make_distort(astcat2, astcat3)

wcs-parameters = LDAC.get_records(astcat3, table='DISTORT')

# determine seeing
data = LDAC.get_records(atscat3, table='OBJECTS')
stellar_radii = []
for object in data:
    if (object['Flag'] == 0 and
        peak_min < object['MaxVal'] < peak_max and
        radius_min < object['FLUX_RADIUS'] < radius_max):
```

```
        stellar_radii.append(object['FLUX_RADIUS'])

SEEING = median(stellar_radii)
```

6.3.5 Seq. — *Cat I: Apply photometric calibration*

Objective:

Apply photometric calibration and illumination correction
The photometric parameters (zeropoint, extinction, gain) are copied to the image header. Given the airmass and exposure time of the observation, the appropriate zeropoint and zeropoint error can be computed.
Zeropoint and extinction are determined differently for Key (u', g', r', i') and User (all other filters) bands. See **req.**561-**req.**564 for details.
Note that the zeropoint (ZP) is expressed in units of magnitude per count per second, Hence, to compute the magnitudes, the exposure time (t_{exp}, keyword: EXPTIME) should be taken into account.

$$m - M = ZP + 2.5 \log t_{exp}$$

Estimated time needed:

5 sec./CCD.

Inputs:

SeqFile– 634 *Astrometrically calibrated science image*
CalFile– 523 *CCD gain*
CalFile– 563 *Zeropoint and extinction - keybands*, or
CalFile– 564 *Zeropoint and extinction - userbands*

Outputs:

SeqFile– 634 *Astrometrically calibrated science image* zeropoint added

Recipe:

This is covered by **Recipe– Reduce** which is documented in **seq.– 632**.

CA:

The image zeropoint and zeropoint error are copied to the image header and descriptor.

CAP:

```
#copy keywords from photometric parameters and gain-linearity measurement
#to science image header
```

6.3.6 Seq. — *Cat I: Dedithering/dejittering*

Objective:

Combine all images of a pointing into a dedithered image and check the quality of the processed result.

Most science applications of OmegaCAM combine data of series of dithered (jittered) observations of a single pointing. The co-addition step maximizes the signal to noise ratio per pixel in the output image, and combines the background subtracted image data, using variance weighting. Since image artifacts (hot, cold, saturated pixels, cosmic-ray events, and satellite tracks) have zero-weight, the co-added image will be much "cleaner" than the input images. A dedithered weight image records the sum of the input weights for each pixel, and is proportional to the reciprocal of the variance in the dedithered image.

By co-adding onto a simple coordinate system—characterized by the projection (Tangential, Conic-Equal-Area), reference coordinates, reference pixel, and pixel scale—the distortions recorded by the astrometric solution are removed from the images.

The consortium proposes to define a standard grid on the Southern sky. Such a standard grid could involve a standardized set of field centers, at for instance 1 degree separation, and a standard set of pixels of for instance 0.2 arcsec size defined with respect to the fieldcenters. See Appendix A6 for a possible definition.

Standardization will result into similar grids of the output images for independent observations (different epochs, different filtering etc) for the same region of sky, without any further administrational burden, facilitating direct image comparison on pixel-to-pixel level without additional interpolations.

Since different input images may have a different zeropoint, the dithering process applies a scaling factor to each input image as well as each weight image.

Note that the current weighting scheme is based on maximizing the signal-to-noise ratio per pixel in the dedithered image. However, when images with different seeing are combined, adding images with bad seeing may lead to a degradation of the signal-to-noise ratio per object, even when the signal-to-noise-ratio per pixel is increased. Therefore, a seeing-based weight-adjustment may be desirable (TBD)

Swarp is a package specifically designed to handle these operations.

The size of the output image is adjusted automatically to completely cover the area covered by the input images. Alternatively, a maximum size of the output image may be defined, which will clip data lying outside this predefined region.

In order to validate the result of the image pipeline, **seq.– 636** should finish with a Quality Check.

A catalog of the dedithered image is obtained in order to measure the PSF variation over the FoV of **OmegaCAM**. This measurement should be compared to the PSFs of the input data (**seq.– 634**) to provide a check on the entire coaddition process. The degradation of the PSF should not exceed the expected degradation due to astrometric uncertainty.

In addition it may be necessary to compare the positions and photometry of the catalogs of the input images to catch possible systematic errors in the relative astrometry and relative photometry.
Finally, the catalog of the dedithered image can be used to check the astrometry w.r.t an external catalog, and to compare the image sensitivity to the one predicted by the ETC. These checks provide end-to-end sanity checks of the entire observation and reduction process.

Estimated time needed:

Regridding: 1 min./CCD Coaddition: Approximately 2 min. for 5 science frames

Inputs:

The input for dedithering is an arbitrary large list of **SeqFile– 634** *Astrometrically calibrated science data* with photometry and seeing present in the descriptors, and
SeqFile– 633 *Individual weight images*

output coordinate system (projection, reference coordinates, pixel scale) - (most likely fixed in the code itself - TBC)

Outputs:

SeqFile– 636 *Dedithered image* including statistics and seeing
SeqFile– 636W *Dedithered weight_image*
SeqFile– 636cat *Catalog*
sensitivity description

Recipe:

Note that this recipe only performs the co-addition part of **seq.– 636.**

```
Coadd -i <regridded_frames> [-t target]

regridded_frames : Regridded science frames as produced by the Reduce recipe
target           : Filename of the resulting coadded image (string)
```

CA:

1. Compute FLUXSCALES necessarily to normalize all input data to a single zeropoint.
2. Resample science data to a new grid. A suggested grid is given in Appendix 6.
3. Coadd input data with **Swarp**.
4. Determine statistics of output image.
5. Run **Sextractor** on the output image.
6. Use the median of the half-light radius (FLUX_RADIUS) as a measure of the seeing.

CAP:

```
Constants:
  SWARP_REGRID_CONFIG : Swarp regridding configuration parameters and their
                        values in case they differ from the Swarp defaults.
```

```
    COMBINE         = 'N'
    CENTER_TYPE     = 'MANUAL'
    MEM_MAX         = 384
    PIXELSCALE_TYPE = 'MANUAL'
    WEIGHT_TYPE     = 'MAP_WEIGHT'
  SWARP_COADD_CONFIG : Swarp coaddition configuration parameters and their
                       values in case they differ from the Swarp defaults.
