EUROPEAN SOUTHERN OBSERVATORY

Organisation Européenne pour des Recherches Astronomiques dans l'Hémisphère Austral

Europäische Organisation für astronomische Forschung in der südlichen Hemisphäre

VERY LARGE TELESCOPE

VIMOS Pipeline User's Guide

VLT-MAN-ESO-19500-3355

Issue 1.0

Date 2004-07-09

Prepared: ESO VIMOS Pipeline Team 2004-07-09 ....
Name Date Signature

Approved: M. Peron ....
Name Date Signature

Released: P. Quinn ....
Name Date Signature

This page was intentionally left blank

Change record

Issue/Rev.	Date	Section/Parag. affected	Reason/Initiation/Documents/Remarks
1.0	09/07/2004	All	First version

This page was intentionally left blank

Contents

1 Introduction

1.1 Purpose

The VIMOS pipeline is a subsystem of the *VLT Data Flow System* (DFS). Its target user is ESO *Data Flow Operations* (DFO) in the generation of master calibration data, in the reduction of scientific exposures, and in the data quality control. It should also serve as a quick look tool for *Paranal Science Operations* (PSO). Additionally, the VIMOS pipeline recipes are made public to the user community, to allow a more personalised precessing of the data from the instrument.

This manual is a complete description of the data reduction recipes used by the the VIMOS pipeline.

This document reflects the status of the VIMOS pipeline as of July 9, 2004 (version 1.3.1).

1.2 Acknowledgements

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

Valuable suggestions on the IFU data reduction pipeline were provided by Eric Emsellem and Arlette Rousset-Pecontal (Centre de Recherche Astronomique de Lyon), and by Martin Roth (Astrophysikalisches Institut Potsdam).

The feedback we received in numerous discussions with our "beta-testers", Martino Romaniello (ESO Data Management Division), Markus Kissler-Patig (ESO Instrumentation Division), and Harald Kuntschner (ST-ECF), was very much appreciated.

Useful advice has been received especially from Sandro D'Odorico (ESO Instrumentation Division).

Gianni Marconi, Stephane Brillant, and Stefano Bagnulo (ESO Paranal Observatory) for the good collaboration and the constant support.

In particular we want to thank Paola Sartoretti (ESO, Data Management Division), who was a continuous source of useful ideas for improving the pipeline recipes, and for their extensive testing.

1.3 Reference documents

[1]	VLT Data Flow System Operations Model for VLT/VLTI Instrumentation	VLT-PLA-ESO-19000-1183
[2]	VLT Data Flow System Specifications for Pipeline and Quality Control	VLT-SPE-ESO-19600-1233
[3]	Data Flow for VLT instruments Requirement Specification	VLT-SPE-ESO-19000-1618
[4]	DFS Pipeline & Quality Control – User Manual	VLT-MAN-ESO-19500-1619
[5]	ESO DICB – Data Interface Control Document	GEN-SPE-ESO-00000-0794
[6]	Common Pipeline Library User Manual	VLT-MAN-ESO-19500-2720
[7]	Gasgano User's Manual	VLT-PRO-ESO-19000-1932

[8] E.Bertin and S.Arnouts (1996), A&A Supp. v.117 p. 393-404

[9] VMPIP User's Guide (*In preparation*)

[10] Douglas J. Mink (1997), Astronomical Data Analysis Software and Systems VI, A.S.P. Conference Series, Vol. 125, p. 249-252

[11] Ronald C. Stone, Jeffrey R. Pier, and David G. Monet (1999), Astron. J., Vol. 118, p.2488

[12] L.Mortara and A.Fowler (1981), Solid State Imagers for Astronomy, in SPIE Vol. 290, p. 28.

[13] E.W.Greisen and M.R.Calabretta (2002), Astronomy & Astrophysics, 395, 1061-1075.

1.4 Abbreviations and acronyms

ADF	Aperture Definition File
ADM	Aperture Definition in mm
CPL	Common Pipeline Library
CRV	Spectral curvature model
DFS	Data Flow System
DMD	Data Management Division
DO	Data Organiser
DRS	Data Reduction System
ESO–MIDAS	ESO's Munich Image Data Analysis System
FITS	Flexible Image Transport System
HR	High Resolution
ICS	Instrument Control Software
IDS	Inverse Dispersion Solution
IRAF	Image Reduction and Analysis Facility
IWS	Instrument WorkStation
LR	Low Resolution
MMU	Mask Manufacturing Unit
MOS	Multi Object Spectroscopy
MR	Medium Resolution
OPT	VIMOS optical distortion model
PAF	VLT PArameter File
PWS	Pipeline WorkStation
QC	Quality Control
RB	Reduction Block
RBS	Reduction Block Scheduler
SAO	Smithsonian Astrophysical Observatory
SOF	Set Of Frames
TCS	Telescope Control Software
UT	Unit Telescope
VIMOS	VIsible Multi-Object Spectrograph
VLT	Very Large Telescope
VMMPS	VIMOS Mask Preparation Software
VMPIP	VIMOS pipeline IWS package

WCS	World Coordinate System

2 Overview

The VIMOS pipeline is a collection of data reduction recipes administrated by the higher level programs of the *VLT Data Flow System* (DFS) [1].

The *VLT Data Flow System* provides an environment where the definition and the execution of the data reduction steps can be made entirely automatic. The fundamental DFS software components making up the required functionality are the *Data Organiser* (DO) and the *Reduction Block Scheduler* (RBS). The DO applies some defined instrument-dependent rules for the classification of the incoming frames, for their association to suitable calibration data, and for launching the appropriate recipe for their standard reduction. The product of the DO is the *Reduction Block* (RB), an ASCII file containing all the necessary information for the specific data reduction. The RB is executed by the RBS, which calls the assigned pipeline recipe with the appropriate parameters and makes the data reduction products available to the *Data Handling Server* (DHS).

The VIMOS pipeline recipes can also be used outside the pipeline context, as stand-alone programs. In this way they may be easily used by astronomers at their home institutes, or taken as building blocks for different data reduction strategies, as in the case of the *VIMOS Pipeline IWS Package* (VMPIP) [9]. However, the pipeline recipes, being designed to operate within the DFS environment, do not repeat the checks already performed by the DO. In other words, data classification and appropriate association are not proofed in any way within a recipe, being taken for granted. The *Gasgano* ([7]) data browser, that is capable of classifying the data frames in the same way as the DO does, greatly assist in the administration of a large quantity of different data frames: VIMOS is a relatively complex instrument, and scientific data coming from different quadrants and produced by different instrument modes should not risk misclassification and/or association.

The VIMOS instrument and the different types of VIMOS raw frames are briefly described in Sections 3 and 4, while the usage of the available reduction recipes is presented in Section 5.

More detailed descriptions of the data reduction algorithms used by the individual pipeline recipes can be found in Sections 6 and 7.

In Appendix A the installation of the VIMOS pipeline recipes is described.

3 The VIMOS instrument

VIMOS has been developed under ESO contract by the VIRMOS Consortium, headed by the Laboratoire d'Astrophysique de Marseille.

The instrument has been made available to the community and started operations in Paranal on April 1^{st}, 2003.

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

3.1 Overview

VIMOS is aimed at survey-type programs with emphasis on large object samples rather than individual objects. VIMOS is designed for Wide Field Imaging (14' x 16') and extremely high Multi Object Spectroscopy capability (up to several hundred slits). In addition, it has a unique *Integral Field Unit* (IFU) providing a field-of-view up to 1 arc minute at 0.67"/fibre in low resolution spectroscopy.

The field-of-view is split in four identical channels. Field lenses provide a corrected telescope focal plane where flat masks are inserted in MOS mode. For the IFU instrument mode a special mask bearing the IFU pseudo-slits is used. Pupil relay lenses, folding mirrors and collimators direct the light to the four cameras. Grisms are inserted in front of the cameras in spectroscopic mode. The detectors are four 2k x 4k EEV CCDs with pixel size 15 μ (see `http://www.eso.org/projects/odt/Vimos/vimos.html` for detector design and performance reports of the four VIMOS CCD systems).

3.2 Direct imaging

The field-of-view consists of 4 quadrants of 7' x 8' each separated by a cross 2' wide, with a sampling of 0.205"/pixel.

The available filters, U, B, V, R, I, and z, are close to the Mould definition, and allow to minimise the colour terms to transform to the Johnson system.

The filter transmission curves are available from `http://www.eso.org/instruments/vimos`.

3.3 Multi-Object-Spectroscopy (MOS)

The multi-object mode of VIMOS uses grisms and masks. ESO distributes the *VIMOS Mask Preparation Software* (VMMPS), a package developed by the VIRMOS Consortium for slit definition and positioning on a preliminary exposure on the sky region to be observed. The user can define rectangular, curved or inclined slits of widths larger than 0.4"[1]. The masks are laser cut in INVAR plates on Paranal with the *Mask Manufacturing Unit* (MMU). The instrument cabinet has a capacity of 15 masks, some of which are meant for maintenance and calibration purposes[2].

There are 6 grisms available, all operating in first order. Their spectral characteristics are given in Table 3.3.1.

[1] The VIMOS pipeline data reduction software does not support yet curved or inclined slits.

[2] For technical reasons only 7 slots of the masks cabinet are currently used.

Grism	**Filter**	λ_c (Å)	λ **range** (Å)	**R**	**Dispersion** (Å/pixel)
LR_red	OS_red	7500	5500 - 9500	210 - 260	7.3
LR_blue	OS_blue	4800	3700 - 6700	180 - 220	5.3
MR	GG435	7000	5000 - 10000	580 - 720	2.5
HR_red	GG475	7400	5650 - 9500	2500 - 3100	0.6
HR_orange	GG435	6310	4550 - 8400	2150 - 2650	0.6
HR_blue	none	5100	3650 - 6900	2050 - 2550	0.5

Table 3.3.1: *VIMOS grisms.* λ_c *is the zero deviation (or central) wavelength, and* R *is the spectral resolution for a 1" MOS slit, corresponding to* $\sim$ 0.8 *IFU fibre. The spectral ranges are given with the specified filter in. The HR_red grism is not available for quadrant 4, and is replaced in this case with a HR_orange grism. The transmission curves for the four grism/filter units are available at* `http://www.eso.org/instruments/vimos`.

With LR grisms, a spectrum will typically span less than 600 pixels along the dispersion direction. This allows a multiplexing factor of 5, *i.e.*, to stack up to five spectra along the dispersion direction, provided that there are enough well spaced targets in the field-of-view.

With MR grisms, a spectrum will span about 2000 pixels when used with the GG435 filter. It is therefore possible to stack up to 2 spectra along the dispersion direction, provided that the slits are positioned at the very edges of the imaging field-of-view.

With HR grisms the spectra extend beyond the detector length, therefore multiplexing is impossible and the observable spectral interval depends on the position of a slit on the mask, spanning about 2400 Å for the HR_red and HR_orange grisms, and about 2000 Å for the HR_blue grism.

A further constraint on the slit positions comes from the presence of the 0^{th}, -1^{st} and 2^{nd} grism diffraction orders. At low spectral resolution, a dim second order spectrum at twice the spectral resolution would be included in the CCD in the case of slits located in the lower (*i.e.*, bluer) regions of the mask. This spectrum would likely contaminate the multiplexed first order spectra on the red side of the CCD. Similarly, a mirrored -1^{st} order spectrum at the same resolution of the 1^{st} order spectrum and with about 1/6 of its luminosity, would be included in the CCD in the case of slits from the highest (*i.e.*, redder) regions of the mask. This spectrum would likely contaminate the multiplexed first order spectra on the blue side of the CCD (see an illustration of -1^{st} contamination on page 96). For this reason multiplexed slits are constrained to be identical, and to have the same position along the cross-dispersion direction: in the assumption of negligible spectral curvatures in all orders, the 0^{th}, 2^{nd} and -1^{st} contaminations would then be removed by the sky subtraction procedure.

Aside from the above considerations, the number of slits that can be accommodated in one mask obviously depends on the target density. Simulations on real fields using VMMPS show that above a density of about $7 \cdot 10^4/\mathrm{degree}^2$ it is possible to define masks with up to 160 10" long slits/quadrant. This number drops to 125 slits/quadrant at a density of about $4 \cdot 10^4/\mathrm{degree}^2$.

3.4 Integral Field Unit (IFU)

The VIMOS IFU is the largest ever made for such an application. It consists of 6400 (80 x 80) fibres, coupled to microlenses. The field-of-view is square, with a continuous spatial sampling (the dead space between fibres is below 10% of the fibre-to-fibre distance). At the entrance of the IFU there is a focal elongator providing two spatial samplings of 0.33"/fibre and 0.67"/fibre.

The fibres are split into 16 bundles of 400 fibres each. Each instrument quadrant receives 4 bundles that are arranged along 4 parallel pseudo-slits providing 4 multiplexed series of 400 spectra each.

The field-of-view is modified according to the used spectral resolution. At low spectral resolution the field is respectively 54" x 54" with 0.67"/fibre, and 27" x 27" with 0.33"/fibre, 80 fibres on a side. All the pseudo-slits are illuminated, and the multiplexed spectra belonging to different pseudo-slits would contaminate each other in some measure. For instance, the second order spectra of a bright object on pseudo-slit 2 of quadrant 2 would contaminate the spectra on pseudo-slits 3 and 4, creating obvious ghosts in the corresponding regions of the reconstructed field-of-view (see Figures 3.5.2 and 3.5.3, pages 18–19).

At medium and high resolution just the 4 central bundles on the IFU head are illuminated (see Figure 3.5.3, page 19). Only one pseudo-slit per quadrant is used, since the spectra span the whole detector and multiplexing is impossible. The field-of-view is therefore 4 times smaller, *i.e.*, 27" x 27" with 0.67"/fibre, and 13" x 13" with 0.33"/fibre, 40 fibres on a side.

The fibre-to-fibre distance at detector level is about 5.0 pixels, while the fibre profile FWHM is about 3.2 pixels. The spectral resolution is approximately 1.25 times the spectral resolution corresponding to a 1" slit in MOS mode (see Table 3.3.1). The spectral coverage is identical to the MOS case for LR and MR grisms. For HR grisms the situation is different because the spectral range is too large to be contained on the CCD, and since the central slit-of-fibres is shifted by about 140 pixels from the chip centre in (spectrally) opposite directions depending on the instrument quadrant, the usable spectral range is reduced by about 160 Å leading to Table 3.4.1.

Grism	**λ range (Å)**
HR_red	6350 - 8600
HR_orange	5250 - 7550
HR_blue	4200 - 6150

Table 3.4.1: *VIMOS IFU usable spectral range in high spectral resolution mode.*

3.5 IFU components numbering scheme

The conventions used in the VIMOS IFU pipeline recipes to indicate IFU fibers, IFU masks and pseudo-slits are described in this section.

IFU masks: VIMOS has 4 IFU masks. They are counted as the VIMOS quadrants to which they correspond, *i.e.*, counterclockwise, with the same convention used in the cartesian plane (see Figure 3.5.1).

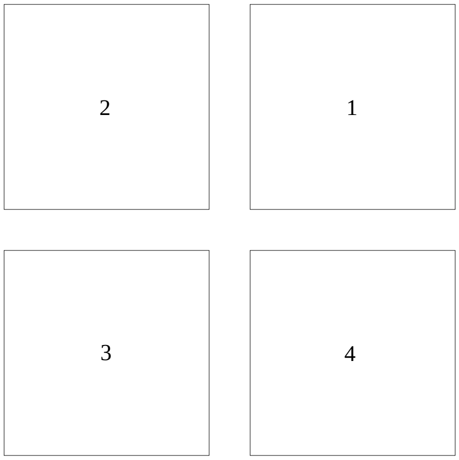


Figure 3.5.1: *Counting VIMOS quadrants.*

In spectral mode, blue is down and red is up in all quadrants.

IFU pseudo-slits: Each VIMOS mask hosts 4 IFU pseudo-slits, numbered from 1 to 4. The pseudo-slit 1 is the one that is somewhat more separated from the other ones (see Figure 3.5.2).