    COMBINE         = 'Y'
    COMBINE_TYPE    = 'WEIGHTED'
    MEM_MAX         = 384
    RESAMPLE        = 'N'
    WEIGHT_TYPE     = 'MAP_WEIGHT'

# regridding (done in Reduce recipe, on individual science image)
regridded_frame = Swarp(science_frame,
                        SWARP_REGRID_CONFIG,
                        CENTER=RA, DEC,
                        PIXEL_SCALE=PIXELSCALE,
                        FSCALE_DEFAULT=science_frame.get_fluxscale())

# coaddition (done in Coadd recipe, on multiple regridded images)
# regridded_frames could be a list of images from a dither Observing Block
coadded_image = Swarp(regridded_frames,
                      SWARP_COADD_CONFIG,
                      WEIGHT_SUFFIX='.weight')

coadded_image.statistics()

catalog = coadded_image.sextractor(weight=coadded_image.weight)
output coadded_image, coadded_weight
```

7 FUNCTIONALITIES

Sections 5 and 6 include for each Data product a list of functionalities required for the reduction algorithms. In this section we give an overview of these functionalities, and describe how the functionalities are provided by external packages and libraries.

Sections 5 and 6 also give pseudo code descriptions of algorithms. The pseudocode describes the algorithms in terms of low-level functionalities. In this section we also describe the (Python) interfaces to low-level functions. This should aid the reader in understanding the pseudocode, which uses the same interfaces.

Configuration files

This section refers to a number of external programs (**Sextractor**, **LDAC**, **Swarp**) that are driven by configuration files. The interfaces to these packages write the configuration files on the fly, based on defaults defined in the interfaces, and overridden by configuration parameters supplied by the user.

Alternatively, these programs can be run as separate programs with default configurations (provided in the distribution) with optional overrides for configuration parameters passed as key-value pairs on the command line.

7.1 Image manipulation ECLIPSE/SWARP

Image manipulation is required throughout the data reduction pipeline. Most of the image manipulation algorithms are provided by eclipse.

The Python interface used by OmegaCAM uses a wrapped version of the eclipse C-library. This makes the C-functions in eclipse available to Python. A wrapper around these function provides an Object-Oriented interface to fundamental data reduction objects: **images**, **cubes** (arrays of images), **headers** and **pixelmaps** ('boolean' images). Hence, the low level C-functions map to methods on these Python objects.

image arithmetic

Standard arithmetic operations on images include addition, subtraction, division and multiplication with two images as operands and an image and a scalar as operand. In-place variations of these operations, whereby the first operant itself is modified by the operation are also used.

Pixelmaps record boolean (True/False, good/bad) values. Arithmetic operations on pixelmaps include logical-AND and logical-OR operations and their in-place variants.

The following table lists typical arithmetic operations on images and pixelmaps, and the corresponding eclipse functions.

Python code	C-function
`ima1 = ima2 + ima3`	`ima1 = image_add(ima2, ima3)`
`ima1 -= ima2`	`imaga_sub_local(ima1, ima2)`
`ima1 = ima2 * scalar`	`ima1 = image_cst_op(ima2, scalar, '*')`
`ima /= scalar`	`image_cst_op_local(ima, scalar)`
`ima1 = scalar ** ima2`	`ima1 = image_cst_op(ima2, scalar, 'e')`
`ima **= scalar`	`image_cst_op_local(ima, scalar, '')`
`pmap1 &= pmap2`	`pixelmap_binary_AND(pmap1, pmap2)`
`pmap1 = pmap2|pmap3`	`tmp = pixelmap_copy(pmap2)`
	`pmap2 = pixelsmap_binary_OR(tmp, pmap2)`

image statistics

The standard statistics routine provides the mean, standard deviation, minimum and maximum of an image. It is also possible to obtain the median. However, since this is a relatively expensive operation, medians should in general not be computed when the image statistics are determined. It is possible to restrict the determination of statistics to pixels lying within a given range or within a given area. It is also possible to provide a pixelmap which masks invalid pixels that should not be included in the statistics determination.

To improve the accuracy of the image statistics it is possible to iteratively reject outliers (e.g.: stars, galaxies, defects) from the image.

The following table lists the available statistics methods and the corresponding **eclipse** functions.

Functionality	Python code	C-function
Standard	`stats = ima.stat()`	`stats = image_stat(ima)`
Restricted	`stats = ima.stat_opts(...)`	`stats = image_stat_opts(ima, ...)`
Iterative	`stats = ima.iter_stat(...)`	`stats = image_iter_stat(ima, ...)`
Iter.Restr.	`stats = ima.iter_stat_opts(...)`	`stats = image_iter_stat_opts(ima, ...)`

trimming and overscan correction

Overscan correction entails the subtraction of a suitable bias level from the image, as determined from the overscan region of the image. In the pipeline the following methods are supported:

0 Do not do overscan correction.

1 Use the median of the prescan-X region

2 Use the median of the overscan-X region

3 Use the median of the prescan-Y region

4 Use the median of the overscan-Y region

5 Per-row subtraction of median of row in prescan-X

5 Per-row subtraction of median of row in overscan-X

Methods 1-4 use **eclipse** function `image_stat(...)` to compute the median of the region and standard image arithmetic to subtract this scalar. Methods 5 and 6 use **eclipse** function `image_oscan(ima, start_column, end_column)` which subtracts the median in a row between the start and end columns from the entire row.

Note that it is important that calibration data and science data use the same overscan correction method.

Trimming means extracting the sub-image containing the exposed part of the raw data. The **eclipse** function `image_getvig(ima, x0, y0, x1, y1)` can be used to extract a region spanned by the lower left corner `(x0, y0)` and the upper right corner `(x1, y1)`.

averaging

Given a list of equally dimensioned input images, an output image containing the average of the input images is determined. Several methods are provided by **eclipse**

Mean The mean of a series of input images is obtained by computing the sum if the images and dividing by the total number of input images, using standard image arithmetic (q.v.). Alternatively the function `cube_avg_linear()` could be used.

Mean with rejection Each output pixel is equal to the mean of the corresponding input pixels that have not been rejected. Rejection algorithms can be based on comparing measured deviation from, for example, the median and the expected deviations in the input data. Examples can be found in **req.**541 and **req.**542.

Median The median can be computed using the function cube_avg_median()

masking

The **eclipse pixelmap** is used to store image masks. The **pixelmap** uses boolean values to indicate good (1) and bad (0) pixels.

A mask is made by the method ima.thresh_to_pixmap(low, high), corresponding to the C-function image_threshold2pixelmap(ima, low, high). The function assigns a value of 1 (good) to pixels within the range $low < I(x, y) < high$ and a value of 0 (bad) outside this range.

To convert a pixelmap to an image, use the method pmap.as_image(), corresponding to the C-function pixelmap_2_image(pmap)

satellite detection

It is expected that at least one chip per dither will be affected by satellite tracks. These linear features can be detected using the Hough algorithm.

A point (x, y) defines a curve in Hough space (ρ, θ), where:

$$\rho = xcos\theta + ysin\theta,$$

corresponding to lines with slopes $0 < \theta < \pi$, passing at a distance ρ from the origin. This means that different points lying on a straight line in image space, will correspond to a single point (ρ, θ) in Hough space.

The Hough algorithm creates a Hough image from an input image, by adding a Hough curve for each input pixel which lies above a given threshold. This Hough image (effectively a histogram of pixels corresponding to possible lines) is clipped, and transformed back into a pixelmap, masking lines with too many contributing pixels.

The Hough transform and inverse transform have been implemented in **eclipse** by the functions hough_transform(), and hough_transform_to_pixelmap() respectively.

cosmic ray detection

Sextractor can be run with a special filter that is only sensitive to cosmic-ray-like signal. This requires a 'retina' filter, which is a neural network that uses the relative signal in neighboring pixels to decide if a pixel is a cosmic. A retina filter, called 'cosmic.ret' is provided. Run Sextractor with FILTER_NAME=cosmic.ret, to run Sextractor in comic ray detection mode. This results in a so-called segmentation map (CHECKIMAGE_TYPE=SEGMENTATION), recording the pixels affected by cosmic ray events. This segmentation can be used to assign a weight of zero to these pixels.

background subtraction

An efficient and robust method to produce an estimate of the background is used internally by Sextractor. This method is based on estimates of the local background on a mesh, using both the mean and the mode of the distribution of local pixel values. The grid of background estimates is then interpolated to produce an estimate of the background at each position in the image.

To use Sextractor to produce a background image, run Sextractor with `CHECKIMAGE_TYPE = 'BACKGROUND'`, to produce the image. Use a high detection threshold to avoid unnecessary computation of source parameters, which we are not interested in here.

coaddition

Given a list of (overlapping) astrometrically calibrated input images and their corresponding weights, build a coadded image.

Image coaddition is divided in two parts, resampling and combination.

Image resampling addresses the problem that two independent observations of the same area of sky will, in general, result in images whose coordinate systems are different. Projection of these coordinate systems to a new coordinate system requires a mapping $x, y \Rightarrow \alpha, \delta \Rightarrow x', y'$. Since the area of sky covered by an input pixel will in general not map directly to an area of sky covered by a single output pixel, some sort of interpolation is required. Unfortunately interpolation will inevitably result in aliasing artifacts, hence, a careful choice of interpolation kernel is required.

Once all input images and their weights have been resampled onto the grid specified by the coordinate system of the coadded image, it is straightforward to compute the coadded image and its weight. Given resampled input images $1 \leq i \leq N$ entering co-addition, one can define for each pixel j:

- the local uncalibrated flux $f_{ij} = \overline{f_{ij}} + \Delta f_{ij}$, where $\overline{f_{ij}}$ is the sky background, and Δf_{ij} the contribution from celestial sources,
- the local uncalibrated variance $\sigma_{ij}^2 = \overline{\sigma_{ij}^2} + \Delta\sigma_{ij}^2$, where $\overline{\sigma_{ij}^2}$ is background noise and $\Delta\sigma_{ij}^2$ the photon shot noise of celestial sources,
- the local, normalized weight w_{ij},
- the electronic gain of the CCD g_i,
- the relative flux scaling factor p_i, deduced from the photometric solution, with $p_i \Delta f_{ij} = p_l \Delta f_{lj}$ for all i, l, j,

  the weight scaling factor k_i.

Optimal weighting is obtained using

$$k_i w_{ij} = \frac{1}{p_i^2 \overline{\sigma_{ij}^2}},$$

leading to a co-added flux (assuming the background of the input images has been subtracted),

$$f_j = \frac{\sum_i k_i w_{ij} p_i \Delta f_{ij}}{\sum_i k_i w_{ij}},$$

and variance

$$\sigma_j^2 = \frac{\sum_i k_i^2 w_{ij}^2 p_i^2 \sigma_{ij}^2}{(\sum_i k_i w_{ij})^2},$$

The software package **Swarp**, developed by E. Bertin (IAP) for Terapix provides a set of regridding and co-addition algorithms specifically optimized to handle large area CCD mosaics

7.2 Catalog manipulation LDAC/ SEXTRACTOR

Several reduction steps, in particular astrometric and photometric calibration, require the manipulation of source lists/catalogs. Most of this functionality is provided by the Leiden Data Analysis Center (**LDAC**) data reduction tools. Catalogs are extracted from the image data using **Sextractor**

Sextractor and **LDAC** are driven by a configuration files. The interfaces to these packages write the configuration files on the fly, based on defaults defined in the interfaces, and overridden by configuration parameters supplied by the user. The programs can also be run separately, based on default configurations (provided in the distribution) with optional overrides for configuration parameters passed on the command line.

source extraction

The detection and measurement of sources in images is a non-trivial task. Fortunately an established, versatile, robust and fast algorithm for the detection of sources is available in the form of the **SExtractor** source extraction package. In addition to its source detection capabilities **SExtractor** provides functionality for the detection of image defects (cosmic rays and bleeding), and for robust background subtraction in the presence of signal.

association

Both astrometric and photometric calibration require the association of a source catalog with a reference catalog. A proper association algorithm takes into account position errors and confusion problems.