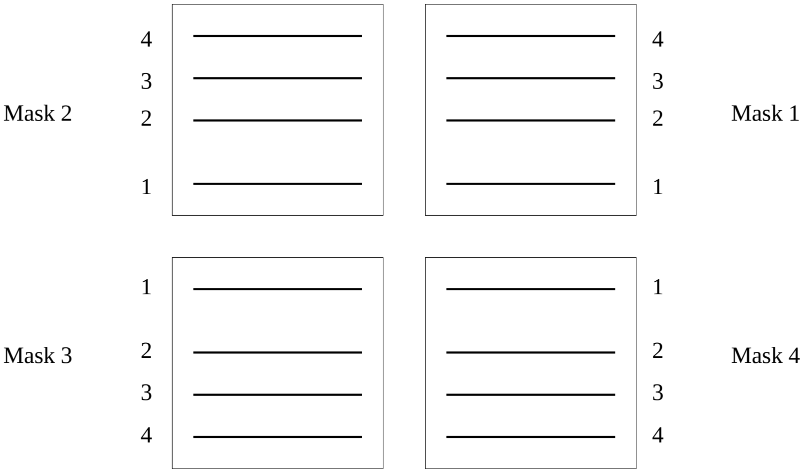
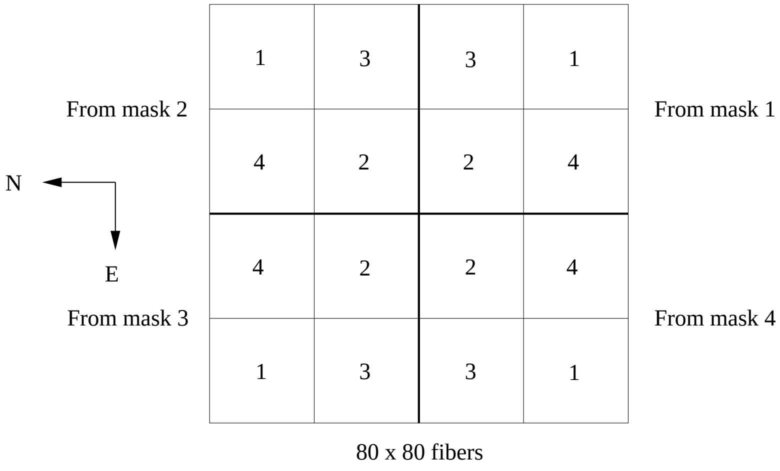


Figure 3.5.2: *Counting IFU pseudo-slits.*

IFU fibers: Each IFU pseudo-slit hosts 400 fibers, divided into 5 blocks of 80 fibers each. The fibers are counted from 1 to 400, always starting from the left.

IFU head: Each pseudo-slit corresponds to a 20x20 region of the 80x80 IFU head (see Figure 3.5.3).



1	3	3	1
4	2	2	4
4	2	2	4
1	3	3	1

Figure 3.5.3: *IFU head. The number of the corresponding pseudo-slit is indicated within each 20x20 fiber module.*

North is to the left, and East is down. The exact spatial position for each individual fiber is listed in the IFU tables (see Table 5.24.3, page 103).

Illuminated pseudo-slits: In LR observations all the pseudo-slits are illuminated (multiplexing). In MR and HR observations, just the central pseudo-slits (numbered 2) are used.

4 VIMOS data

VIMOS data can be separated into *raw* frames and *product* frames. Raw frames are the unprocessed output of the VIMOS instrument observations, while product frames are either the result of the VIMOS pipeline processing (as reduced frames, master calibration frames, etc.), or are outsourced (as standard stars catalogs, lists of grism characteristics, etc.).

Any raw or product frame can be classified on the basis of a set of keywords read from its header. Data classification is typically carried out by the DO or by *Gasgano* [7], that apply the same set of classification rules. The association of a raw frame with calibration data (*e.g.*, of a science frame with a master bias frame) can be obtained by matching the values of a different set of header keywords.

Each kind of raw frame is typically associated to a single VIMOS pipeline recipe, *i.e.*, the recipe assigned to the reduction of that specific frame type. In the pipeline environment this recipe would be launched automatically. In some cases two recipes are assigned, one meant for the reduction of a single frame of that type, and the other for the reduction of a *stack* of frames of the same type, as happens in the case of jittered science observations.

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

In the following all raw and product VIMOS data frames are listed, together with the keywords used for their classification and correct association. The indicated *DO category* is a label assigned to any data type after it has been classified, which is then used to identify the frames listed in the *Set of Frames* (see Section 5.2, page 40).

4.1 Raw frames

Raw frames can be distinguished in *general* frames, *direct imaging* frames, *MOS* frames and *IFU* frames. Their intended use is implicitly defined by the assigned recipe.

4.1.1 General frames

These are data that are in principle independent of the instrument mode (direct imaging, MOS, or IFU), as is the case for bias and dark exposures. The keyword `ESO INS MODE` is set accordingly to 'IMG' for direct imaging frames, and to 'MOS' for any calibration associated to spectroscopy (either MOS or IFU), to indicate the intended use for the data.

- **Bias:**

  DO category: `BIAS`
  Processed by: `vmbias`

Classification keywords:	Association keywords:	Note:
`DPR CATG = CALIB`	`INS MODE`	Instrument mode
`DPR TYPE = BIAS`	`OCS CON QUAD`	Quadrant used
`DPR TECH = IMAGE`	`DET CHIP1 ID`	Chip identification

	DET WIN1 NY	No of pixels in y
	DET WIN1 BINX	Binning along X
	DET WIN1 BINY	Binning along Y
	DET READ MODE	Readout method
	DET READ SPEED	Readout speed
	DET READ CLOCK	Readout clock pattern

- **Dark current:**

  DO category: DARK
  Processed by: vmdark

Classification keywords:	Association keywords:	Note:
DPR CATG = CALIB	INS MODE	Instrument mode
DPR TYPE = DARK	OCS CON QUAD	Quadrant used
DPR TECH = IMAGE	DET CHIP1 ID	Chip identification
	DET WIN1 NY	No of pixels in y
	DET WIN1 BINX	Binning along X
	DET WIN1 BINY	Binning along Y
	DET READ MODE	Readout method
	DET READ SPEED	Readout speed
	DET READ CLOCK	Readout clock pattern

- **Screen flat field for gain determination and bad pixels detection:**

  DO category: DETECTOR_PROPERTIES
  Processed by: vmdet

Classification keywords:	Association keywords:	Note:
DPR CATG = CALIB	INS MODE	Instrument mode
DPR TYPE = FLAT,LAMP	OCS CON QUAD	Quadrant used
DPR TECH = IMAGE or MOS	DET CHIP1 ID	Chip identification
TPL ID = VIMOS_img_tec_DetLin	DET WIN1 NY	No of pixels in y
or VIMOS_mos_tec_DetLin	DET WIN1 BINX	Binning along X
	DET WIN1 BINY	Binning along Y
	DET READ MODE	Readout method
	DET READ SPEED	Readout speed
	DET READ CLOCK	Readout clock pattern

4.1.2 Direct imaging frames

The direct imaging mode is used to record signal without using any grism.

- **Exposure of calibration mask:**

  DO category: `MASK_TO_CCD`
  Processed by: `vmmasktoccd`

Classification keywords:	Association keywords:	Note:
`DPR CATG = CALIB`	`INS MODE`	Instrument mode
`DPR TYPE = OTHER,LAMP`	`OCS CON QUAD`	Quadrant used
`DPR TECH = IMAGE`	`INS ADF ID`	ADF file ID
`TPL ID = VIMOS_img_tec_MaskToCcd`	`INS FILT[1-4] ID`	Filter ID on each beam
	`DET CHIP1 ID`	Chip identification
	`DET WIN1 NY`	No of pixels in y
	`DET WIN1 BINX`	Binning along X
	`DET WIN1 BINY`	Binning along Y
	`DET READ MODE`	Readout method
	`DET READ SPEED`	Readout speed
	`DET READ CLOCK`	Readout clock pattern

- **Preimaging for MOS mask preparation:**

  DO category: `IMG_PREIMAGING`
  Processed by: `vmimpreimaging`

Classification keywords:	Association keywords:	Note:
`DPR CATG = SCIENCE`	`INS MODE`	Instrument mode
`DPR TYPE = OBJECT`	`OCS CON QUAD`	Quadrant used
`DPR TECH = IMAGE,PRE`	`INS FILT[1-4] ID`	Filter ID on each beam
	`DET CHIP1 ID`	Chip identification
	`DET WIN1 NY`	No of pixels in y
	`DET WIN1 BINX`	Binning along X
	`DET WIN1 BINY`	Binning along Y
	`DET READ MODE`	Readout method
	`DET READ SPEED`	Readout speed
	`DET READ CLOCK`	Readout clock pattern

- **Twilight flat field:**

  DO category: `IMG_SKY_FLAT`
  Processed by: `vmimflatsky`

Classification keywords:	Association keywords:	Note:
`DPR CATG = CALIB`	`INS MODE`	Instrument mode
`DPR TYPE = FLAT,SKY`	`OCS CON QUAD`	Quadrant used
`DPR TECH = IMAGE`	`INS FILT[1-4] ID`	Filter ID on each beam
	`DET CHIP1 ID`	Chip identification
	`DET WIN1 NY`	No of pixels in y
	`DET WIN1 BINX`	Binning along X
	`DET WIN1 BINY`	Binning along Y
	`DET READ MODE`	Readout method
	`DET READ SPEED`	Readout speed
	`DET READ CLOCK`	Readout clock pattern

- **Screen flat field:**

  DO category: `IMG_SCREEN_FLAT`
  Processed by: `vmimflatscreen`

Classification keywords:	Association keywords:	Note:
`DPR CATG = CALIB`	`INS MODE`	Instrument mode
`DPR TYPE = FLAT,LAMP`	`OCS CON QUAD`	Quadrant used
`DPR TECH = IMAGE`	`INS FILT[1-4] ID`	Filter ID on each beam
`TPL ID = VIMOS_img_cal_ScreenFlat`	`INS LAMP[1-5] ID`	Calib. lamps ID
	`INS LAMP[1-5] STATE`	Lamp state
	`DET CHIP1 ID`	Chip identification
	`DET WIN1 NY`	No of pixels in y
	`DET WIN1 BINX`	Binning along X
	`DET WIN1 BINY`	Binning along Y
	`DET READ MODE`	Readout method
	`DET READ SPEED`	Readout speed
	`DET READ CLOCK`	Readout clock pattern

- **Standard stars field:**

  DO category: `IMG_STANDARD`
  Processed by: `vmimstandard`

Classification keywords:	Association keywords:	Note:
`DPR CATG = CALIB`	`INS MODE`	Instrument mode
`DPR TYPE = STD`	`OCS CON QUAD`	Quadrant used
`DPR TECH = IMAGE`	`INS FILT[1-4] ID`	Filter [1-4] on each beam
	`DET CHIP1 ID`	Chip identification
	`DET WIN1 NY`	No of pixels in y
	`DET WIN1 BINX`	Binning along X
	`DET WIN1 BINY`	Binning along Y

	DET READ MODE	Readout method
	DET READ SPEED	Readout speed
	DET READ CLOCK	Readout clock pattern

- **Astrometric stars field:**

  DO category: `IMG_ASTROMETRY`
  Processed by: `vmskyccd`

Classification keywords:	Association keywords:	Note:
DPR CATG = CALIB	INS MODE	Instrument mode
DPR TYPE = ASTROMETRY	OCS CON QUAD	Quadrant used
DPR TECH = IMAGE	OBS TARG NAME	Astrometric field used
	INS FILT[1-4] ID	Filter ID on each beam
	DET CHIP1 ID	Chip identification
	DET WIN1 NY	No of pixels in y
	DET WIN1 BINX	Binning along X
	DET WIN1 BINY	Binning along Y
	DET READ MODE	Readout method
	DET READ SPEED	Readout speed
	DET READ CLOCK	Readout clock pattern

- **Scientific observation:**

  DO category: `IMG_SCIENCE`
  Processed by: `vmimobsstare, vmimobsjitter`

Classification keywords:	Association keywords:	Note:
DPR CATG = SCIENCE	INS MODE	Instrument mode
DPR TYPE = OBJECT	OCS CON QUAD	Quadrant used
DPR TECH = IMAGE	INS FILT[1-4] ID	Filter ID on each beam
	DET CHIP1 ID	Chip identification
	DET WIN1 NY	No of pixels in y
	DET WIN1 BINX	Binning along X
	DET WIN1 BINY	Binning along Y
	DET READ MODE	Readout method
	DET READ SPEED	Readout speed
	DET READ CLOCK	Readout clock pattern

4.1.3 MOS frames

The Multi-Object Spectroscopy mode is used to obtain simultaneous spectra from several objects in the field-of-view.

- **Screen flat field:**

  DO category: `MOS_SCREEN_FLAT`
  Processed by: `vmspflat`

Classification keywords:	Association keywords:	Note:
`DPR CATG = CALIB`	`INS MODE`	Instrument mode
`DPR TYPE = FLAT,LAMP`	`OCS CON QUAD`	Quadrant used
`DPR TECH = MOS`	`INS FILT[1-4] NAME`	Filter name on each beam
`TPL ID = VIMOS_mos_cal_ScreenFlat`	`INS GRIS[1-4] ID`	Grism ID on each beam
or `VIMOS_mos_tec_Startr`	`INS MASK[1-4] ID`	Mask ID on each beam
	`INS MSHU[1-4] MODE`	Mask shutter mode
	`INS LAMP[1-5] ID`	Calib. lamps ID
	`INS LAMP[1-5] STATE`	Lamp state
	`DET CHIP1 ID`	Chip identification
	`DET WIN1 NY`	No of pixels in y
	`DET WIN1 BINX`	Binning along X
	`DET WIN1 BINY`	Binning along Y
	`DET READ MODE`	Readout method
	`DET READ SPEED`	Readout speed
	`DET READ CLOCK`	Readout clock pattern

- **Arc lamp spectra:**

  DO category: `MOS_ARC_SPECTRUM`
  Processed by: `vmspcaldisp`

Classification keywords:	Association keywords:	Note:
`DPR CATG = CALIB`	`INS MODE`	Instrument mode
`DPR TYPE = WAVE,LAMP`	`OCS CON QUAD`	Quadrant used
`DPR TECH = MOS`	`INS FILT[1-4] NAME`	Filter name on each beam
	`INS GRIS[1-4] ID`	Grism ID on each beam
	`INS MASK[1-4] ID`	Mask ID on each beam
	`INS MSHU[1-4] MODE`	Mask shutter mode
	`INS LAMP[1-5] ID`	Calib. lamps ID
	`INS LAMP[1-5] STATE`	Lamp state
	`DET CHIP1 ID`	Chip identification
	`DET WIN1 NY`	No of pixels in y
	`DET WIN1 BINX`	Binning along X
	`DET WIN1 BINY`	Binning along Y

	DET READ MODE	Readout method
	DET READ SPEED	Readout speed
	DET READ CLOCK	Readout clock pattern

- **Standard star spectrum:**

  DO category: `MOS_STANDARD`
  Processed by: *not yet implemented*

Classification keywords:	Association keywords:	Note:
DPR CATG = CALIB	INS MODE	Instrument mode
DPR TYPE = STD	OCS CON QUAD	Quadrant used
DPR TECH = MOS	INS FILT[1-4] NAME	Filter name on each beam
	INS GRIS[1-4] ID	Grism ID on each beam
	INS MASK[1-4] ID	Mask ID on each beam
	INS MSHU[1-4] MODE	Mask shutter mode
	DET CHIP1 ID	Chip identification
	DET WIN1 NY	No of pixels in y
	DET WIN1 BINX	Binning along X
	DET WIN1 BINY	Binning along Y
	DET READ MODE	Readout method
	DET READ SPEED	Readout speed
	DET READ CLOCK	Readout clock pattern

- **Scientific observation:**

  DO category: `MOS_SCIENCE`
  Processed by: `vmmosobsstare, vmmosobsjitter`

Classification keywords:	Association keywords:	Note:
DPR CATG = SCIENCE	INS MODE	Instrument mode
DPR TYPE = OBJECT	OCS CON QUAD	Quadrant used
DPR TECH = MOS	INS FILT[1-4] NAME	Filter name on each beam
	INS GRIS[1-4] ID	Grism ID on each beam
	INS MASK[1-4] ID	Mask ID on each beam
	INS MSHU[1-4] MODE	Mask shutter mode
	DET CHIP1 ID	Chip identification
	DET WIN1 NY	No of pixels in y
	DET WIN1 BINX	Binning along X
	DET WIN1 BINY	Binning along Y
	DET READ MODE	Readout method
	DET READ SPEED	Readout speed
	DET READ CLOCK	Readout clock pattern

4.1.4 IFU frames

The IFU mode is used to get in a spatially continuous way simultaneous spectra from a selected sky region.