Catalog association is provided by the **LDAC** program `associate`

merge

In some cases it may be necessary to merge overlapping catalogues into a new catalogue, containing a suitable average of the different measurement of a single object in different observations.

Catalog merging is provided by the **LDAC** program `make_ssc`

astrometric solution

The astrometric solution perfoms, based on the a-priori pointing knowledge of the telescope and the chip layout, a pattern matching between the extracted observed objects and a reference catalog allowing quick, inaccurate pairing. To increase the reliability of the initial pairing, previous determinations of the full astrometric solution are used as a first step approximation.

Using this pairing information the parameters of the astrometric model are derived in an iterative process. The full astrometric solutions determine both the projection of the individual chip pixel coordinates onto the perfect (x,y) coordinate space of the focal plane and the projection of this perfect plane to sky coordinates. This effectively maps the light path through the telescope and the camera for the particular pointing from the pixel plane to the sky coordinate domain. If multiple pointings are available (as is the case in **Mode– dither** N= or **Mode– jitter** N=) different pixels and/or detectors map the same sky area. The source extraction precision in arcseconds is much higher than the positional accuracy of the reference catalog. Incorporating the overlap information therefore enhances the accuracy of the final astrometric solution to allow more precise co-addition.

The astrometric solution can be derived in several ways of increasing complexity and potentially increasing accuracy. Single chip solutions where the astrometric solution for each CCD chip is derived independently is the simplest and least accurate. When multiple pointings on the same area of sky are performed close in time, for each CCD chip one will have multiply observed the set of references stars and one can also use non-reference stars observed multiple times among the pointing series. Still an astrometric solution per chip is derived, which is more accurate than the single chip, single pointing case. The most robust way of determining an astrometric solution of a multi CCD camera is to make use of the dithering pattern which creates not only overlaps of one CCD chip onto itself, but also among the different chips of the camera. Now a full astrometric solution can be derived for all chips simultaneously using all overlap cases (chips onto themselves, between different chips). Obviously this can produce the most accurate astrometric solution of the three.

As is customary, the plate model is not defined as a mapping between pixel coordinates and sky coordinates (α,δ), as this would obviously have singularities at the poles, but instead as a mapping between the pixel space and so-called normal coordinates. The normal coordinates (ζ, η) are the gnomonic projection of the absolute sky positions relative to the field center. There is some level of non-linearity here, since the field center itself depends on the position reconstruction, but for realistic initial errors this is very small and easily absorbed in the plate model. With these notations, the mapping between a source detected at pixel coordinates (x,y) in frame f and its normal coordinates (ζ, η) relative to the nominal center of field f is of the form

$$\zeta(x, y, f) = a_f + \sum_{i+j=1}^{i+j=p} \sum_{k=0}^{n} b_{ijk} x^i y^j f^k$$

$$\eta(x, y, f) = c_f + \sum_{i+j=1}^{i+j=p} \sum_{k=0}^{n} d_{ijk} x^i y^j f^k$$

where the a_f, c_f, b_{ijk}, and d_{ijk} are initially unknown coefficients, to be determined by the least squares adjustment. We actually use a different maximum degree in the field number variable for each spatial term, since flexures only induce measurable effects for the low order terms (zero point and focal scale).

astrometric solution – Overlap solution

Upon closer examination, the star density of the reference catalog is indeed almost always sufficient. One does, however, gain accuracy by incorporating the better positional accuracy within the chips of a detector than that with respect to an external reference frame. Giving overlap position determinations higher weight than those with respect to the external reference system one enhances positional accuracy.

The coefficients of the plate model will thus be determined by simultaneously minimizing (in a least square sense) the distance to the reference stars and the overlap detections. The corresponding quadratic form can be written as:

$$S = \sum_{f=1}^{n_{field}} \sum_{i=1}^{n_{ref}^{f}} \frac{\left\| \zeta(x_i, y_i, f) - \zeta_i^f \right\|^2}{\sigma^2_{mes}(i) + \sigma^2_{cat}(i)} + \sum_{f=1}^{n_{field}} \sum_{i=1}^{n_{ref}^{f}} \frac{\left\| \eta(x_i, y_i, f) - \eta_i^f \right\|^2}{\sigma^2_{mes}(i) + \sigma^2_{cat}(i)}$$
$$+ \sum_{f_1=1}^{n_{field}} \sum_{f_2=1}^{n_{field}} \sum_{i=1}^{n_{over}^{f_1 f_2}} \frac{\left\| (\zeta(x_i, y_i, f_1) - \zeta(x_i, y_i, f_1)) - (\zeta_i^{f_1} - \zeta_i^{f_2}) \right\|^2}{\sigma^2_{mes}(i, f_1) + \sigma^2_{mes}(i, f_2)}$$
$$+ \sum_{f_1=1}^{n_{field}} \sum_{f_2=1}^{n_{field}} \sum_{i=1}^{n_{over}^{f_1 f_2}} \frac{\left\| (\eta(x_i, y_i, f_1) - \eta(x_i, y_i, f_1)) - (\eta_i^{f_1} - \eta_i^{f_2}) \right\|^2}{\sigma^2_{mes}(i, f_1) + \sigma^2_{mes}(i, f_2)}$$

where ζ and η are the multivariate polynomial functions described above, ζ_i^f and η_i^f are the normal coordinates of reference star i relative to the center of field f, and $(\zeta_i^{f_1} - \zeta_i^{f_2})$ and $(\eta_i^{f_1} - \eta_i^{f_2})$ are the difference of the normal coordinates of overlap star i relative to the centers of fields f_1 and f_2. In principle, the latter quantities can only be evaluated once the coordinates are at least approximately known, but in practice they only vary by a few milliarcseconds across the overlap zone. One can also note that the solutions for ζ and η are not coupled, and that one thus independently minimizes S_ζ and S_η, the parts of S depending on respectively ζ and η.

Astrometric solution – Implementation

We consider here an ensemble of n_f fields containing a total of N_{ref} reference stars and N_{over} stars in overlaps, and adjust a plate model which has n_{par} smoothly varying functions (such as the polynomials discussed above) with coefficients noted as b_p, plus one zero point per field noted as b_f.

For a practical discussion of the implementation, it is convenient to rewrite the quadratic form of the above equation in its canonical form

$$S = (A \cdot X - B)^T \cdot (A \cdot X - B)$$

where A is a rectangular matrix (the design matrix) whose dimensions are the total number of constraints (reference+overlap) and the total number of fitted coefficients (field centers+smooth functions). X and B are vectors containing respectively the fitted coefficients and the constraints data. If one defines X, B, and A by respectively

$$X_i = a_f \text{ for } i \le n_f$$
$$X_i = b_p \text{ for } i > n_f$$

$$B_i = \zeta_{if_i} \text{ for } i \le N_{ref}$$
$$B_i = (\zeta_{if_i^1} - \zeta_{if_i^2}) \text{ for } i > N_{ref}$$

$$A_{ij} = \delta_{jf_i}/\sigma_i \text{ for } i \le N_{ref} \text{ and } j \le n_f$$
$$A_{ij} = f(x_i, y_i, f_i)/\sigma_i \text{ for } i \le N_{ref} \text{ and } j > n_f$$
$$A_{ij} = (+\delta_{jf_i^1} - \delta_{jf_i^2})/\sigma_i \text{ for } i > N_{ref} \text{ and } j \le n_f$$
$$A_{ij} = (f(x_i, y_i, f_i^1) - f(x_i, y_i, f_i^2))/\sigma_i \text{ for } i > N_{ref} \text{ and } j > n_f$$

where δ_{ij} is as usual a Kronecker symbol. A similar formulation obviously exists for η.

Under this canonical formulation of the least square quadratic form, it is easy to show (see for instance the least square chapter in Numerical Recipes) that the X which minimizes S is the solution of the system of $(n_{par} + n_f)$ linear equations $(A^T \cdot A) \cdot X = A^T B$, known as the normal equations.

The plate model coefficients can thus be obtained by computing $(A^T \cdot A)$ and solving the normal equations. This method is generally known to have relatively poor numerical stability. Once the pixel coordinates are remapped to [0,1] however, the problem has good conditioning, and direct comparison with more accurate methods shows good agreement for modest polynomial degrees, even in single precision. Alternatively, A is sparse enough that solving the equation with the conjugate gradient algorithm is a possibility. This iterative minimization algorithm converges in at most $n_{par} + n_f$ steps, each of which needs one multiplication by A and its transpose. It has good stability, is about as fast as computing $(A^T \cdot A)$, but needs a larger memory space.

If on the other hand an analytic estimate of the standard errors is needed, $(A^T \cdot A)^{-1}$ (which is the covariance matrix) must be evaluated anyway. Solving the normal equations then becomes much cheaper, though we may still use the conjugate gradient for the minimization, because of its better numerical behaviour.

The complete implementation of these algorithms is provided by the following sequence of **LDAC** programs

`preastrom` Used to determine an initial estimate of the astrometric solution. Inputs are the source catalog and the astrometric reference catalog. Outputs are a source catalog with astrometric solution, and a catalog of reference objects with the image region aplastrom, associate, and make_ssc are used to make a single source catalog containing the paring between the (pre-)astrometrically calibrated source catalog and the reference catalog.

astrom Compute the full astrometric solution. Inputs are the pre-astrometrically calibrated catalog, and the paired source-reference catalog. Output is the astrometrically calibrated catalog.

aplastrom Applies the astrometric solution in astrometrically calibrated catalog to the sources in that catalog.

make_distort computes the wcs-parameters for the image and stores these in the astrometrically calibrated catalog.

seeing

The seeing is robustly characterized by the half light radius FLUX_RADIUS of unsaturated stellar objects, obtained from **sextractor**

select stars

Star galaxy separation is functionality provided by **sextractor**, or one can use the half light radius to select stellar objects.

PSF fitting

The PSF is characterized by the second order moments of the brightness distribution of stellar objects. The distribution of the position angles and elongations of the bounding ellipse has to be evaluated. **Sextractor** provides measured second order moments, and the parameters (elongation, major/minor axes, position angle) that can be derived from these.

More robust estimates of the second order moments can obtained by fitting PSF models to stellar objects. This functionality is also provided by **Sextractor**, but currently undocumented, and therefore also not supported part of the current software.

7.3 Header manipulation ECLIPSE

Header manipulation, i.e.: reading and writing and modifying (representations of) FITS headers, constitutes basic functionality, provided by many astronomical data reduction packages. Missing from most header manipulation software however are validation tools. We also note that it is nowhere specified in what order header keywords are written, while some sort of fixed order would greatly improve human readability of FITS headers.

Currently, only the basic header functionalities provided by **eclipse** are used. Eclipse uses the **qfits** library for manipulating headers. In the pipeline a **header** object is used to manipulate headers.

reading

Header reading includes copying of the FITS header in some kind of internal representation, as well as the extraction of values for a particular keyword from a given header. A header is read using `qfits_header_load(fitsfile, extension)`. The basic function to obtain a value from a key is `qfits_header_getstr(header, key)` which returns a string containing the value.

writing

Header writing includes writing a FITS header based on some internal representation, as well as writing a value for a particular keyword in a given header. Header are written when saving images, using `image_save_fits_hdrdumo(image, fitsfile, header)`. Individual keys are written using `qfits_header_add(header, key, val, comment)` if the key does not exist yet and else: `qfits_header_mod(header, key, val, comment)`.

Header writing should conform to the formatting directives given by a Data Interface Dictionary. However, no automatic checks are performed (yet) to ensure this.

validation

Header validation should be based on Data Interface Dictionaries. There are currently no tools that verify existing FITS headers against Data Interface Dictionaries. The Fits Translation Utility (FTU) only validates new header items written by this tool.