- **Screen flat field:**

  DO category: `IFU_SCREEN_FLAT`
  Processed by: `vmifucalib`

Classification keywords:	Association keywords:	Note:
`DPR CATG = CALIB`	`INS MODE`	Instrument mode
`DPR TYPE = FLAT,LAMP`	`OCS CON QUAD`	Quadrant used
`DPR TECH = IFU`	`INS FILT[1-4] NAME`	Filter name on each beam
	`INS GRIS[1-4] ID`	Grism ID on each beam
	`INS LAMP[1-5] ID`	Calib. lamps ID
	`INS LAMP[1-5] STATE`	Lamp state
	`INS IFUE MAG`	IFU magnification
	`INS IFUS MODE`	IFU shutter mode
	`DET CHIP1 ID`	Chip identification
	`DET WIN1 NY`	No of pixels in y
	`DET WIN1 BINX`	Binning along X
	`DET WIN1 BINY`	Binning along Y
	`DET READ MODE`	Readout method
	`DET READ SPEED`	Readout speed
	`DET READ CLOCK`	Readout clock pattern

- **Arc lamp spectra:**

  DO category: `IFU_ARC_SPECTRUM`
  Processed by: `vmifucalib`

Classification keywords:	Association keywords:	Note:
`DPR CATG = CALIB`	`INS MODE`	Instrument mode
`DPR TYPE = WAVE,LAMP`	`OCS CON QUAD`	Quadrant used
`DPR TECH = IFU`	`INS FILT[1-4] NAME`	Filter name on each beam
	`INS GRIS[1-4] ID`	Grism ID on each beam
	`INS LAMP[1-5] ID`	Calib. lamps ID
	`INS LAMP[1-5] STATE`	Lamp state
	`INS IFUE MAG`	IFU magnification
	`INS IFUS MODE`	IFU shutter mode
	`DET CHIP1 ID`	Chip identification
	`DET WIN1 NY`	No of pixels in y
	`DET WIN1 BINX`	Binning along X
	`DET WIN1 BINY`	Binning along Y
	`DET READ MODE`	Readout method

	`DET READ SPEED`	Readout speed
	`DET READ CLOCK`	Readout clock pattern

- **Standard star spectra:**

  DO category: `IFU_STANDARD`
  Processed by: `vmifustandard`

Classification keywords:	Association keywords:	Note:
`DPR CATG = CALIB`	`INS MODE`	Instrument mode
`DPR TYPE = STD`	`OCS CON QUAD`	Quadrant used
`DPR TECH = IFU`	`INS FILT[1-4] NAME`	Filter name on each beam
	`INS GRIS[1-4] ID`	Grism ID on each beam
	`INS IFUE MAG`	IFU magnification
	`INS IFUS MODE`	IFU shutter mode
	`DET CHIP1 ID`	Chip identification
	`DET WIN1 NY`	No of pixels in y
	`DET WIN1 BINX`	Binning along X
	`DET WIN1 BINY`	Binning along Y
	`DET READ MODE`	Readout method
	`DET READ SPEED`	Readout speed
	`DET READ CLOCK`	Readout clock pattern

- **Scientific observation:**

  DO category: `IFU_SCIENCE`
  Processed by: `vmifuscience`

Classification keywords:	Association keywords:	Note:
`DPR CATG = SCIENCE`	`INS MODE`	Instrument mode
`DPR TYPE = OBJECT`	`OCS CON QUAD`	Quadrant used
`DPR TECH = IFU`	`INS FILT[1-4] NAME`	Filter name on each beam
	`INS GRIS[1-4] ID`	Grism ID on each beam
	`INS IFUE MAG`	IFU magnification
	`INS IFUS MODE`	IFU shutter mode
	`DET CHIP1 ID`	Chip identification
	`DET WIN1 NY`	No of pixels in y
	`DET WIN1 BINX`	Binning along X
	`DET WIN1 BINY`	Binning along Y
	`DET READ MODE`	Readout method
	`DET READ SPEED`	Readout speed
	`DET READ CLOCK`	Readout clock pattern

4.2 Product frames

As with raw frames, product frames can be distinguished in *general* frames, *direct imaging* frames, *MOS* frames, and *IFU* frames. These frames are classified by the DO of by *Gasgano* [7] according to their own DO category (keyword `PRO CATG`), assigned to them at creation time. For this reason no classification keyword is listed in this section. The name of the recipe (or recipes) used to create a given product is given here.

4.2.1 General frames

- **Master bias:**

  DO category: `MASTER_BIAS`
  Created by: `vmbias`

Association keywords:	Note:
`OCS CON QUAD`	Quadrant used
`DET CHIP1 ID`	Chip identification
`DET WIN1 NY`	No of pixels in y
`DET WIN1 BINX`	Binning along X
`DET WIN1 BINY`	Binning along Y
`DET READ MODE`	Readout method
`DET READ SPEED`	Readout speed
`DET READ CLOCK`	Readout clock pattern

- **Master dark:**

  DO category: `MASTER_DARK`
  Created by: `vmdark`

Association keywords:	Note:
`OCS CON QUAD`	Quadrant used
`DET CHIP1 ID`	Chip identification
`DET WIN1 NY`	No of pixels in y
`DET WIN1 BINX`	Binning along X
`DET WIN1 BINY`	Binning along Y
`DET READ MODE`	Readout method
`DET READ SPEED`	Readout speed
`DET READ CLOCK`	Readout clock pattern

- **List of bad pixels positions:**

  DO category: `CCD_TABLE`
  Created by: `vmdet`

Association keywords:	Note:
`OCS CON QUAD`	Instrument quadrant
`DET CHIP1 ID`	Chip identification
`DET WIN1 NY`	No of pixels in y
`DET WIN1 BINX`	Binning along X
`DET WIN1 BINY`	Binning along Y
`DET READ MODE`	Readout method
`DET READ SPEED`	Readout speed
`DET READ CLOCK`	Readout clock pattern

4.2.2 Direct imaging frames

- **Master sky flat field:**

  DO category: `IMG_MASTER_SKY_FLAT`
  Created by: `vmimflatsky`

Association keywords:	Note:
`OCS CON QUAD`	Quadrant used
`INS FILT[1-4] ID`	Filter ID for beam 1 to 4
`DET CHIP1 ID`	Chip identification
`DET WIN1 NY`	No of pixels in y
`DET WIN1 BINX`	Binning along X
`DET WIN1 BINY`	Binning along Y
`DET READ MODE`	Readout method
`DET READ SPEED`	Readout speed
`DET READ CLOCK`	Readout clock pattern

- **Master screen flat field:**

  DO category: `IMG_MASTER_SCREEN_FLAT`
  Created by: `vmimflatscreen`

Association keywords:	Note:
`OCS CON QUAD`	Quadrant used
`INS FILT[1-4] ID`	Filter ID for beam 1 to 4
`DET CHIP1 ID`	Chip identification
`DET WIN1 NY`	No of pixels in y
`DET WIN1 BINX`	Binning along X
`DET WIN1 BINY`	Binning along Y
`DET READ MODE`	Readout method
`DET READ SPEED`	Readout speed
`DET READ CLOCK`	Readout clock pattern

- **Combined screen flat field:**

  DO category: `IMG_COMBINED_SCREEN_FLAT`
  Created by: `vmimflatscreen`

  No association rules are defined for a combined screen flat field. This dataset is not used by any pipeline recipe, and is only created for data quality control purposes.

- **Reduced scientific observation:**

  DO category: `IMG_SCIENCE_REDUCED`
  Created by: `vmimpreimaging, vmimobsstare, vmimobsjitter`

Association keywords:	Note:
`OCS CON QUAD`	Quadrant used
`INS FILT[1-4] ID`	Filter ID for beam 1 to 4
`DET CHIP1 ID`	Chip identification
`DET WIN1 NY`	No of pixels in y
`DET WIN1 BINX`	Binning along X
`DET WIN1 BINY`	Binning along Y

- **Reduced standard stars field:**

  DO category: `IMG_STANDARD_REDUCED`
  Created by: `vmimstandard`

Association keywords:	Note:
`OCS CON QUAD`	Quadrant used
`INS FILT[1-4] ID`	Filter ID for beam 1 to 4
`DET CHIP1 ID`	Chip identification
`DET WIN1 NY`	No of pixels in y
`DET WIN1 BINX`	Binning along X
`DET WIN1 BINY`	Binning along Y

- **List of detected sources:**

  DO category: `IMG_GALAXY_TABLE`
  Created by: `vmimobsstare, vmimobsjitter, vmimstandard`

Association keywords:	Note:
`OCS CON QUAD`	Quadrant used
`INS FILT[1-4] ID`	Filter ID for beam 1 to 4

- **List of identified stars:**

  DO category: `IMG_STAR_MATCH_TABLE`
  Created by: `vmimstandard`

Association keywords:	Note:
`OCS CON QUAD`	Quadrant used
`INS FILT[1-4] ID`	Filter ID for beam 1 to 4

- **Astrometric catalog:**

  DO category: `ASTROMETRIC_TABLE`
  Created by: *external*

Association keywords:	Note:
`OBS TARG NAME`	Astrometric field name

- **Photometric catalog:**

  DO category: `PHOTOMETRIC_CATALOG`
  Created by: *external*

  No association keyword required.

- **Coefficients for photometric calibration:**

  DO category: `PHOTOMETRIC_TABLE`
  Created by: `vmimcalphot`

Association keywords:	Note:
`OCS CON QUAD`	Quadrant used
`INS FILT[1-4] ID`	Filter ID for beam 1 to 4

4.2.3 MOS frames

- **Master screen flat field:**

  DO category: `MOS_MASTER_SCREEN_FLAT`
  Created by: `vmspflat`

Association keywords:	Note:
`OCS CON QUAD`	Quadrant used
`INS FILT[1-4] NAME`	Filter name for beam 1 to 4
`INS GRIS[1-4] ID`	Grism ID on each beam
`INS MASK[1-4] ID`	Mask ID for beam 1 to 4
`INS MSHU[1-4] MODE`	Mask shutter mode for beam 1 to 4
`DET NAME`	Name of detector
`DET CHIP1 ID`	Chip identification
`DET WIN1 NY`	No of pixels in y
`DET WIN1 BINX`	Binning along X
`DET WIN1 BINY`	Binning along Y

- **Combined screen flat field:**

  DO category: `MOS_COMBINED_SCREEN_FLAT`
  Created by: `vmspflat`

Association keywords:	Note:
`OCS CON QUAD`	Quadrant used
`INS FILT[1-4] NAME`	Filter name for beam 1 to 4
`INS GRIS[1-4] ID`	Grism ID on each beam
`INS MASK[1-4] ID`	Mask ID for beam 1 to 4
`INS MSHU[1-4] MODE`	Mask shutter mode for beam 1 to 4
`DET NAME`	Name of detector
`DET CHIP1 ID`	Chip identification
`DET WIN1 NY`	No of pixels in y
`DET WIN1 BINX`	Binning along X
`DET WIN1 BINY`	Binning along Y

- **Calibration lamp lines catalog:**

  DO category: `LINE_CATALOG`
  Created by: *external*

Association keywords:	Note:
`OCS CON QUAD`	Quadrant used
`INS GRIS[1-4] ID`	Grism ID for beam 1 to 4

- **Grism dependent parameters:**

  DO category: `GRISM_TABLE`
  Created by: *external*

Association keywords:	Note:
`OCS CON QUAD`	Instrument quadrant
`INS FILT[1-4] NAME`	Filter name for beam 1 to 4
`INS GRIS[1-4] ID`	Grism ID for beam 1 to 4

- **Spectral extraction parameters:**

  DO category: `EXTRACT_TABLE`
  Created by: `vmspcaldisp`

  No association rules are defined for an extraction table. This dataset is not used within the on-line pipeline process. The user should ensure that the correct extraction table is specified when calling the recipes *vmmosobsstare* and *vmmosobsjitter*.

- **Extracted spectra:**

  DO category: `MOS_SCIENCE_REDUCED`
  Created by: `vmmosobsstare, vmmosobsjitter`

Association keywords:	Note:
`OCS CON QUAD`	Quadrant used
`INS FILT[1-4] NAME`	Filter name for beam 1 to 4
`INS GRIS[1-4] ID`	Grism ID on each beam
`INS MASK[1-4] ID`	Mask ID for beam 1 to 4
`INS MSHU[1-4] MODE`	Mask shutter mode for beam 1 to 4
`DET CHIP1 ID`	Chip identification
`DET WIN1 NY`	No of pixels in y
`DET WIN1 BINX`	Binning along X
`DET WIN1 BINY`	Binning along Y

- **Position of detected objects on extracted slit spectra:**

  DO category: `WINDOW_TABLE`
  Created by: `vmmosobsstare, vmmosobsjitter`

  No association rules are defined for a window table, because this dataset is not used by any pipeline recipe.

- **Identification of extracted spectra:**

  DO category: `OBJECT_TABLE`
  Created by: `vmmosobsstare, vmmosobsjitter`

  No association rules are defined for an object table, because this dataset is not used by any pipeline recipe.

- **Extracted standard star spectrum:**

  DO category: `MOS_STANDARD_REDUCED`
  Created by: *not yet implemented*

Association keywords:	Note:
`OCS CON QUAD`	Quadrant used
`INS FILT[1-4] NAME`	Filter name for beam 1 to 4
`INS GRIS[1-4] ID`	Grism ID on each beam
`INS MASK[1-4] ID`	Mask ID for beam 1 to 4
`INS MSHU[1-4] MODE`	Mask shutter mode for beam 1 to 4
`DET CHIP1 ID`	Chip identification
`DET WIN1 NY`	No of pixels in y
`DET WIN1 BINX`	Binning along X
`DET WIN1 BINY`	Binning along Y

- **Spectro-photometric standard star fluxes:**

  DO category: `SPECTRO_PHOTOMETRIC_STANDARD`
  Created by: *external*

Association keywords:	Note:
`OBJECT`	Object name

- **Spectral response:**

  DO category: `MOS_RESPONSE_FUNCTION`
  Created by: *not yet implemented*

Association keywords:	Note:
`OCS CON QUAD`	Quadrant used
`INS FILT[1-4] NAME`	Filter name for beam 1 to 4
`INS GRIS[1-4] ID`	Grism ID on each beam
`DET CHIP1 ID`	Chip identification

4.2.4 IFU frames

- **Master screen flat field:**

  DO category: `IFU_MASTER_SCREEN_FLAT`
  Created by: `vmifucalib`

Association keywords:	Note:
`OCS CON QUAD`	Quadrant used
`INS IFUE MAG`	IFU magnification
`INS IFUS MODE`	IFU shutter mode
`INS FILT[1-4] NAME`	Filter name for beam 1 to 4
`INS GRIS[1-4] ID`	Grism ID on each beam
`DET NAME`	Name of detector
`DET CHIP1 ID`	Chip identification
`DET WIN1 NY`	No of pixels in y
`DET WIN1 BINX`	Binning along X
`DET WIN1 BINY`	Binning along Y

- **Spectral response:**

  DO category: `IFU_RESPONSE_FUNCTION`
  Created by: *not yet implemented*

Association keywords:	Note:
`OCS CON QUAD`	Quadrant used
`INS IFUE MAG`	IFU magnification
`INS FILT[1-4] NAME`	Filter name for beam 1 to 4
`INS GRIS[1-4] ID`	Grism ID on each beam
`DET CHIP1 ID`	Chip identification

- **Reconstructed field-of-view:**

  DO category: `IFU_FOV`
  Created by: `vmifuscience, vmifustandard`

Association keywords:	Note:
`OCS CON QUAD`	Quadrant used
`INS IFUE MAG`	IFU magnification
`INS FILT[1-4] NAME`	Filter name for beam 1 to 4
`INS GRIS[1-4] ID`	Grism ID on each beam
`DET CHIP1 ID`	Chip identification

- **Sky spectrum:**

  DO category: `IFU_SCIENCE_SKY`
  Created by: `vmifustandard`

Association keywords:	Note:
`OCS CON QUAD`	Quadrant used
`INS IFUE MAG`	IFU magnification
`INS FILT[1-4] NAME`	Filter name for beam 1 to 4
`INS GRIS[1-4] ID`	Grism ID on each beam
`DET CHIP1 ID`	Chip identification

- **Standard star spectrum:**

  DO category: `IFU_STANDARD_EXTRACTED`
  Created by: `vmifustandard`

Association keywords:	Note:
`OCS CON QUAD`	Quadrant used
`INS IFUE MAG`	IFU magnification
`INS FILT[1-4] NAME`	Filter name for beam 1 to 4
`INS GRIS[1-4] ID`	Grism ID on each beam
`DET CHIP1 ID`	Chip identification

- **Reduced standard star fiber spectra:**

  DO category: `IFU_STANDARD_REDUCED`
  Created by: `vmifustandard`

Association keywords:	Note:
`OCS CON QUAD`	Quadrant used
`INS IFUE MAG`	IFU magnification

`INS FILT[1-4] NAME`	Filter name for beam 1 to 4
`INS GRIS[1-4] ID`	Grism ID on each beam
`DET CHIP1 ID`	Chip identification

- **Reduced science spectra:**

  DO category: `IFU_SCIENCE_REDUCED`
  Created by: `vmifuscience`

Association keywords:	Note:
`OCS CON QUAD`	Quadrant used
`INS IFUE MAG`	IFU magnification
`INS FILT[1-4] NAME`	Filter name for beam 1 to 4
`INS GRIS[1-4] ID`	Grism ID on each beam
`DET CHIP1 ID`	Chip identification

- **IFU fiber identification file:**

  DO category: `IFU_IDENT`
  Created by: *external*

Association keywords:	Note:
`OCS CON QUAD`	Quadrant used
`INS GRIS[1-4] ID`	Grism ID on each beam

- **IFU wavelength calibration:**

  DO category: `IFU_IDS`
  Created by: `vmifucalib`

Association keywords:	Note:
`OCS CON QUAD`	Quadrant used
`INS FILT[1-4] NAME`	Filter name for beam 1 to 4
`INS GRIS[1-4] ID`	Grism ID on each beam

- **IFU extraction mask:**

  DO category: `IFU_TRACE`
  Created by: `vmifucalib`

Association keywords:	Note:
`OCS CON QUAD`	Quadrant used
`INS FILT[1-4] NAME`	Filter name for beam 1 to 4
`INS GRIS[1-4] ID`	Grism ID on each beam

- **IFU relative transmission factors:**

  DO category: `IFU_TRANSMISSION`
  Created by: `vmifucalib`

Association keywords:	Note:
`OCS CON QUAD`	Quadrant used
`INS GRIS[1-4] ID`	Grism ID on each beam

5 Cookbook

In this section the usage of the VIMOS pipeline recipes is described. In particular, typical data reduction sessions for each instrument operating mode are presented in Sections 5.5, 5.6, and 5.7.