7.4 Miscellaneous functionalities

Miscellaneous functionalities required for **OmegaCAM** data reduction include:

regression/(non-)linear fitting

Some results (zeropoints, gains, illumination correction parameters) are obtained from fits to multi dimensional data. The software provides one- and two-dimensional least squares fitting algorithm for first and second order polynomials

A1 LIST of CALIBRATION REQUIREMENTS

5.2.1 Req.– *Cat I: CCD read noise - doit* . . . 39
5.2.2 Req.– *Cat I: Hot pixels* . . . 41
5.2.3 Req.– *Cat I: CCD gain* . . . 43
5.2.4 Req.– *Cat III: Electromagnetic Compatibility* . . . 46
5.2.5 Req.– *Cat III: Electrical cross talk* . . . 47
5.3.1 Req.– *Cat I: CCD Dark Current - doit* . . . 49
5.3.2 Req.– *Cat I: CCD Particle Event Rate* . . . 53
5.3.3 Req.– *Cat I: CCD Linearity* . . . 54
5.3.4 Req.– *Cat III: CCD Charge Transfer Efficiency* . . . 57
5.3.5 Req.– *Cat I: CCD Cold Pixels* . . . 59
5.3.6 Req.– *Cat III: CCD Hysteresis, strong signal* . . . 61
5.4.1 Req.– *Cat I: Bias - doit* . . . 66
5.4.2 Req.– *Cat I: Flat-field - dome key bands + user bands - doit* . . . 70
5.4.3 Req.– *Cat I: Flat-field - twilight* . . . 75
5.4.4 Req.– *Cat I: Flat-field - night sky* . . . 80
5.4.5 Req.– *Cat I: Flat-field - fringing* . . . 84
5.4.6 Req.– *Cat I: Flat-field - master flat* . . . 88
5.4.7 Req.– *Cat I: Quick detector responsivity check-doit* . . . 92
5.4.8 Req.– *Cat II: Illumination correction* . . . 94
5.5.1 Req.– *Cat III: Position of Camera in focal plane* . . . 100
5.5.2 Req.– *Cat III: Telescope Pointing and offsetting* . . . 102
5.5.3 Req.– *Cat III: Telescope and Field Rotator tracking* . . . 103
5.5.4 Req.– *Cat III: PSF Anisotropy* . . . 105
5.5.5 Req.– *Cat I: The astrometric solution for templates - doit -see 6.3.4* . . . 108
5.5.6 Req.– *Cat III: The astrometric solution for Guide CCD's* . . . 109
5.6.1 Req.– *Cat III: Shutter Timing* . . . 115
5.6.2 Req.– *Cat I: Photometric Calibration - monitoring* . . . 117
5.6.3 Req.– *Cat I: Photometric Calibration - zeropoint keybands - doit* . . . 126
5.6.4 Req.– *Cat I: Photometric Calibration - zeropoint user bands* . . . 132
5.6.5 Req.– *Cat I: Filter band passes - user bands vs key bands* . . . 136
5.6.6 Req.– *Cat III: Dependency on angle - ADC, rotator/reproducibility* . . . 137
5.6.7 Req.– *Cat III: Linearity (as a function of flux)* . . . 138
5.6.8 Req.– *Cat III: Detection limit and ETC calibrations* . . . 139
5.6.9 Req.– *Cat I: Secondary Standards* . . . 140
5.7.1 Req.– *Cat III: Camera focus/tilt* . . . 144
5.7.2 Req.– *Cat III: Ghosts - ADC* . . . 150

A2 LIST of RAW CALIBRATION DATA

	volume per **req.**run and per year
RawData– 523 *CCD gain*	10Gb 0.52Tb
RawData– 531 *Dark current*	0.75Gb 40Gb
RawData– 541 *Raw Bias frame*	5Gb 0.51Tb
RawData– 542 *Dome flat frame*	2Gb 0.66Tb
RawData– 543 *Twilightflat frame*	10Gb 3.3Tb
RawData– 547 *Quick check*	0.5Gb 115Gb
RawData– 562 *Monitor*	0.5Gb 0.5Tb
RawData– 563 *Zeropoint- Key*	2.5Gb 0.8Tb
RawData– 564 *Zeropoint - User*	2.5Gb 0.8Tb
RawData– 565 *User -> key*	25Gb 25Gb
RawData– 569 *Secondary Standards*	64 Gb 1.5Tb

A3 LIST of DFS I/O CALIBRATION FILES

volume per file and per year	
CalFile– 521 *QC-parameter read_noise*	2 kb, 1 Mb
CalFile– 522 *bad/hot pixel map*	0.25Gb 25Gb
CalFile– 523 *QC parameter gain e^-/ADU*	2kb .1Mb
CalFile– 531 *Dark count rate for each CCD*	2kb .1Mb
CalFile– 532 *Particle event rate*	2kb .1Mb
CalFile– 533P *CCD Linearity Plot*	2kb .1Mb
CalFile– 533M *CCD Linearity Map*	0.25Gb 0.3Tb
CalFile– 534 *charge transfer efficiency factors*	.2Mb .4Mb
CalFile– 535 *Cold pixel map*	.25Gb 1Gb
CalFile– 536 *CCD Hysteresis*	2kb 2kb
CalFile– 541 *Master Bias frame*	1Gb 104Gb
CalFile– 542 *Master Domeflat frame*	1Gb 0.330Tb
CalFile– 542L *Dome Lamp*	2kb 1Mb
CalFile– 543 *Master Twilightflat frame*	4Gb 1.3Tb
CalFile– 544 *Nightsky flat frame*	1Gb 1Tb
CalFile– 545 *ff-fringe*	1Gb 0.3Tb
CalFile– 546 *Master flatfield*	1Gb 0.3Tb
CalFile– 547 *Quick check*	1Gb 0.33Tb
CalFile– 547r *Quick check - day report*	3kb 1Mb
CalFile– 548F *Illumination correction fit coefficients*	15kb 180kb
CalFile– 548 *Illumination correction*	1Gb 10Gb
CalFile– 554 *PSF anisotropy*	.1Mb 1 Mb
CalFile– 556 *Astrometric solution - Guide CCD*	2 kb 1 Mb
CalFile– 562Su *Sky brightness-u'*	1 kb 1 kb
CalFile– 562Sg *Sky brightness-g'*	1 kb 1 kb
CalFile– 562Sr *Sky brightness-r'*	1 kb 1 kb
CalFile– 562Si *Sky brightness-i'*	1 kb 1 kb
CalFile– 562 *Extinction-night report*	2 kb 1 Mb
CalFile– 562T *Photom + Sky*	5Mb 2Gb
CalFile– 563 *Zeropoint - key bands*	4kb 1.2Mb
CalFile– 564 *zeropoint - User bands*	4kb 1.2Mb
CalFile– 565 *User -> key*	4kb 1Mb
CalFile– 565C *composite -> monolithic*	4kb 1Mb
CalFile– 569 *Secondary Standard Catalog*	1 Mb 1 Gb

A4 LIST of DFS INPUT REFERENCE CATALOGUES

CalFile– in1 *US-NAVAL Observatory A2.0*	6 Gbyte
CalFile– 569E *Primary Standard stars - external Landolt fields*	100 Mb
CalFile– 564E *Standard extinction curve*	0.1Mb

NOTE:

Both the preparatory programme and **req.**569 will produce **CalFile– 569** *Secondary standards*.

CalFile– 569 *Secondary Standard stars - preparatory programme*	100 Mb

A5 LIST of SEQUENTIAL STEPS in IMAGE PIPELINE

6.3.1 Seq.– *Cat I: Ingestion and image statistics* 159
6.3.2 Seq.– *Cat I: Trim, de-bias, flatfield* 161
6.3.3 Seq.– *Cat I: Construct weigths* 166
6.3.4 Seq.– *Cat I: Astrometric calibration, determine seeing* 169
6.3.5 Seq.– *Cat I: Apply photometric calibration* 173
6.3.6 Seq.– *Cat I: Dedithering/dejittering* 174

A6 SKYGRID of PROJECTION CENTRES

The following table describes a grid on the sky for projection and co-addition purposes in a condensed format. It contains 94 strips as function of decreasing declination ($0° \geq \delta \geq -90°$). For each strip the size in degrees and the number of $1° \times 1°$ fields per strip is given. The last column contains the overlap between fields in %. By mirroring the grid along the equator one obtains a grid for the northern hemisphere. The combination of the grids for both hemispheres is a grid for the entire sky.

	$-\delta$	size				$-\delta$	size				$-\delta$	size		
	°	°	#	%		°	°	#	%		°	°	#	%
1	0.00	360.00	378	5.0	33	30.64	309.74	323	4.3	65	61.28	173.01	182	5.2
2	0.96	359.95	378	5.0	34	31.60	306.64	320	4.4	66	62.23	167.71	176	4.9
3	1.91	359.80	378	5.1	35	32.55	303.44	317	4.5	67	63.19	162.36	170	4.7
4	2.87	359.55	378	5.1	36	33.51	300.16	314	4.6	68	64.15	156.97	164	4.5
5	3.83	359.20	377	5.0	37	34.47	296.80	311	4.8	69	65.11	151.54	158	4.3
6	4.79	358.74	376	4.8	38	35.43	293.35	308	5.0	70	66.06	146.06	152	4.1
7	5.74	358.19	375	4.7	39	36.38	289.83	304	4.9	71	67.02	140.54	146	3.9
8	6.70	357.54	374	4.6	40	37.34	286.22	300	4.8	72	67.98	134.98	140	3.7
9	7.66	356.79	373	4.5	41	38.30	282.53	296	4.8	73	68.94	129.39	134	3.6
10	8.62	355.94	372	4.5	42	39.26	278.76	292	4.7	74	69.89	123.76	128	3.4
11	9.57	354.99	371	4.5	43	40.21	274.91	288	4.8	75	70.85	118.09	122	3.3
12	10.53	353.94	370	4.5	44	41.17	270.99	284	4.8	76	71.81	112.39	116	3.2
13	11.49	352.79	369	4.6	45	42.13	266.99	280	4.9	77	72.77	106.66	110	3.1
14	12.45	351.54	368	4.7	46	43.09	262.92	276	5.0	78	73.72	100.90	104	3.1
15	13.40	350.19	367	4.8	47	44.04	258.78	272	5.1	79	74.68	95.11	98	3.0
16	14.36	348.75	366	4.9	48	45.00	254.56	267	4.9	80	75.64	89.30	92	3.0
17	15.32	347.21	365	5.1	49	45.96	250.27	262	4.7	81	76.60	83.46	86	3.0
18	16.28	345.57	363	5.0	50	46.91	245.91	257	4.5	82	77.55	77.59	80	3.1
19	17.23	343.84	361	5.0	51	47.87	241.48	252	4.4	83	78.51	71.71	74	3.2
20	18.19	342.01	359	5.0	52	48.83	236.99	247	4.2	84	79.47	65.80	68	3.3
21	19.15	340.08	357	5.0	53	49.79	232.43	242	4.1	85	80.43	59.88	62	3.5
22	20.11	338.06	355	5.0	54	50.74	227.80	237	4.0	86	81.38	53.94	56	3.8
23	21.06	335.95	353	5.1	55	51.70	223.11	232	4.0	87	82.34	47.98	50	4.2
24	22.02	333.74	350	4.9	56	52.66	218.36	227	4.0	88	83.30	42.01	44	4.7
25	22.98	331.43	347	4.7	57	53.62	213.54	222	4.0	89	84.26	36.03	38	5.5
26	23.94	329.04	344	4.5	58	54.57	208.67	217	4.0	90	85.21	30.04	32	6.5
27	24.89	326.55	341	4.4	59	55.53	203.74	212	4.1	91	86.17	24.05	26	8.1
28	25.85	323.97	338	4.3	60	56.49	198.75	207	4.1	92	87.13	18.04	19	5.3
29	26.81	321.31	335	4.3	61	57.45	193.71	202	4.3	93	88.09	12.03	13	8.1
30	27.77	318.55	332	4.2	62	58.40	188.61	197	4.4	94	89.04	6.02	7	16.4
31	28.72	315.70	329	4.2	63	59.36	183.46	192	4.7	95	89.90	0.63	1	-
32	29.68	312.77	326	4.2	64	60.32	178.26	187	4.9					

A7 CALIBRATION VERIFICATION MATRIX - URD

5.2.1 Req.– *Cat I: CCD read noise - doit* .... ⊔

Required accuracy, constraints:

Readout noise less than 5e$^-$.
Variation in readout noise w.r.t. reference value less than 0.5e$^-$.

5.2.2 Req.– *Cat I: Hot pixels* .... ⊔

Required accuracy, constraints:

Number of hot pixels to be determined by experience/lab values.
The total number of bad pixels (hot pixels + cold pixels) is less than 80000 (checked in **req. 535** *Cold pixels*
Difference in number of hot pixels w.r.t. reference value, less than 100.

5.2.3 Req.– *Cat I: CCD gain* .... ⊔

Required accuracy, constraints:

Accuracy: In units of e$^-$/ADU, from lab values or found empirically. Variation in time less than 1%.