5.1 VIMOS pipeline recipes

The current VIMOS pipeline is based on a set of 20 stand-alone recipes, assigned to different fundamental operations:

Creation of general calibration data:

vmdet: creating a bad pixel table, and determining CCD gain and read-out-noise from a sequence of flat fields at different exposure levels.

vmbias: creating a master bias from a sequence of raw bias frames.

vmdark: creating a master dark from a sequence of raw dark frames.

Creation of direct imaging calibration data:

vmimflatscreen: creating a master screen flat field from a sequence of screen flat field frames.

vmimflatsky: creating a master sky flat field from a sequence of sky flat field frames.

vmmasktoccd: computing transformation between mask and CCD coordinates from an exposure with a pinhole mask.

vmskyccd: computing the distortions of the Sky to CCD transformation from an exposure on a field of astrometric stars.

Direct imaging flux calibration:

vmimstandard: reducing a photometric standard stars field image, and determining the frame magnitude zeropoint.

vmimcalphot: determining the mean zeropoint and, optionally, the extinction coefficient and the colour term from any number of star match tables produced by the *vmimstandard* recipe.

Direct imaging data reduction:

vmimpreimaging: reducing a preimaging exposure for the preparation of a MOS mask.

vmimobsstare: reducing a science exposure.

vmimobsjitter: reducing a stack of jittered science exposures.

Creation of MOS calibration data:

vmspflat: producing a spectral master flat field.

vmspcaldisp: computing the optical distortion, the spectral curvature, and the inverse dispersion models.

MOS data reduction:

vmmosobsstare: reducing a science exposure.

vmmosobsjitter: reducing a stack of jittered science exposures.

Creation of IFU calibration data:

vmifucalib: producing an extraction mask, a wavelength calibration, and a fiber-to-fiber relative transmission correction.

vmifustandard: extracting the total standard star spectrum.

IFU data reduction:

vmifuscience: reducing a science exposure.

vmifucombine: compose an image of the field-of-view.

In the next sections a general description on the use of recipes is given, together with more detailed information on the individual recipes.

5.2 Set of frames

Each pipeline recipe is run on a set of input FITS data files. The filenames are listed together with their DO category in an ASCII file, the *Set of Frames* (SOF), that is required when launching a recipe. SOF files containing the frames selected by the user are automatically created by *Gasgano* [7].

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

```
VIMOS.2003-02-04T16:31:53.040.fits MOS_ARC_SPECTRUM
/cal/vimos/mos/cal/mbias.3.fits MASTER_BIAS
/cal/vimos/mos/cal/lcat_LR_red.3.tfits LINE_CATALOG
/cal/vimos/mos/cal/grs_LR_red.3.tfits GRISM_TABLE
```

The pipeline recipe will access the listed files when required by the reduction algorithm.

Note that the VIMOS pipeline recipes do not proof in any way the correctness of the classification tags specified in the SOF. In the above example, the recipe *vmspcaldisp* will treat the frame `VIMOS.2003-02...` as a MOS_ARC_SPECTRUM, the frame `/cal/vimos/mos/cal/mbias.3.fits` as a MASTER_BIAS, etc., even when they do not contain this type of data. The recipe will also assume that all frames are associated correctly, *i.e.*, that they all come from the same instrument quadrant, using the same filter, grism, etc., and that the appropriate calibration files have been specified.

The reason of this lack of control is that the VIMOS recipes are just the DRS component of the complete pipeline running on Paranal, where the task of data classification and association is carried out by the Data Organiser. On the other side, using *Gasgano* [7] as an interface to the pipeline recipes will permit to classify the data frames exactly as the DO does.

A recipe handling an incorrect SOF may stop with not immediately understandable error messages at best. In the worst cases, the recipe would apparently run without any problem, producing results that may look reasonable while they are instead flawed.

5.3 Recipe configuration

Each pipeline recipe may be assigned a configuration file, containing the default values of the parameters related to that recipe. The possibility to create recipe configuration files, and the way information is conventionally stored in such files, depends on the front-end application used for launching the pipeline recipes[3]. In this section the recipe configuration files created and handled by *EsoRex* are described. Please refer to the Gasgano User's Manual [7] for information about saving recipe configuration parameters.

An *EsoRex* recipe configuration file is normally generated in the directory `$HOME/.esorex` by *EsoRex* (see Section 5.4, on page 41) at installation time, and has the same name as the recipe to which it is related, with the filename extension `.rc`. For instance, the recipe *vmbias* has its *EsoRex* generated configuration file named `vmbias.rc`.

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

```
# --StackMethod
# Stacking method to apply. ( Average | Median | MinMax | Ksigma | Auto )
vimos.Parameters.stacking.method=Average
```

In this example, the parameter `vimos.Parameters.stacking.method` is set to the value `Average`. In the configuration file generated by *EsoRex*, one or more comment lines are added containing information about the possible values of the parameter, and an alias that could be used as a command line option (see again Section 5.4 on page 41).

The hierarchy introduced in the parameter names has currently no effect. Although it is not used at the moment, it is envisaged that this feature will be used in future releases to avoid potential name clashes. The shorter parameter aliases are made available for use on the command line.

The parameter names belonging to the recipe specific configuration files are described in the corresponding recipe sections.

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

5.4 Running a recipe using EsoRex

The VIMOS pipeline recipes are implemented as plugins, used by front-end applications like *EsoRex* or *Gasgano* [7] to execute the recipes.

The file organiser *Gasgano* [7] provides an intuitive GUI for launching the recipes, applying the same rules used by the DO for the available data frames classification, whereas *EsoRex* is the most basic way for executing the pipeline recipes from the shell prompt, that shall be described here.

[3] The plugin concept allows the implementation of different front-end applications with specific functionalities. For a complete description on how to create a pipeline recipe launcher, please refer to the CPL User Manual [6].

The basic format for using *EsoRex* is as follows:

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

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

```
esorex --recipes
```

To get help for an individual recipe (in the subsequent examples, *vmbias* is used), the following is used:

```
esorex --help vmbias
```

In order to display the current parameters setting of a recipe, either the recipe configuration file located under `$HOME/.esorex` may be viewed (if present), or the following command may be used:

```
esorex --params vmbias
```

If the default recipe configuration file is not found, or a particular value is not configured within this file, then the system defaults will be shown and used.

A recipe can be run by specifying its name to *EsoRex*, together with the name of a SOF (see Section 5.2, page 40). For instance, the following command line would be used to run the recipe *vmbias* for processing the files specified in the SOF `vmbias.sof`:

```
esorex vmbias vmbias.sof
```

A recipe configuration file different from the default one (see Section 5.3, page 41) can also be specified on the command line:

```
esorex --recipe-config=my_alternative_config.rc vmbias vmbias.sof
```

The recipe parameters can be modifyed either by editing directly the used configuration file, or by specifying new parameter values on the command line using the command line options defined for this purpose. Such command line options should be inserted after the recipe name, and before the SOF name. For instance, to set the *vmbias* recipe `StackMethod` parameter to `Median`, the following should be typed:

```
esorex vmbias --StackMethod=Median vmbias.sof
```

There are also parameters used to configure the *EsoRex* launcher, that may be listed in an `esorex.rc` configuration file located under `$HOME/.esorex`. On the command line, the *EsoRex* options must be inserted before, and not after, the specified recipe name. The *EsoRex* options are those that are recipe independent, as for instance the verbosity level, the directory where the recipe products should be written, or the permission to overwrite old products with new ones.

Here are some more examples of running a recipe:

```
esorex --output-prefix=test vmmasktoccd --CleanCosmics=true test.sof
esorex --verbose=debug vmskyccd --SExtractor.Window=1,1,1900,2300 skyccd
esorex vmspflat --ComputeQC=false --FlatMethod=Median in.sof
esorex --time=true vmspcaldisp --CleanBadPixel=false input1.sof
```

In the *vmmasktoccd* example the cosmic ray cleaning is switched on, and the prefix `test_` is prepended to output products names. The input SOF is the file `test.sof`.

In the *vmskyccd* example the verbosity level is set to `debug`, so that all messages are displayed, including the debug ones. In addition to that, SExtractor operations are restricted to the specified image region. The input SOF is a file named `skyccd`.

In the *vmspflat* example the computation of quality control parameters is turned off, and the flat field trend removal (used in the flat field normalisation) is performed using a median filter. The input SOF is `in.sof`.

Finally, in the *vmspcaldisp* example the execution of the recipe is timed and bad pixels cleaning is turned off. The input SOF is `input1.sof`.

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

5.5 Example of imaging data reduction

A simple, typical imaging data reduction procedure is described here. It is assumed that the following data are available:

One scientific exposure:

```
      VIMOS.2004-09-25T09:27:15.336.fits     IMG_SCIENCE
```

One standard star field exposure:

```
      VIMOS.2004-09-25T08:20:10.006.fits     IMG_STANDARD
```

Five bias exposures:

```
      VIMOS.2004-09-25T08:00:27.821.fits     BIAS
      VIMOS.2004-09-25T08:01:05.604.fits     BIAS
      VIMOS.2004-09-25T08:01:44.091.fits     BIAS
      VIMOS.2004-09-25T08:02:22.070.fits     BIAS
      VIMOS.2004-09-25T08:03:01.042.fits     BIAS
```

Four twilight flat fields:

```
    VIMOS.2004-09-25T10:00:25.956.fits     IMG_SKY_FLAT
    VIMOS.2004-09-25T10:01:09.806.fits     IMG_SKY_FLAT
    VIMOS.2004-09-25T10:01:51.246.fits     IMG_SKY_FLAT
    VIMOS.2004-09-25T10:03:13.433.fits     IMG_SKY_FLAT
```

Three dome flat fields:

```
    VIMOS.2004-09-25T13:57:07.351.fits     IMG_SCREEN_FLAT
    VIMOS.2004-09-25T13:57:52.041.fits     IMG_SCREEN_FLAT
    VIMOS.2004-09-25T13:58:36.842.fits     IMG_SCREEN_FLAT
```

All the listed data are meant to belong to the same VIMOS quadrant.

In the following, it is also assumed for simplicity that, in the *EsoRex* configuration file, the flag `suppress-prefix` is set to `TRUE`, so that the product file names will just be identical to their product categories, with an extension `.fits` for images, and `.tfits` for tables. Moreover, it is assumed that all the handled files (inputs and products) are located in the current directory. The only exception is represented by the standard calibration tables (*e.g.*, star catalogues), that here are assumed to be located under `/cal/vimos/ima`.

The procedure is as follows:

First, a master bias is created with the recipe *vmbias* (see Section 5.9, page 53). The following set-of-frames may be prepared:

File: *bias.sof*

```
    VIMOS.2004-09-25T08:00:27.821.fits     BIAS
    VIMOS.2004-09-25T08:01:05.604.fits     BIAS
    VIMOS.2004-09-25T08:01:44.091.fits     BIAS
    VIMOS.2004-09-25T08:02:22.070.fits     BIAS
    VIMOS.2004-09-25T08:03:01.042.fits     BIAS
```

The following command line can be given at the shell prompt:

```
  esorex vmbias --StackMethod=Average bias.sof
```

The file `master_bias.fits`, containing the bias master calibration obtained from the input exposures, is created.

For creating the master flat field calibration, the recipe *vmimflatsky* (see Section 5.12, page 63) is applied to the four input twilight flat field exposures. The SOF may be prepared as follows:

File: *sky.sof*

```
VIMOS.2004-09-25T10:00:25.956.fits     IMG_SKY_FLAT
VIMOS.2004-09-25T10:01:09.806.fits     IMG_SKY_FLAT
VIMOS.2004-09-25T10:01:51.246.fits     IMG_SKY_FLAT
VIMOS.2004-09-25T10:03:13.433.fits     IMG_SKY_FLAT
master_bias.fits                       MASTER_BIAS
```

where the newly created master bias calibration is used.

The command line:

```
esorex vmimflatsky --StackMethod=Median sky.sof
```

will create the file `img_master_sky_flat.fits`, that can be used for the flat field correction of the scientific exposure.

Alternatively, a master sky flat field calibration may be created by combining the dome flat field exposures, carrying more accurate information about the high frequency fixed pattern noise, with the sky flat field exposures, carrying more reliable information on the large scale trends of the CCD illumination. For processing the raw screen flat field exposures, the recipe *vmimflatscreen* is used (see Section 5.11, page 59), and the SOF may be prepared as follows:

File: *dome.sof*

```
VIMOS.2004-09-25T13:57:07.351.fits     IMG_SCREEN_FLAT
VIMOS.2004-09-25T13:57:52.041.fits     IMG_SCREEN_FLAT
VIMOS.2004-09-25T13:58:36.842.fits     IMG_SCREEN_FLAT
master_bias.fits                       MASTER_BIAS
```

The command line:

```
esorex vmimflatscreen dome.sof
```

will create the two files `img_master_screen_flat.fits` and `img_combined_screen_flat.fits`. The master flat field will be added to the set-of-frames of the *vmimflatsky* recipe:

File: *sky.sof*

```
VIMOS.2004-09-25T10:00:25.956.fits     IMG_SKY_FLAT
VIMOS.2004-09-25T10:01:09.806.fits     IMG_SKY_FLAT
VIMOS.2004-09-25T10:01:51.246.fits     IMG_SKY_FLAT
VIMOS.2004-09-25T10:03:13.433.fits     IMG_SKY_FLAT
master_bias.fits                       MASTER_BIAS
img_combined_screen_flat.fits          IMG_MASTER_SCREEN_FLAT
```

This will be called as before:

```
esorex vmimflatsky --StackMethod=Median sky.sof
```

With the master bias and the master flat field it is now possible to reduce the scientific observation. The set-of-frames may be defined as follows:

File: *science.sof*

```
VIMOS.2004-09-25T09:27:15.336.fits     IMG_SCIENCE
master_bias.fits                       MASTER_BIAS
img_master_sky_flat.fits               IMG_MASTER_SKY_FLAT
/cal/vimos/ima/cal/badpixel.1.tfits    CCD_TABLE
/cal/vimos/ima/cal/ipc_R.1.tfits       PHOTOMETRIC_TABLE
```

A bad-pixel table is specified in the set-of-frames, because in this example a bad-pixel correction will be requested when running the *vmimobsstare* recipe (see Section 5.18, page 80). The specified photometric table from the calibration directory tree carries the standard magnitude zeropoint for a given filter and a given instrument quadrant. This magnitude zeropoint will be copied to the header of the reduced image. Note that in this example it is assumed that the data belong to the 1st quadrant, and were obtained using the R filter.

The scientific exposure is then reduced by running:

```
esorex vmimobsstare --CleanBadPixel=TRUE science.sof
```

This will create the reduced scientific exposure, `img_science_reduced.fits`, and the list of detected objects, `img_galaxy_table.tfits`.

Alternatively, since a standard star field observation from the same night is available, a more reliable photometric table may be computed. For this purpose, the following set-of-frames may be created:

File: *standard.sof*

```
VIMOS.2004-09-25T08:20:10.006.fits     IMG_STANDARD
master_bias.fits                       MASTER_BIAS
img_master_sky_flat.fits               IMG_MASTER_SKY_FLAT
/cal/vimos/ima/cal/phstd_stetson.tfits PHOTOMETRIC_CATALOG
```

This is processed by the *vmimstandard* recipe (see Section 5.15, page 72):

```
esorex vmimstandard standard.sof
```

Three files are produced: the reduced image, `img_standard_reduced.fits`; the list of detected objects, `img_galaxy_table.tfits`; and most important `img_star_match_table.tfits`, the list of identified standard stars, that includes the differences between the catalog magnitude and the instrumental magnitude for each detected standard star.