5.2.4 Req.– *Cat III: Electromagnetic Compatibility* .... ⊔

Required accuracy, constraints:

Difference between read noise under operational conditions and the standard read noise measurement should be smaller than 20% for external and 10% for internal causes of interference.

5.2.5 Req.– *Cat III: Electrical cross talk* .... ⊔

Required accuracy, constraints:

10^{-5}

5.3.1 Req.– *Cat I: CCD Dark Current - doit* .... ⊔

Required accuracy, constraints:

Dark count rate should be less than the equivalent of 3 e$^-$/pixel/hour in ADUs excluding bad pixels.
Accuracy of determining particle event rate 1 ADU/cm^2/hour.
Particle event rates should be identical for each chip.

5.3.2 Req.– *Cat I: CCD Particle Event Rate* .... ☐

Required accuracy, constraints:

better than 1 ADU/cm^2/hour

5.3.3 Req.– *Cat I: CCD Linearity* .... ☐

Required accuracy, constraints:

better than 1% on the photometric scale

5.3.4 Req.– *Cat III: CCD Charge Transfer Efficiency* .... ☐

Required accuracy, constraints:

CTE $>$ 0.999995 per parallel or serial shift.

5.3.5 Req.– *Cat I: CCD Cold Pixels* .... ☐

Required accuracy, constraints:

Quality Check: Number of hot pixels to be determined by experience/lab values. The total number of bad pixels (hot pixels + cold pixels) is less than 80000. Difference in number of cold pixels w.r.t. reference version less than 100.

5.3.6 Req.– *Cat III: CCD Hysteresis, strong signal* .... ☐

5.4.1 Req.– *Cat I: Bias - doit* .... ☐

Required accuracy, constraints:

The required accuracy per pixel in the master bias frame is "nominal read-outnoise/$\sqrt{10}$".

For the quality check: Since an overscan correction is performed, the deviation of the mean level of master bias (bias level) from zero, should be less than TBD.

5.4.2 Req.– *Cat I: Flat-field - dome key bands + user bands - doit* .... ☐

Required accuracy, constraints:

Accurately measuring pixel-to-pixel gain variations as small as 1%. Re-insertion of the filter shall not alter the flat field stucture by more than 0.3% (rms, measured over the full detector area).

5.4.3 Req.– *Cat I: Flat-field - twilight* .... ☐

Required accuracy, constraints:

Mean levels should be approximately 20000 ADU.

5.4.4 Req.– *Cat I: Flat-field - night sky* . . . . ⊔

Required accuracy, constraints:

This procedure would benefit from a prior detection and masking of bright objects. If no masks of bright objects are available, then a minimum of 15 frames should be included.

5.4.5 Req.– *Cat I: Flat-field - fringing* . . . . ⊔

5.4.6 Req.– *Cat I: Flat-field - master flat* . . . . ⊔

5.4.7 Req.– *Cat I: Quick detector responsivity check-doit* . . . . ⊔

Required accuracy, constraints:

1%

5.4.8 Req.– *Cat II: Illumination correction* . . . . ⊔

Required accuracy, constraints:

better than 1% for the amplitude over a single CCD.

5.5.1 Req.– *Cat III: Position of Camera in focal plane* . . . . ⊔

Required accuracy, constraints:

Internal precision: 0.3 pixel. External precision limited by reference catalog

5.5.2 Req.– *Cat III: Telescope Pointing and offsetting* . . . . ⊔

Required accuracy, constraints:

1 arc second

5.5.3 Req.– *Cat III: Telescope and Field Rotator tracking* . . . . ⊔

Required accuracy, constraints:

VST requirements:
free tracking better than 0.2 arcsec r.m.s.
autoguiding tracking better 0.05 arcsec

5.5.4 Req.– *Cat III: PSF Anisotropy* . . . . ⊔

Required accuracy, constraints:

better than 1%

5.5.5 Req.– *Cat I: The astrometric solution for templates - doit -see 6.3.4* . . . . . . ⊔

5.5.6 Req.– *Cat III: The astrometric solution for Guide CCD's* . . . . ⊔

Required accuracy, constraints:

1 arcsec rms for the accuracy with respect to the external standard;

External precision is driven by the position reference catalog. This is in the case of the USNO-A2 catalog of the order 0.3" with possible systematic excursions to 1".

5.6.1 Req.– *Cat III: Shutter Timing* . . . . ⊔

Required accuracy, constraints:

Timing error less than 0.2%.

5.6.2 Req.– *Cat I: Photometric Calibration - monitoring* . . . . ⊔

Required accuracy, constraints:

all photometry better than 1-2% on the photometric scale.

5.6.3 Req.– *Cat I: Photometric Calibration - zeropoint keybands - doit* . . . . ⊔

Required accuracy, constraints:

1% on the photometric scale

5.6.4 Req.– *Cat I: Photometric Calibration - zeropoint user bands* . . . . ⊔

Required accuracy, constraints:

2% on the photometric scale for broad bands and 5% for narrow band filters.

5.6.5 Req.– *Cat I: Filter band passes - user bands vs key bands* . . . . ⊔

Required accuracy, constraints:

10% on the photometric scale (formal spec) and 2% (goal) for broad band filters
1% for composite -> key

5.6.6 Req.– *Cat III: Dependency on angle - ADC, rotator/reproducibility* . . . . ⊔

Required accuracy, constraints:

1% on the photometric scale

5.6.7 Req.– *Cat III: Linearity (as a function of flux)* . . . . ⊔

Required accuracy, constraints:

Better than 1% on the photometric scale

5.6.8 Req.– *Cat III: Detection limit and ETC calibrations* . . . . ⊔

Required accuracy, constraints:

10 % in detection limit

5.6.9 Req.– *Cat I: Secondary Standards* . . . . ⊔

Required accuracy, constraints:

0.02 mag in individual secondary standards stars. The sources in **CalFile– 569** *s*hould cover a FoV of approximately 1.2 square degree.

5.7.1 Req.– *Cat III: Camera focus/tilt* . . . . ⊔

5.7.2 Req.– *Cat III: Ghosts - ADC* . . . . ⊔