The star match table is then included in the set-of-frames of the *vmimcalphot* recipe (see Section 5.16, page 77):

File: *phot.sof*

```
img_star_match_table.tfits              IMG_STAR_MATCH_TABLE
/cal/vimos/ima/cal/ipc_R.1.tfits        PHOTOMETRIC_TABLE
```

This is processed with:

```
esorex vmimcalphot phot.sof
```

generating a new photometric table, `photometric_table.tfits`, that will then replace the standard one in the `science.sof` set-of-frames shown above.

5.6 Example of MOS data reduction

A simple, typical MOS data reduction procedure is described here. It is assumed that the following data are available:

One scientific exposure:

```
VIMOS.2004-09-27T02:39:11.479.fits     MOS_SCIENCE
```

Five bias exposures:

```
VIMOS.2004-09-27T08:00:27.821.fits     BIAS
VIMOS.2004-09-27T08:01:05.604.fits     BIAS
VIMOS.2004-09-27T08:01:44.091.fits     BIAS
VIMOS.2004-09-27T08:02:22.070.fits     BIAS
VIMOS.2004-09-27T08:03:01.042.fits     BIAS
```

Three flat field exposures:

```
VIMOS.2004-09-27T18:59:03.641.fits     MOS_SCREEN_FLAT
VIMOS.2004-09-27T19:00:07.828.fits     MOS_SCREEN_FLAT
VIMOS.2004-09-27T19:01:14.252.fits     MOS_SCREEN_FLAT
```

One arc lamp exposure:

```
VIMOS.2004-09-27T19:33:03.631.fits     MOS_ARC_SPECTRUM
```

All the listed data are meant to belong to the same VIMOS quadrant.

In the following, it is also assumed for simplicity that, in the *EsoRex* configuration file, the flag `suppress-prefix` is set to `TRUE`, so that the product file names will just be identical to their product categories, with an extension `.fits` for images, and `.tfits` for tables. Moreover, it is assumed that all the handled files (inputs and products) are located in the current directory. The only exception is represented by the standard calibration tables (*e.g.*, line catalogues), that here are assumed to be located under `/cal/vimos/mos`.

The procedure is as follows:

First, a master bias is created with the recipe *vmbias*, exactly as described in Section 5.5, page 43.

The product `master_bias.fits` is used in the reduction of the flat field, arc lamp, and scientific exposures.

In order to process the available flat field exposures, the recipe *vmspflat* is used (see Section 5.20, page 87). The input SOF may be defined as follows:

File: *mosflat.sof*

```
      VIMOS.2004-09-27T18:59:03.641.fits     MOS_SCREEN_FLAT
      VIMOS.2004-09-27T19:00:07.828.fits     MOS_SCREEN_FLAT
      VIMOS.2004-09-27T19:01:14.252.fits     MOS_SCREEN_FLAT
      master_bias.fits                       MASTER_BIAS
      /cal/vimos/mos/cal/grs_LR_red.1.tfits  GRISM_TABLE
```

where it is assumed that the data belong to the 1st quadrant, and were obtained using the LR_red grism.

The following command line can be given at the shell prompt:

```
   esorex vmspflat --StackMethod=Average mosflat.sof
```

The products are the MOS master screen flat, `mos_master_screen_flat.fits`, and the non-normalised screen flat field, `mos_combined_screen_flat.fits`. The latter should be specified in the set-of-frames associated to the *vmspcaldisp* recipe (see Section 5.21, page 91), used to reduce the arc lamp exposure, and to determine the spectral distortions of the instrument:

File: *distortions.sof*

```
      VIMOS.2004-09-27T19:33:03.631.fits     MOS_ARC_SPECTRUM
      master_bias.fits                       MASTER_BIAS
      mos_combined_screen_flat.fits          MOS_COMBINED_SCREEN_FLAT
      /cal/vimos/mos/cal/grs_LR_red.1.tfits  GRISM_TABLE
      /cal/vimos/mos/cal/lcat_LR_red.1.tfits LINE_CATALOG
```

This is processed by:

```
   esorex vmspcaldisp distortions.sof
```

The products are the extraction table, `extract_table.tfits`, that will make possible to extract the scientific slit spectra according to the determined instrument distortions; and the image of the extracted arc lamp slit spectra, `mos_arc_spectrum_extracted.fits`, that is just used for quality control purposes.

The extraction table is specified in the last set-of-frames, the one associated to the recipe *vmmosobsstare* (see Section 5.22, page 97), used to reduce and extract the scientific spectra:

File: *objects.sof*

```
VIMOS.2004-09-27T02:39:11.479.fits     MOS_SCIENCE
master_bias.fits                       MASTER_BIAS
extract_table.tfits                    EXTRACT_TABLE
/cal/vimos/mos/cal/grs_LR_red.1.tfits  GRISM_TABLE
```

This is processed by:

```
esorex vmmosobsstare objects.sof
```

The products are the extracted object spectra, `mos_science_reduced.fits`; the sky subtracted slit spectra, `mos_science_extracted.fits`; the sky spectra, `mos_science_sky.fits`; and a couple of tables containing the identification of the extracted object spectra and their position along the slit, `object_table.tfits` and `window_table.tfits`.

5.7 Example of IFU data reduction

A simple, typical IFU data reduction procedure is described here. It is assumed that the following data are available:

One scientific exposure:

```
VIMOS.2004-04-27T04:15:24.227.fits     IFU_SCIENCE
```

Five bias exposures:

```
VIMOS.2004-04-27T08:00:27.821.fits     BIAS
VIMOS.2004-04-27T08:01:05.604.fits     BIAS
VIMOS.2004-04-27T08:01:44.091.fits     BIAS
VIMOS.2004-04-27T08:02:22.070.fits     BIAS
VIMOS.2004-04-27T08:03:01.042.fits     BIAS
```

Three flat field exposures:

```
VIMOS.2004-04-27T04:38:44.038.fits     IFU_SCREEN_FLAT
VIMOS.2004-04-27T04:40:32.250.fits     IFU_SCREEN_FLAT
VIMOS.2004-04-27T04:42:20.470.fits     IFU_SCREEN_FLAT
```

One arc lamp exposure:

```
VIMOS.2004-04-27T04:44:14.842.fits     IFU_ARC_SPECTRUM
```

All the listed data are meant to belong to the same VIMOS quadrant.

In the following, it is also assumed for simplicity that, in the *EsoRex* configuration file, the flag `suppress-prefix` is set to `TRUE`, so that the product file names will just be identical to their product categories, with an extension `.fits` for images, and `.tfits` for tables. Moreover, it is assumed that all the handled files (inputs and products) are located in the current directory. The only exception is represented by the standard calibration tables (*e.g.*, line catalogues), that here are assumed to be located under `/cal/vimos/mos`.

The procedure is as follows:

First, a master bias is created with the recipe *vmbias*, exactly as described in Section 5.5, page 43.

The product `master_bias.fits` is used in the reduction of the flat field, arc lamp, and scientific exposures.

The flat field and the arc lamp exposures are processed to determine the extraction mask, the wavelength calibration, and the fiber-to-fiber relative transmission correction. The input set-of-frames may be defined as follows:

File: *calib.sof*

```
VIMOS.2004-04-27T04:38:44.038.fits          IFU_SCREEN_FLAT
VIMOS.2004-04-27T04:40:32.250.fits          IFU_SCREEN_FLAT
VIMOS.2004-04-27T04:42:20.470.fits          IFU_SCREEN_FLAT
VIMOS.2004-04-27T04:44:14.842.fits          IFU_ARC_SPECTRUM
master_bias.fits                            MASTER_BIAS
/cal/vimos/mos/cal/lcat_HR_red.tfits        LINE_CATALOG
/cal/vimos/mos/cal/ifu_ident_HR_red.1.fits  IFU_IDENT
```

This will be processed with:

```
esorex vmifucalib calib.sof
```

For data quality control, the following products will be created: an image of the extracted and wavelength calibrated arc lamp spectra, `ifu_arc_spectrum_extracted.fits`; an image of the extracted and wavelength calibrated flat field spectra, `ifu_flat_spectrum_extracted.fits`; and an image obtained by the combination of all the raw input flat field exposures, `ifu_master_screen_flat.fits`. For the purpose of reducing the scientific data, also the following files are created: the extraction mask, `ifu_trace.tfits`; the wavelength calibration, `ifu_ids.tfits`; and the relative transmission factors, `ifu_transmission.tfits`.

These files are included in the set-of-frames prepared for the scientific data reduction process:

File: *ifuscience.sof*

```
VIMOS.2004-04-27T04:15:24.227.fits IFU_SCIENCE
master_bias.fits                   MASTER_BIAS
ifu_ids.tfits                      IFU_IDS
ifu_trace.tfits                    IFU_TRACE
ifu_transmission.tfits             IFU_TRANSMISSION
```

This will be processed with:

```
esorex vmifuscience ifuscience.sof
```

The following files will be created: an image containing the extracted, transmission corrected, and wavelength calibrated scientific spectra, `ifu_science_reduced.fits`, and an image of the reconstructed IFU field-of-view, `ifu_fov.fits`.

5.8 vmdet

The VIMOS pipeline recipe *vmdet* is used to estimate the *read-out-noise* (RON) and the gain of the CCD, and to determine the positions of the bad pixels.

The input SOF should contain at least five pairs of flat field exposures, all belonging to the same quadrant, each pair corresponding to a different exposure time. The flat fields can be produced either in imaging or in MOS mode. In MOS mode a HR grism is used, in order to illuminate the CCD also beyond the central region used in direct imaging mode, but no mask is inserted at the telescope focal plane. This type of exposure cannot really be considered a *spectral* flat field, because the CCD is exposed to "white" light (*i.e.*, without a wavelength dependency along the dispersion direction). The flat fields generated for the purpose of determining the detector properties (produced by the technical templates `VIMOS_img_tec_DetLin` and `VIMOS_mos_tec_DetLin`) are assigned the DO category `DETECTOR_PROPERTIES`, to distinguish them from the more common `IMG_SCREEN_FLAT` or `MOS_SCREEN_FLAT` that are used to produce master calibrations.

All the files that must be included in the input SOF are listed in table 5.8.1.

DO category	**Type**	**Explanation**	**Required**
DETECTOR_PROPERTIES	Raw frame	Flat field exposure	$\surd$
MASTER_BIAS	Calibration	Master bias	$\surd$

Table 5.8.1: *Input files for the vmdet recipe.*

The products of the *vmdet* recipe are indicated in Table 5.8.2. Only the primary product, the bad pixel table, is copied (or moved) to the product directory. Other products are generated only on request (typically for debug purposes) and are not assigned a DO category as they would not be used anywhere in further data processing steps.

File name	**DO category**	**Type**	**Explanation**
ccd_table.tfits	CCD_TABLE	FITS	Bad pixel table
bad_pixel_map.fits		FITS	Bad pixel image
error_image.fits		FITS	Error image

Table 5.8.2: *Products of the vmdet recipe.*

The *vmdet* parameters are listed in Table 5.8.3.

Parameter	Possible values	Explanation
DetectionMode	Intolerant Tolerant	All pixels with anomalous response are bad Only non-linear pixels are bad
DetectionThreshold	*float* (sigma)	Tolerance on bad pixel detection
CreateBadPixelMap	true false	Create a bad pixel image Do not create it
CreateErrorImage	true false	Create an error image Do not create it

Table 5.8.3: *vmdet parameters.*

A more complete description of the parameters meaning is also given:

CreateBadPixelMap: If this parameter is set, a bad pixel image reflecting the content of the created bad pixel table is created. This may be useful for determining the optimal settings for the parameters *DetectionMode* and *DetectionThreshold*, viewing the frequency of "bad" pixels and their spatial distribution.

CreateErrorImage: If this parameter is set an error image is created. The error image contains the values of the RMS of the residuals for each linear fitting done for bad pixel detection.

DetectionMode: Method used for detecting bad pixels. Possible settings are:

Intolerant: A pixel is flagged as "bad" when the slope of the linear fit of each image median exposure level versus the corresponding pixel values deviates from the local average of all slopes by more than the threshold specified in *DetectionThreshold*.

Tolerant: The same method as in the "*Intolerant*" *DetectionMode* is applied, but before linear fitting the measured pixel values are normalised so that the maximum pixel value is equal to the maximum median exposure level.

DetectionThreshold: Number of standard deviations from the mean slope of the CCD response, that are necessary for classifying a pixel as "bad".

A description of the algorithms used in this recipe is given in Section 7.7, page 124.

5.9 vmbias

The VIMOS pipeline recipe *vmbias* is used to create a master bias frame from a set of raw bias frames. All the files that must be included in the input SOF are listed in Table 5.9.1.

DO category	Type	Explanation	Required
BIAS	Raw frame	Bias exposure	√
CCD_TABLE	Calibration	Bad pixel table	

Table 5.9.1: *Input files for the vmbias recipe.*

A bad pixel table needs to be specified only if the cleaning of bad pixels is requested. In the calibration directories there is one CCD_TABLE file for each quadrant, named `badpixel.`q`.tfits` (where q is the quadrant number). Care should be made in selecting the appropriate bad pixel tables for imaging and spectral instrument modes, that have the same names but are located in separated directories.

The only product of the *vmbias* recipe is the master bias, as indicated in Table 5.9.2.

File name	DO category	Type	Explanation
master_bias.fits	MASTER_BIAS	FITS	Master bias

Table 5.9.2: *Product of the vmbias recipe.*

The *vmbias* parameters are described in Table 5.9.3.

A more complete description of the parameters meaning is also given:

AllowSingleFrames: If this parameter is set, then a master bias is produced also from a single input bias. In this case the *StackMethod* is ignored.

CleanBadPixel: Bad pixel correction on the master bias. If this option is turned on, a bad pixel table should be specified in the input SOF (see Table 5.9.1). The bad pixel correction algorithm is described in Section 7.1, page 121.

CleanCosmic: Cosmic ray events removal from each input bias. The cosmic ray rejection algorithm is described in Section 7.2, page 122.

ComputeQC: If this parameter is set, *Quality Control* (QC) parameters will be computed and written to the header of the output master bias and to an output QC PAF file named `qc0000.paf`. This file is not classified as a pipeline recipe product, as it is an intermediate dataset that in the standard pipeline operations would be translated into new entries in the QC log file. Currently the QC parameters computed by *vmbias* are:

`QC BIAS MEAN:` Mean value of the 1600x1800 central pixels of the first raw bias listed in the SOF.

`QC BIAS MEDIAN:` Median value of the 1600x1800 central pixels of the first raw bias listed in the SOF.

Parameter	Possible values	Explanation
AllowSingleFrames	true false	A single input bias is also allowed More than one input bias is required
StackMethod	Average Median MinMax Ksigma Auto	Master bias is average of input biases Master bias is median of input biases Master bias is obtained with min-max rejection Master bias is obtained with K-sigma clipping Optimal combination of input biases
MinRejection	*int*	No. of lowest rejected values for rejection method
MaxRejection	*int*	No. of highest rejected values for rejection method
KSigmaLow	*float* (sigma)	Low threshold for K-sigma clipping method
KSigmaHigh	*float* (sigma)	High threshold for K-sigma clipping method
RemoveOverscan	true false	Remove overscan regions from master bias Keep overscan regions in master bias
CleanBadPixel	true false	Interpolate bad pixels on master bias No bad pixel correction
CleanCosmic	true false	Remove cosmic ray events from each bias No cosmic ray removal
ComputeQC	true false	Compute QC parameters Do not compute QC parameters

Table 5.9.3: *vmbias parameters.*

`QC BIAS RMS`: The population standard deviation of the 1600x1800 central pixels of the first input bias.

`QC RON`: Population standard deviation of the 1600x1800 central pixels of the difference between the first and the second raw biases listed in the SOF, divided by $\sqrt{2}$.

`QC BIAS FPN`: The population standard deviation of the 1600x1800 central pixels of the difference between the first raw bias and the second raw bias shifted by 10x10 pixels, is computed. This is the combination of fixed-pattern-noise and read-out-noise (scaled by $\sqrt{2}$). The read-out-noise contribution (`QC RON`) is then quadratically subtracted from the total noise.

`QC BIAS STRUCT`: The population standard deviation of the 1600x1800 central pixels of the first raw bias (`QC BIAS RMS`) is the combination of structure, fixed-pattern-noise, and read-out-noise. The read-out-noise (`QC RON`) and the fixed-pattern-noise (`QC BIAS FPN`) contributions are quadratically subtracted from this value.

`QC BIAS MASTER MEAN`: Mean value of the 1600x1800 central pixels of the product master bias.

`QC BIAS MASTER MEDIAN`: Median value of the 1600x1800 central pixels of the product master bias.

`QC BIAS MASTER RMS`: Population standard deviation of all the 1600x1800 central pixel values of the product master bias.

`QC BIAS MASTER NOISE`: The expected noise is computed as the value of `QC RON`, divided by the square root of the number of raw bias frames used in the construction of the master bias. Next, the population standard deviation of the 1600x1800 central pixel values of the master bias is determined, excluding from the computation all values differing from `QC BIAS MASTER MEDIAN` more than three times the expected noise.

`QC BIAS MASTER FPN`: The population standard deviation of the difference between the central 1600x1800 pixels of the master bias, and the region of the master bias shifted 10x10 pixels from the central one, is computed. This is the combination of fixed-pattern-noise and white noise (scaled by $\sqrt{2}$). The white-noise contribution (`QC BIAS MASTER NOISE`) is then quadratically subtracted from the total noise.

`QC BIAS MASTER STRUCT`: The population standard deviation of the 1600x1800 central pixels of the master bias is computed. This is the combination of structure, fixed-pattern-noise, and white-noise. The white-noise (`QC BIAS MASTER NOISE`) and the fixed-pattern-noise (`QC BIAS MASTER FPN`) contributions are then quadratically subtracted.

KSigmaHigh: Number of standard deviations above the median pixel value for rejecting a pixel value when *StackMethod* is set to *"Ksigma"*.

KSigmaLow: Number of standard deviations below the median pixel value for rejecting a pixel value when *StackMethod* is set to *"Ksigma"*.

MaxRejection: Number of highest pixel values to be rejected when *StackMethod* is set to *"MinMax"*.

MinRejection: Number of lowest pixel values to be rejected when *StackMethod* is set to *"MinMax"*.

RemoveOverscan: When this parameter is set, the overscan regions are removed from the product master bias.

StackMethod: Combination method of input biases for master bias creation. See Section 7.6 for a complete description of all the combination methods. Possible settings are:

Auto: Given the number of input biases, an optimal bias combination method is selected. Currently this is always going to the method *"Average"*.

Average: The master bias is the mean of the input frames.

Ksigma: The master bias is the mean of the input frames, after K-sigma screening of pixel values. The number of sigma to be applied in the rejection is specified by the parameters *KSigmaLow* and *KSigmaHigh*.

Median: The master bias is the median of the input frames.

MinMax: The master bias is the mean of the input frames, after rejection of minimum and maximum values. The number of values to reject is specified by the parameters *MinRejection* and *MaxRejection*.

A description of the algorithms used in this recipe is given in Section 7.8, page 126.

5.10 vmdark

The VIMOS pipeline recipe *vmdark* is used to create a master dark frame from a set of raw dark frames. All the files that must be included in the input SOF are listed in Table 5.10.1.

DO category	Type	Explanation	Required
DARK	Raw frame	Dark exposure	√
MASTER_BIAS	Calibration	Master bias	√
CCD_TABLE	Calibration	Bad pixel table	

Table 5.10.1: *Input files for the vmdark recipe.*

A bad pixel table needs to be specified only if the cleaning of bad pixels is requested. In the calibration directories there is one CCD_TABLE file for each quadrant, named `badpixel.`q`.tfits` (where q is the quadrant number). Care should be made in selecting the appropriate bad pixel tables for imaging and spectral instrument modes, that have the same names but are located in separated directories.

The only product of the *vmdark* recipe is the master dark, as indicated in Table 5.10.2.

File name	DO category	Type	Explanation
master_dark.fits	MASTER_DARK	FITS	Master dark

Table 5.10.2: *Product of the vmdark recipe.*

The *vmdark* parameters are listed in Table 5.10.3.

A more complete description of the parameters meaning is also given:

AllowSingleFrames: If this parameter is set, then a master dark is produced also from a single input dark frame. In that case the *StackMethod* is ignored.

BiasMethod: Method for bias removal from the input dark frames. The bias removal procedure is described in some detail in Section 7.3. Possible settings are:

Master: After master bias subtraction, prescan and overscan regions are trimmed away from the dark frame.

Zmaster: After master bias subtraction the overscan correction is applied before trimming away the overscan regions.

CleanBadPixel: Bad pixel correction on the master dark. If this option is turned on, a bad pixel table should be specified in the input SOF (see Table 5.10.1). The bad pixel correction algorithm is described in Section 7.1, page 121.

CleanCosmic: Cosmic ray events removal from each input dark. The cosmic ray rejection algorithm is described in Section 7.2, page 122.

Parameter	Possible values	Explanation
AllowSingleFrames	true false	A single input dark is also allowed More than one input dark is required
StackMethod	Average Median MinMax Ksigma Auto	Master dark is average of input darks Master dark is median of input darks Master dark is obtained with min-max rejection Master dark is obtained with K-sigma clipping Optimal combination of input darks
MinRejection	*int*	No. of lowest rejected values for rejection method
MaxRejection	*int*	No. of highest rejected values for rejection method
KSigmaLow	*float* (sigma)	Low threshold for K-sigma clipping method
KSigmaHigh	*float* (sigma)	High threshold for K-sigma clipping method
BiasMethod	Master Zmaster	Bias removal with no overscan correction Bias removal with overscan correction
CleanBadPixel	true false	Interpolate bad pixels on master dark No bad pixel correction
CleanCosmic	true false	Remove cosmic ray events from each dark No cosmic ray removal
CosmicThreshold	*float*	Sigmas above level discriminator
CosmicRatio	*float*	Peak/neighbours discriminator
ComputeQC	true false	Compute QC parameters Do not compute QC parameters

Table 5.10.3: *vmdark parameters.*

ComputeQC: If this parameter is set, *Quality Control* (QC) parameters will be computed and written to the header of the output master dark and to an output QC PAF file named `qc0000.paf`. This file is not classified as a pipeline recipe product, as it is an intermediate dataset that in the standard pipeline operations would be translated into new entries in the QC log file. Currently the QC parameters computed by *vmdark* are:

`QC DARK MASTER MEAN:` Mean value of the 1600x1800 central pixels of the product master dark (ADU/s).

`QC DARK MASTER RMS:` Population standard deviation of all 1600x1800 central pixel values of the product master dark (ADU/s).

`QC DARK MASTER MEDIAN:` Median value of the 1600x1800 central pixels of the product master dark (ADU/s).

`QC DARK CURRENT:` Simple conversion of `QC DARK MASTER MEDIAN` into e^-/pixel/hour.

`QC DARK CURRENT RMS:` Simple conversion of `QC DARK CURRENT RMS` into e^-/pixel/hour.

CosmicRatio: Critical ratio for discriminating between objects and cosmic rays. This parameter is effective when *CleanCosmic* is set.

CosmicThreshold: Threshold for the selection of cosmic rays candidates. This parameter is effective when *CleanCosmic* is set.

KSigmaLow: Number of standard deviations below the median pixel value for rejecting a pixel value when *StackMethod* is set to *"Ksigma"*.

KSigmaHigh: Number of standard deviations above the median pixel value for rejecting a pixel value when *StackMethod* is set to *"Ksigma"*.

MaxRejection: Number of highest pixel values to be rejected when *StackMethod* is set to *"MinMax"*.

MinRejection: Number of lowest pixel values to be rejected when *StackMethod* is set to *"MinMax"*.

StackMethod: Combination method of input darks for master dark creation. See Section 7.6 for a complete description of all the combination methods. Possible settings are:

Auto: Given the number of input darks, an optimal dark combination method is selected. Currently this is always going to the method *"Average"*.

Average: The master dark is the mean of the input frames.

Ksigma: The master dark is the mean of the input frames, after K-sigma screening of pixel values. The number of sigma to be applied in the rejection is specified by the parameters *KSigmaLow* and *KSigmaHigh*.

Median: The master dark is the median of the input frames.

MinMax: The master dark is the mean of the input frames, after rejection of minimum and maximum values. The number of values to reject is specified by the parameters *MinRejection* and *MaxRejection*.

A description of the algorithms used in this recipe is given in Section 7.9, page 126.

5.11 vmimflatscreen

The VIMOS pipeline recipe *vmimflatscreen* is used to create a master screen flat field from a set of raw screen flat fields. The master screen flat field is not used directly in the flat field correction of scientific data, but it is optionally used just in the creation of a master sky flat field (see Section 5.12, page 63).

All the files that must be included in the input SOF are listed in table 5.11.1.

DO category	**Type**	**Explanation**	**Required**
IMG_SCREEN_FLAT	Raw frame	Screen flat field exposure	√
MASTER_BIAS	Calibration	Master bias	√
MASTER_DARK	Calibration	Master dark	
CCD_TABLE	Calibration	Bad pixel table	

Table 5.11.1: *Input files for the vmimflatscreen recipe.*

A bad pixel table needs to be specified only if the cleaning of bad pixels is requested. In the calibration directories there is one CCD_TABLE file for each quadrant, named `badpixel.`q`.tfits` (where q is the quadrant number). Care should be made in selecting the appropriate bad pixel tables for imaging and spectral instrument modes, that have the same names but are located in separated directories.

The primary product of the *vmimflatscreen* recipe is the normalised master screen flat field, as indicated in Table 5.11.2. A secondary product is the combined screen flat field, that is the result of the combination of all inputs but without any normalisation applied, and is just used for data quality control.

File name	**DO category**	**Type**	**Explanation**
img_master_screen_flat.fits	IMG_MASTER_SCREEN_FLAT	FITS	Master screen flat field
img_combined_screen_flat.fits	IMG_COMBINED_SCREEN_FLAT	FITS	Combined screen flat field

Table 5.11.2: *Products of the vmimflatscreen recipe.*

The *vmimflatscreen* parameters are listed in table 5.11.3.

A more complete description of the parameters meaning is also given:

AllowSingleFrames: If this parameter is set, then a master screen flat field is produced also from a single input screen flat field frame. In that case the *StackMethod* is ignored.

BiasMethod: Method for bias removal from the input screen flat field frames. The bias removal procedure is described in some detail in Section 7.3. Possible settings are:

- **Master:** After master bias subtraction, prescan and overscan regions are trimmed away from the flat field frame.
- **Zmaster:** After master bias subtraction the overscan correction is applied before trimming away the overscan regions.

Parameter	**Possible values**	**Explanation**
AllowSingleFrames	true false	A single input flat field is also allowed More than one input flat field is required
StackMethod	Average Median MinMax Ksigma Auto	Combined flat field is average of inputs Combined flat field is median of inputs Combined flat field is obtained with min-max rejection Combined flat field is obtained with K-sigma clipping Optimal combination of input flat fields
MinRejection	*int*	No. of lowest rejected values for rejection method
MaxRejection	*int*	No. of highest rejected values for rejection method
KSigmaLow	*float* (sigma)	Low threshold for K-sigma clipping method
KSigmaHigh	*float* (sigma)	High threshold for K-sigma clipping method
BiasMethod	Master Zmaster	Bias removal with no overscan correction Bias removal with overscan correction
CleanBadPixel	true false	Interpolate bad pixels on product flat fields No bad pixel correction
CleanCosmic	true false	Remove cosmic ray events from each flat field No cosmic ray removal
CosmicThreshold	*float*	Sigmas above level discriminator
CosmicRatio	*float*	Peak/neighbours discriminator
SmoothBoxSize	*int* (pixel)	Size of smoothing running box
SmoothMethod	Median Average	Median of values in running box Average of values in running box
ComputeQC	true false	Compute QC parameters Do not compute QC parameters

Table 5.11.3: *vmimflatscreen parameters.*

CleanBadPixel: Bad pixel correction on the products. If this option is turned on, a bad pixel table should be specified in the input SOF (see Table 5.12.1). The bad pixel correction algorithm is described in Section 7.1, page 121.

CleanCosmic: Cosmic ray events removal from each input flat field. The cosmic ray rejection algorithm is described in Section 7.2, page 122.

ComputeQC: If this parameter is set, *Quality Control* (QC) parameters will be computed and written to the header of the output master screen flat field and to an output QC PAF file named `qc0000.paf`. This file is not classified as a pipeline recipe product, as it is an intermediate dataset that in the standard pipeline operations would be translated into new entries in the QC log file. The QC parameters are computed only if the exposure time of the first two raw screen flat fields listed in the input SOF is the same (within 4%). Currently the QC parameters computed by *vmimflatscreen* are:

`QC CONAD:` Conversion factor from ADU to electrons (e^-/ADU). The difference frame of the first two raw screen flat fields listed in the input SOF is computed. Then the 1600x1800 central region of the image is divided into 16x18 100x100 boxes. For each one of these boxes, the median signal level from the first raw frame is divided by the variance in the difference frame scaled by 2. The median value of the 16x18 values obtained is the accepted value for the gain conversion factor.

`QC CONAD RMS:` The rms of the 16x18 values obtained in the determination of `QC CONAD` is computed, and divided by the square root of 16x18.

`QC FLAT PHN:` Photon noise (in ADU). The standard deviation of the 1600x1800 pixel central region of the difference of the first two raw screen flat fields listed in the input SOF is computed and then scaled by $\sqrt{2}$.

`QC FLAT FPN:` Fixed pattern noise (in ADU). The difference between the 1600x1800 central pixels of the first frame, and the same region shifted by 10x10 pixels in the second frame, is computed. The standard deviation of the signal is the combination of fixed pattern noise and photon noise (scaled by $\sqrt{2}$). The photon noise `QC FLAT PHN` is then quadratically subtracted.

`QC FLAT STRUCT:` Screen flat field structure (in ADU). The population standard deviation of the 1600x1800 central pixels of the first flat field in the input SOF is computed. This is the combination of structure, fixed pattern noise `QC FLAT FPN`, and photon noise `QC FLAT PHN`. The photon noise and the fixed pattern noise are then quadratically subtracted.

`QC FLAT EFFICIENCY:` Signal per unit of exposure (in ADU/s). The median level of the 1600x1800 central pixels of the first input screen flat field is divided by its exposure time.

`QC FLAT MASTER MEDIAN:` Median value of the 1600x1800 central pixels of the combined screen flat field.

`QC FLAT MASTER RMS:` Population standard deviation of the 1600x1800 central pixels of the combined screen flat field.

CosmicRatio: Critical ratio for reducing the effect of variable background on cosmic rays identification. This parameter is effective when *CleanCosmic* is set.

CosmicThreshold: Threshold for the selection of cosmic rays candidates. This parameter is effective when *CleanCosmic* is set.

KSigmaHigh: Number of standard deviations above the median pixel value for rejecting a pixel value when *StackMethod* is set to *"Ksigma"*.

KSigmaLow: Number of standard deviations below the median pixel value for rejecting a pixel value when *StackMethod* is set to *"Ksigma"*.

MaxRejection: Number of highest pixel values to be rejected when *StackMethod* is set to *"MinMax"*.

MinRejection: Number of lowest pixel values to be rejected when *StackMethod* is set to *"MinMax"*.

SmoothBoxSize: Length in pixel of the side of the square smoothing box used in the normalisation of the master flat field.

SmoothMethod: The smoothing method used in the normalisation of the master flat field. Possible settings are:

Average: The central pixel within the smoothing running box is replaced with the average of the values of the pixels contained in the box.

Median: The central pixel within the smoothing running box is replaced with the median of the values of the pixels contained in the box.

StackMethod: Combination method of input screen flat fields for combined flat field creation. See Section 7.6 for a complete description of all the combination methods. Possible settings are:

Auto: Given the number of input screen flat fields, an optimal combination method is selected. Currently this is always going to the method *"Average"*.

Average: The combined screen flat field is the mean of the input frames.

Ksigma: The combined screen flat field is the mean of the input frames, after K-sigma screening of pixel values. The number of sigma to be applied in the rejection is specified by the parameters *KSigmaLow* and *KSigmaHigh*.

Median: The combined screen flat field is the median of the input frames.

MinMax: The combined screen flat field is the mean of the input frames, after rejection of minimum and maximum values. The number of values to reject is specified by the parameters *MinRejection* and *MaxRejection*.

A description of the algorithms used in this recipe is given in Section 7.10, page 127.

5.12 vmimflatsky

The VIMOS pipeline recipe *vmimflatsky* is used to create a master sky flat field from a set of raw sky flat fields. The master sky flat field is the dataset used for the flat field correction of scientific data.

All the files that must be included in the input SOF are listed in table 5.12.1.

DO category	Type	Explanation	Required
IMG_SKY_FLAT	Raw frame	Sky flat field exposure	√
MASTER_BIAS	Calibration	Master bias	√
MASTER_DARK	Calibration	Master dark	
IMG_MASTER_SCREEN_FLAT	Calibration	Master screen flat field	
CCD_TABLE	Calibration	Bad pixel table	

Table 5.12.1: *Input files for the vmimflatsky recipe.*

A bad pixel table needs to be specified only if the cleaning of bad pixels is requested. In the calibration directories there is one CCD_TABLE file for each quadrant, named `badpixel.`q`.tfits` (where q is the quadrant number). Care should be made in selecting the appropriate bad pixel tables for imaging and spectral instrument modes, that have the same names but are located in separated directories.

The only product of the *vmimflatsky* recipe is the normalised master sky flat field, as indicated in Table 5.12.2.

File name	DO category	Type	Explanation
img_master_sky_flat.fits	IMG_MASTER_SKY_FLAT	FITS	Master sky flat field

Table 5.12.2: *Products of the vmimflatsky recipe.*

The *vmimflatsky* parameters are listed in table 5.12.3.

A more complete description of the parameters meaning is also given:

AllowSingleFrames: If this parameter is set, then a master sky flat field is produced also from a single input sky flat field frame. In that case the *StackMethod* is ignored.

BiasMethod: Method for bias removal from the input sky flat field frames. The bias removal procedure is described in some detail in Section 7.3. Possible settings are:

- **Master:** After master bias subtraction, prescan and overscan regions are trimmed away from the flat field frame.
- **Zmaster:** After master bias subtraction the overscan correction is applied before trimming away the overscan regions.

CleanBadPixel: Bad pixel correction on the master sky flat field. If this option is turned on, a bad pixel table should be specified in the input SOF (see Table 5.12.1). The bad pixel correction algorithm is described in Section 7.1, page 121.

Parameter	**Possible values**	**Explanation**
AllowSingleFrames	true false	A single input flat field is also allowed More than one input flat field is required
StackMethod	Average Median MinMax Ksigma Auto	Combined flat field is average of inputs Combined flat field is median of inputs Combined flat field is obtained with min-max rejection Combined flat field is obtained with K-sigma clipping Optimal combination of input flat fields
MinRejection	*int*	No. of lowest rejected values for rejection method
MaxRejection	*int*	No. of highest rejected values for rejection method
KSigmaLow	*float* (sigma)	Low threshold for K-sigma clipping method
KSigmaHigh	*float* (sigma)	High threshold for K-sigma clipping method
BiasMethod	Master Zmaster	Bias removal with no overscan correction Bias removal with overscan correction
CleanBadPixel	true false	Interpolate bad pixels on master sky flat No bad pixel correction
CleanCosmic	true false	Remove cosmic ray events from each flat field No cosmic ray removal
CosmicThreshold	*float*	Sigmas above level discriminator
CosmicRatio	*float*	Peak/neighbours discriminator
SmoothBoxSize	*int* (pixel)	Size of smoothing running box
SmoothMethod	Median Average	Median of values in running box Average of values in running box
ComputeQC	true false	Compute QC parameters Do not compute QC parameters

Table 5.12.3: *vmimflatsky parameters.*

CleanCosmic: Cosmic ray events removal from each input flat field. The cosmic ray rejection algorithm is described in Section 7.2, page 122.

ComputeQC: If this parameter is set, *Quality Control* (QC) parameters will be computed and written to the header of the output master sky flat field and to an output QC PAF file named `qc0000.paf`. This file is not classified as a pipeline recipe product, as it is an intermediate dataset that in the standard pipeline operations would be translated into new entries in the QC log file. Currently the QC parameters computed by *vmimflatsky* are:

`QC SKY FLAT FLUX:` Mean value of the 1600x1800 central pixels of the first sky flat field listed in the input SOF, after bias removal and division by the exposure time.

`QC SKY FLAT RMS:` The population standard deviation of the 1600x1800 central pixels of the normalised master sky flat field.

`QC SKY FLAT STRUCT:` The standard deviation `QC SKY FLAT RMS` can be seen as the combination of large scale structure with noise sources. The difference between the master and the master itself shifted by 10x10 pixels is computed, and the variance of the 1600x1800 central pixels of the result is computed and corrected by a factor 2. This evaluation of other noise sources is then quadratically subtracted from the total standard deviation.

CosmicRatio: Critical ratio for reducing the effect of variable background on cosmic rays identification. This parameter is effective when *CleanCosmic* is set.

CosmicThreshold: Threshold for the selection of cosmic rays candidates. This parameter is effective when *CleanCosmic* is set.

KSigmaHigh: Number of standard deviations above the median pixel value for rejecting a pixel value when *StackMethod* is set to *"Ksigma"*.

KSigmaLow: Number of standard deviations below the median pixel value for rejecting a pixel value when *StackMethod* is set to *"Ksigma"*.

MaxRejection: Number of highest pixel values to be rejected when *StackMethod* is set to *"MinMax"*.

MinRejection: Number of lowest pixel values to be rejected when *StackMethod* is set to *"MinMax"*.

SmoothBoxSize: Length in pixel of the side of the square smoothing box used in the normalisation of the master flat field.

SmoothMethod: The smoothing method used in the normalisation of the master flat field. Possible settings are:

- **Median:** The central pixel within the smoothing running box is replaced with the median of the values of the pixels contained in the box.
- **Average:** The central pixel within the smoothing running box is replaced with the average of the values of the pixels contained in the box.

StackMethod: Combination method of input sky flat fields for combined flat field creation. See Section 7.6 for a complete description of all the combination methods. Possible settings are:

- **Auto:** Given the number of input sky flat fields, an optimal combination method is selected. Currently this is always going to the method *"Average"*.
- **Average:** The combined sky flat field is the mean of the input frames.
- **Ksigma:** The combined sky flat field is the mean of the input frames, after K-sigma screening of pixel values. The number of sigma to be applied in the rejection is specified by the parameters *KSigmaLow* and *KSigmaHigh*.
- **Median:** The combined sky flat field is the median of the input frames.
- **MinMax:** The combined sky flat field is the mean of the input frames, after rejection of minimum and maximum values. The number of values to reject is specified by the parameters *MinRejection* and *MaxRejection*.

A description of the algorithms used in this recipe is given in Section 7.11, page 127.

5.13 vmmasktoccd

The VIMOS pipeline recipe *vmmasktoccd* is used to determine the CCD to Mask transformation and its inverse (see Section 6.2.1, page 113). SExtractor v2.1.6 [8] is used for determining the positions of the spotlights on a direct imaging exposure of a lamp and a calibration mask containing a regular grid of pinholes. The relation between mask and CCD positions is then determined[4].

The files to be included in the input SOF are listed in Table 5.13.1.

DO category	Type	Explanation	Required
MASK_TO_CCD	Raw frame	Pinhole mask exposure	√
MASTER_BIAS	Calibration	Master bias	√
MASTER_DARK	Calibration	Master dark	
IMG_MASTER_SKY_FLAT	Calibration	Master flat field	
CCD_TABLE	Calibration	Bad pixel table	

Table 5.13.1: *Input files for the vmmasktoccd recipe.*

A bad pixel table needs to be specified only if the cleaning of bad pixels is requested. In the calibration directories there is one CCD_TABLE file for each quadrant, named `badpixel.`q`.tfits` (where q is the quadrant number). Care should be made in selecting the appropriate bad pixel tables for imaging and spectral instrument modes, that have the same names but are located in separated directories.

The only product of the *vmmasktoccd* recipe is indicated in Table 5.13.2. This PAF file is copied (or moved) to the product directory, and it is identical to the produced *Instrument WorkStation* (IWS) configuration file `IMG_mask2ccd_`q`.cmf` (where q indicates the VIMOS quadrant number) that is created in the same directory where the recipe is launched.

File name	DO category	Type	Explanation
IMG_mask2ccd_q.paf		PAF	Mask to CCD configuration file

Table 5.13.2: *Products of the vmmasktoccd recipe.*

The transformation and distortion models related to the CCD and the mask focal planes are described in Section 6.2.1, page 113. Typically, the RMS of the CCD to mask model residuals is about $5 \cdot 10^{-3}$ mm, while the RMS of the inverse transformation is about 0.04 pixel (being related through the 8.4 mm/pixel scale factor). Of course the real accuracy of the model may be better than that, considering that about 500 pinholes positions are fitted to determine the transformations: if the distribution of the residuals were poissonian, the model accuracy would be more accurate than the residuals population RMS by a factor of the order of $\sqrt{500/22} \simeq 5$ (where 22 is the number of the model's free parameters).

The *vmmasktoccd* parameters are listed in Table 5.13.3.

A more complete description of the parameters meaning is also given:

[4] The *vmmasktoccd* recipe is not distributed outside ESO because of licensing problems.

Parameter	**Possible values**	**Explanation**
BiasMethod	Master Zmaster	Bias removal with no overscan correction Bias removal with overscan correction
CleanBadPixel	true false	Clean bad pixels Do not clean bad pixels
CleanCosmic	true false	Clean cosmic rays Do not clean cosmic rays
CosmicThreshold	*float*	Sigmas above level discriminator
CosmicRatio	*float*	Peak/neighbours discriminator
Iterations	*int*	Number of model iterations
PolyOrderX	*int*	Order of the x distortion model
PolyOrderY	*int*	Order of the y distortion model
SearchRadius	*float* (pixel)	Max distance from expected positions
SExtractor.FilterName	*file*	SExtractor convolution mask
SExtractor.HolesParam	*file*	SExtractor output parameters
SExtractor.HolesSex	*file*	SExtractor configuration file
SExtractor.PsfEx	*file*	PSF modeling executable
SExtractor.PsfexDefault	*file*	PSF modeling configuration file
SExtractor.PsfParam	*file*	SExtractor output parameters for PSF modeling
SExtractor.PsfSex	*file*	SExtractor configuration file used for PSF modeling
SExtractor.SExtractor	*file*	SExtractor executable
SExtractor.StarNnwName	*file*	SExtractor neural network weights

Table 5.13.3: *vmmasktoccd parameters.*

BiasMethod: Method for bias removal from the pinhole mask image. The bias removal procedure is described in some detail in Section 7.3. Possible settings are:

Master: After master bias subtraction, prescan and overscan regions are trimmed away from the pinhole mask image.

Zmaster: After master bias subtraction the overscan correction is applied before trimming away the overscan regions.

CleanBadPixel: Bad pixel correction on the pinhole mask image. If this option is turned on, a bad pixel table should be specified in the input SOF (see Table 5.13.1). The bad pixel correction algorithm is described in Section 7.1, page 121.

CleanCosmic: Cosmic ray events removal from pinhole mask image. The cosmic ray rejection algorithm is described in Section 7.2, page 122.

CosmicRatio: Critical ratio for discriminating between objects and cosmic rays. This parameter is effective when *CleanCosmic* is set.

CosmicThreshold: Threshold for the selection of cosmic rays candidates. This parameter is effective when *CleanCosmic* is set.

Iterations: Max number of iterations of the distortion model fit.

PolyOrderX: Order of the bivariate polynomial for the x coordinate distortion. Note that VMMPS requires this to be set to 3.

PolyOrderY: Order of the bivariate polynomial for the y coordinate transformation. Note that VMMPS requires this to be set to 3.

SearchRadius: Max distance (in pixels) from expected position where a pinhole is searched.

The parameters belonging to the SExtractor group should not be modified.

A description of the algorithms used in this recipe is given in Section 7.12, page 128.

5.14 vmskyccd

The VIMOS pipeline recipe *vmskyccd* is used to determine the CCD to Sky distortion and its inverse (see Section 6.2.2, page 114). SExtractor v2.1.6 [8] is used for detecting in the field-of-view the objects matching the entries of an astrometric catalog. The deviations from the theoretical relation between CCD and celestial coordinates (WCS) are then determined and modeled[5].

The files to be included in the input SOF are listed in Table 5.14.1.

DO category	Type	Explanation	Required
IMG_ASTROMETRY	Raw frame	Astrometric field exposure	√
MASTER_BIAS	Calibration	Master bias	√
MASTER_DARK	Calibration	Master dark	
IMG_MASTER_SKY_FLAT	Calibration	Master flat field	√
ASTROMETRIC_TABLE	Calibration	Astrometric catalog	√
PHOTOMETRIC_TABLE	Calibration	Photometric table	
CCD_TABLE	Calibration	Bad pixel table	

Table 5.14.1: *Input files for the vmskyccd recipe.*

A bad pixel table needs to be specified only if the cleaning of bad pixels is requested, and the photometric table is required only if star matching is based also on comparing the measured and the catalog magnitudes.

In the calibration directories there is one CCD_TABLE file for each quadrant, named `badpixel.`q`.tfits` (where q is the quadrant number). Care should be made in selecting the appropriate bad pixel tables for imaging and spectral instrument modes, that have the same names but are located in separated directories.

The standard photometric tables in the calibration directories are named `ipc_`f`i.`q`.tfits` (where f is the filter name and q the quadrant number), while the astrometric catalogue is contained in tables `region`x`.tfits` (where x is a character between A and P). The name of the table to be used to process a given exposure can be derived from the value of its FITS header keyword `ESO OBS TARG NAME`.

The products of the *vmskyccd* recipe are indicated in Table 5.14.2.

[5] The *vmskyccd* recipe is not distributed outside ESO.

File name	DO category	Type	Explanation
IMG_sky2ccd_*q*.paf		PAF	Sky to CCD configuration file
img_astrometry_reduced.fits	IMG_ASTROMETRY_REDUCED	FITS	Reduce astrometric field exposure

Table 5.14.2: *Products of the vmskyccd recipe.*

The PAF file is copied (or moved) to the product directory, and is identical to the *Instrument WorkStation* (IWS) configuration file `IMG_sky2ccd_`*q*`.cmf` (where *q* indicates the VIMOS quadrant number) that is created in the same directory where the recipe is launched.

The reduced astrometric image is a byproduct of the distortions modeling, and can be used to ensure that the data are suitable for the purpose. Quality indicators as the *seeing* are not critical in this context, but the number and the distribution of the identified astrometric stars in the field-of-view is of fundamental importance for a realistic modeling of the distortions through polynomial fitting. This, together with the RMS of the model residuals (see Section 6.2.2, page 114), should make possible to discriminate between a safe and an unsafe modeling of the sky to CCD distortions.

The *vmskyccd* parameters are listed in Table 5.14.3.

Only few parameters of the SExtractor group are listed here: almost all the entries of the SExtractor configuration file, `$PIPE_HOME/vimos/config/vimos.sex`, can be overruled by the SExtractor group parameter values specified in the `vmskyccd.rc` configuration file. Conventionally, a SExtractor configuration parameter name matches the *vmskyccd* configuration parameter obtained by lowercasing all of its characters with the exception of the first character of each word, and removing the underscore. For instance, the SExtractor parameter `CATALOG_TYPE` matches the *vmskyccd* parameter `CatalogType`. When using the corresponding command line option, the SExtractor group must be explicitly specified. For instance, the command option corresponding to the parameter `CatalogType` is `--SExtractor.CatalogType`.

The `Window` parameter has no counterpart in the SExtractor configuration file, and is used to restrict the SExtractor operations to just a part of the input image. This is useful to avoid false detections along the vignetted image region.

A more complete description of the parameters not belonging to the SExtractor group is given here:

ApplyZeropoint: If this parameter is set, a photometric table must be specified in the input SOF, and the magnitude zeropoint, the extinction coefficient, and the colour term (with the colour it refers to) are copied from the photometric table to the reduced astrometric image header. If this parameter is set to false, the parameters *MagInitial* and *MagFinal* are ignored.

BiasMethod: Method for bias removal from the astrometric image. The bias removal procedure is described in some detail in Section 7.3. Possible settings are:

Master: After master bias subtraction, prescan and overscan regions are trimmed away from the astrometric image.

Zmaster: After master bias subtraction the overscan correction is applied before trimming away the overscan regions.

Parameter	**Possible values**	**Explanation**
ApplyZeropoint	true false	Apply zeropoint correction Do not apply zeropoint correction
BiasMethod	Master Zmaster	Bias removal with no overscan correction Bias removal with overscan correction
CleanBadPixel	true false	Clean bad pixels Do not clean bad pixels
CleanCosmic	true false	Clean cosmic rays Do not clean cosmic rays
CosmicThreshold	*float*	Sigmas above level discriminator
CosmicRatio	*float*	Peak/neighbours discriminator
KSigmaClip	*float*	Sigma clipping threshold used in star match
MagFinal	*float*	Magnitude tolerance for final star match
MagInitial	*float*	Magnitude tolerance for initial star match
MagLimit	*float*	Max magnitude for object selection
RemoteCatalog	true false	*Not yet implemented* No online access to a catalog server
SearchRadius	*float* (”)	Max distance for star matching
StarIndex	*float*	Min stellarity index for stars selection
TemperatureCheck	true false	Check beam and ambient temperatures consistency Do not check consistency of temperatures
TemperatureTolerance	*float* (^oC)	Max difference beam - ambient temperatures
SExtractor.FilterName	*file*	SExtractor convolution mask
SExtractor.SExtractor	*file*	SExtractor executable
SExtractor.StarNnwName	*file*	SExtractor neural network weights
SExtractor.Window	*file*	Image region where SExtractor is applied

Table 5.14.3: *vmskyccd parameters.*

CleanBadPixel: Bad pixel correction on the astrometric image. If this option is turned on, a bad pixel table should be specified in the input SOF (see Table 5.14.1). The bad pixel correction algorithm is described in Section 7.1, page 121.

CleanCosmic: Cosmic ray events removal from astrometric image. The cosmic ray rejection algorithm is described in Section 7.2, page 122.

CosmicThreshold: Threshold for the selection of cosmic rays candidates. This parameter is effective when *CleanCosmic* is set.

CosmicRatio: Critical ratio for discriminating between objects and cosmic rays. This parameter is effective when *CleanCosmic* is set.

KSigmaClip: Number of sigmas used in the spatial rejection of matched stars. Stars found within the specified *SearchRadius* are compared to their expected catalog positions, and the sigma of the residuals distribution is computed. Then, all the stars having a distance from the expected position greater than the specified threshold are rejected.

MagFinal: Magnitude tolerance applied to matched stars, after the K-sigma clipping has been applied. The stars magnitudes should not differ from the corresponding catalog magnitudes more than this value.

MagInitial: Magnitude tolerance applied in star matching.

MagLimit: Limiting magnitude of objects to be matched with the astrometric stars from the input catalog.

SearchRadius: Max distance (in arcseconds) from the expected position of a catalog star.

StarIndex: This index can have a value between 0 (minimal stellarity) and 1 (maximal stellarity). Only the identified objects having a stellarity index greater than the specified value will be taken as stars, to be matched with the standard stars from the input catalog.

TemperatureCheck: For associating the detected stars with the reference catalog, their coordinates must be corrected applying a "first guess" CCD to Sky distortion corrected for changes in the beam temperature (see Sections 6.2.2 and 7.13). If this parameter is set, the beam temperature is compared with the temperatures of the other beams and with the ambient temperature. If the difference between the beam temperature and the ambient temperature is greater than the threshold specified by *TemperatureTolerance*, then the temperature used will be the mean of the beams temperatures that differ from the ambient temperature less than the specified threshold. If not a single beam temperature is within tolerance, then the ambient temperature is used in place of the beam temperature. This option was just introduced to prevent possible problems with the beam temperature sensors.

TemperatureTolerance: Max tolerated difference between ambient and beam temperatures. This parameter is effective only if *TemperatureCheck* is set.

The order of the polynomials used in the distortions modeling is taken from the "first guess" model found in the input astrometric image header (see Table 6.2.3, page 115).

The astrometric catalog currently used by *vmskyccd* (see reference [11]) is located in the directory `$PIPE_HOME/vimos/ /ima/cal`, and it is split into a number of tables named `regionA.tfits`, `regionB.tfits`, `...`, `regionP.tfits`. The table indicated in the SOF should match the content of the header entry `OBS TARG NAME` of the input astrometric image.

It should be noted that in this astrometric catalog just the R band magnitudes are given, therefore astrometric fields exposures should always be made with the R filter.

A description of the algorithms used in this recipe is given in Section 7.12, page 128.

5.15 vmimstandard

The VIMOS pipeline recipe *vmimstandard* is used to determine the instrumental magnitude of the stars matching the entries of a photometric catalog. SExtractor v2.1.6 [8] is used for the source detection task.

The files to be included in the input SOF are listed in Table 5.15.1.

DO category	Type	Explanation	Required
IMG_STANDARD	Raw frame	Standard stars field	√
IMG_MASTER_SKY_FLAT	Calibration	Master sky flat field	√
MASTER_BIAS	Calibration	Master bias	√
MASTER_DARK	Calibration	Master dark	
PHOTOMETRIC_CATALOG	Calibration	Photometric catalog	√
CCD_TABLE	Calibration	Bad pixel table	
PHOTOMETRIC_TABLE	Calibration	Photometric table	

Table 5.15.1: *Input files for the vmimstandard recipe.*

The bad pixel table needs to be specified only if the cleaning of bad pixels is requested. In the calibration directories there is one CCD_TABLE file for each quadrant, named `badpixel.`q`.tfits` (where q is the quadrant number). Care should be made in selecting the appropriate bad pixel tables for imaging and spectral instrument modes, that have the same names but are located in separated directories.

The optional determination of the frame magnitude zeropoint from the table of detected standard stars (see ahead) would require to specify in the input SOF a photometric table. The photometric table simply holds the necessary parameters for the magnitude zeropoint computation, as listed in Table 5.15.2. The standard photometric tables in the calibration directories are named `ipc_`f`i.`q`.tfits` (where f is the filter name and q the quadrant number).

Keyword	Example	Explanation
`PRO MAG ZERO`	28.15	Expected magnitude zeropoint
`PRO EXTINCT`	0.25	Atmospheric extinction coefficient
`PRO COLTERM`	0.01	Correction for star colour
`PRO COLOUR`	'B-V'	Colour system used
`PRO MAGZERO RMS`	0.05	Error on expected zeropoint
`PRO EXTINCT RMS`	0.00	Error on extinction coefficient
`PRO COLTERM RMS`	0.00	Error on colour term

Table 5.15.2: *Photometric table entries.*

The photometric catalog currently used can be found in the directory `$PIPE_HOME/vimos/ima/cal`, in the file `phstd_stetson.tfits` (see Table 5.15.3). This table includes the photometric stars from the Stetson's fields (see `http://cadcwww.dao.nrc.ca/standards`); Landolt's stars (Landolt 1992, AJ 104, 340) that can be found in the Stetson's fields are also included, to permit the determination of zeropoints also in the U band.

The products of the *vmimstandard* recipe are indicated in Table 5.15.4.

Column name	Explanation
`ID`	Star identification string
`RA`	RA of star
`DEC`	Dec of star
`MAG_U`	U magnitude of star
`MAG_B`	B magnitude of star
`MAG_V`	V magnitude of star
`MAG_R`	R magnitude of star
`MAG_I`	I magnitude of star

Table 5.15.3: *Photometric catalog entries.*

File name	DO category	Type	Explanation
img_star_match_table.tfits	IMG_STAR_MATCH_TABLE	FITS	List of identified standard stars
img_standard_reduced.fits	IMG_STANDARD_REDUCED	FITS	Reduced standard stars field
img_galaxy_table.tfits	IMG_GALAXY_TABLE	FITS	List of detected objects

Table 5.15.4: *Products of the vmimstandard recipe.*

The galaxy table is the output of SExtractor [8] converted into FITS format (see Table 5.18.3, page 81). The star match table is the list of identified standard stars, with their positions on sky and CCD and their instrumental and catalog magnitudes, as shown in Table 5.15.5.

The same procedure applied by the recipe *vmimobsstare* (see Section 5.18, page 80) is used to reduce the standard field image. The reduced image is a byproduct of the star matching task, and can be used for quality control purposes.

The *vmimstandard* parameters are listed in Table 5.15.6.

Only few parameters of the SExtractor group are listed here: almost all the entries of the SExtractor configuration file, `$PIPE_HOME/vimos/config/vimos.sex`, can be overruled by the SExtractor group parameter values specified in the `vmimstandard.rc` configuration file. Conventionally, a SExtractor configuration parameter name matches the *vmimstandard* configuration parameter obtained by lowercasing all of its characters with the exception of the first character of each word, and removing the underscore. For instance, the SExtractor parameter `CATALOG_TYPE` matches the *vmimstandard* parameter `CatalogType`. When using the corresponding command line option, the SExtractor group must be explicitly specified. For instance, the command option corresponding to the parameter `CatalogType` is `--SExtractor.CatalogType`.

The `Window` parameter has no counterpart in the SExtractor configuration file, and is used to restrict the SExtractor operations to just a part of the input image. This is useful to avoid false detections along the vignetted image region.

A more complete description of the parameters not belonging to the SExtractor group is given here:

BiasMethod: Method for bias removal from the input standard stars field exposures. The bias removal procedure is described in some detail in Section 7.3. Possible settings are:

Column name	Explanation
NUMBER	Object number in SExtractor output (galaxy table)
ID	Star identification string, from input catalog
X_IMAGE	X image pixel position of matched star (SExtractor)
Y_IMAGE	Y image pixel position of matched star (SExtractor)
X_WORLD	RA of matched star (SExtractor)
Y_WORLD	Dec of matched star (SExtractor)
MAG	Instrumental magnitude (SExtractor)
RA	Catalog RA of matched star
DEC	Catalog Dec of matched star
MAG_U	Catalog U magnitude of matched star
MAG_B	Catalog B magnitude of matched star
MAG_V	Catalog V magnitude of matched star
MAG_R	Catalog R magnitude of matched star
MAG_I	Catalog I magnitude of matched star

Table 5.15.5: *Star match table entries.*

Master: After master bias subtraction, prescan and overscan regions are trimmed away from the standard stars field frame.

Zmaster: After master bias subtraction the overscan correction is applied before trimming away the overscan regions.

CleanBadPixel: Bad pixel correction on the reduced standard stars field exposure. If this option is turned on, a bad pixel table should be specified in the input SOF (see Table 5.15.1). The bad pixel correction algorithm is described in Section 7.1, page 121.

CleanCosmic: Cosmic ray events removal from input standard stars field exposure. The cosmic ray rejection algorithm is described in Section 7.2, page 122.

ComputeQC: If this parameter is set, *Quality Control* (QC) parameters will be computed and written partly to the header of the output galaxy table and partly to the header of the reduced standard field frame (depending on the dataset used for their computation). They will also be written to two output QC PAF files named `qc0000.paf` and `qc0001.paf`. These files are not classified as pipeline recipe products, as they are intermediate datasets that in the standard pipeline operations would be translated into new entries in the QC log file. The QC parameters computed by *vmimstandard* are the same that are computed by the pipeline recipe *vmimobsstare* (see Section 5.18, page 80), with some extra ones that are all written to the header of the reduced standard stars field image, and to the file `qc0001.paf`:

`QC ZEROPOINT NSTARS:` Number of stars used in the frame zeropoint computation.

`QC ZEROPOINT:` Gain corrected frame zeropoint. This is computed as a robust estimate of the values Z_i obtained for all the identified standard stars:

$$Z_i = \Delta M_i + E \cdot A + C \cdot C_i$$

where, for a given star i, ΔM_i is the observed difference between the catalog magnitude and the instrumental magnitude in the appropriate band, C_i is the known colour index, A is the airmass,

Parameter	**Possible values**	**Explanation**
BiasMethod	Master Zmaster	Bias removal with no overscan correction Bias removal with overscan correction
CleanBadPixel	true false	Clean bad pixels Do not clean bad pixels
CleanCosmic	true false	Clean cosmic rays Do not clean cosmic rays
CosmicThreshold	*float*	Sigmas above level discriminator
CosmicRatio	*float*	Peak/neighbours discriminator
ComputeQC	true false	Compute QC parameters Do not compute QC parameters
MagLimit	*float*	Max magnitude for star selection
StarIndex	*float*	Index used for star/galaxy discrimination
SearchRadius	*float* (arcsec)	Search radius used in stars identification
KSigmaClip	*float* (sigma)	Sigma clipping factor used in star matching
MagInitial	*float*	Magnitude tolerance for initial stars selection
MagFinal	*float*	Magnitude tolerance for final stars selection
MinStars	*int*	Minimum number of matching stars required
TemperatureCheck	true false	Check beam and ambient temperatures consistency Do not check consistency of temperatures
TemperatureTolerance	*float* (^oC)	Max difference beam - ambient temperatures
ReduceAnyFrame	true false	Any input image is reduced Reduce only if pointing to standard stars field
SExtractor.FilterName	*file*	SExtractor convolution mask
SExtractor.SExtractor	*file*	SExtractor executable
SExtractor.StarNnwName	*file*	SExtractor neural network weights
SExtractor.Window	*file*	Image region where SExtractor is applied

Table 5.15.6: *vmimstandard parameters.*

read from the keyword `PRO AIRMASS` of the input frame header, E is the atmospheric extinction coefficient at the considered band, read from the photometric table (see Table 5.15.2), and C is the colour term. The estimate Z is computed using a biweight estimator function taken from the Rostat program written by T.Beers for robust statistics on a data set (see Beers, Flynn, Gebhardt 1990, AJ 100, 32). This estimate is then corrected for the instrument gain:

$$Z_g = Z + 2.5log_{10}g$$

where g is the CCD gain factor in e^-/ADU. The Z frame zeropoint, not normalised to the gain factor and more useful for data analysis purposes, is saved to the header of the reduced frame at the keyword `PRO MAG ZERO`, together with its error `PRO MAGZERO RMS`.

`QC ZEROPOINT RMS`: Error on gain corrected frame zeropoint as obtained from the biweight estimator function.

CosmicRatio: Critical ratio for discriminating between objects and cosmic rays. This parameter is effective when *CleanCosmic* is set.

CosmicThreshold: Threshold for the selection of cosmic rays candidates. This parameter is effective when *CleanCosmic* is set.

KSigmaClip: Number of sigmas used in the spatial rejection of matched stars. Stars found within the specified *SearchRadius* are compared to their expected catalog positions, and the sigma of the residuals distribution is computed. Then, all the stars having a distance from the expected position greater than the specified threshold are rejected.

MagFinal: Magnitude tolerance applied to matched stars, after the K-sigma clipping has been applied. The stars magnitudes should not differ from the corresponding catalog magnitudes more than this value.

MagInitial: Magnitude tolerance applied in star matching.

MagLimit: Limiting magnitude of objects to be matched with the standard stars from the input catalog.

MinStars: Minimum required number of stars for the creation of a star match table.

ReduceAnyFrame: Normally the *vmimstandard* recipe would attempt to reduce any dataset classified as a standard stars field exposure. However, during the time critical on-line processing, it may be appropriate not to reduce systematically all the incoming frames. This is because the same standard stars field is exposed once for each VIMOS quadrant, and to reduce images from the temporarily unused quadrants is not a requirement. Setting this parameter to *false* would prevent the processing of such images. For an off-line processing this parameter would be typically set to *true*.

SearchRadius: Max distance (in arcseconds) from the expected position of a catalog star.

StarIndex: This index can have a value between 0 (minimal stellarity) and 1 (maximal stellarity). Only the identified objects having a stellarity index greater than the specified value will be taken as stars, to be matched with the standard stars from the input catalog.

TemperatureCheck: For associating the detected stars with the reference catalog, their coordinates must be corrected applying the CCD to Sky distortion corrected for changes in the beam temperature (see Sections 6.2.2 and 7.13). If this parameter is set, the beam temperature is compared with the temperatures of the other beams and with the ambient temperature. If the difference between the beam temperature and the ambient temperature is greater than the threshold specified by *TemperatureTolerance*, then the temperature used will be the mean of the beams temperatures that differ from the ambient temperature less than the specified threshold. If not a single beam temperature is within tolerance, then the ambient temperature is used in place of the beam temperature. This option was just introduced to prevent possible problems with the beam temperature sensors.

TemperatureTolerance: Max tolerated difference between ambient and beam temperatures. This parameter is effective only if *TemperatureCheck* is set.

A description of the algorithms used in this recipe is given in Section 7.14, page 131.

5.16 vmimcalphot

The VIMOS pipeline recipe *vmimcalphot* is used to determine night zeropoints, atmospheric extinction coefficients, and colour terms, from a set of star match tables produced by the recipe *vmimstandard* (see Section 5.15, page 72). The star match tables may refer to different standard star fields, but they must all be derived from exposures made with the same filter and the same quadrant.

The files to be included in the input SOF are all listed in Table 5.16.1.

DO category	Type	Explanation	Required
IMG_STAR_MATCH_TABLE	Pipeline product	List of identified standard stars	√
PHOTOMETRIC_TABLE	Calibration	Photometric table	√

Table 5.16.1: *Input files for the vmimcalphot recipe.*

The standard photometric tables in the calibration directories are named `ipc_`f`i.`q`.tfits` (where f is the filter name and q the quadrant number).

The only product of this recipe is an upgraded PHOTOMETRIC_TABLE (Table 5.15.2, page 72), carrying the newly computed zeropoint, and, if requested, new extinction and colour coefficients.

File name	DO category	Type	Explanation
photometric_table.tfits	PHOTOMETRIC_TABLE	FITS	Upgraded photometric table

Table 5.16.2: *Product of the vmimcalphot recipe.*

The *vmimcalphot* parameters are listed in Table 5.16.3.

Parameter	Possible values	Explanation
Extinction	true false	Compute the extinction coefficient Do not compute the extinction coefficient
ColorTerm	true false	Compute the colour term Do not compute the colour term
UseColorTerm	true false	Use colour term in zeropoint computation Do not use colour term in zeropoint computation

Table 5.16.3: *vmimcalphot parameters.*

A more complete description of the parameters meaning is also given:

ColorTerm: If both this and the *Extinction* parameters are set, the difference between the catalog magnitude and the instrumental magnitude of all stars is seen as a function of the airmass and of the star colour index; a first order bivariate linear fit is then made to derive simultaneously the extinction coefficient and the colour term (beside the magnitude zeropoint). Alternatively, if *Extinction* is false, the difference between the catalog magnitude and the instrumental magnitude is seen just as a function of the star colour index, and in this case a linear fit between colour index and magnitude difference is made. If the computation of the extinction coefficient is requested, at least four star match tables obtained from observations at different airmasses should be specified in input. Otherwise, it is advisable to specify just one input star match table (*i.e.*, with all stars at the same airmass), containing at least four stars with different colour indexes.

Extinction: The case in which both this and the *ColorTerm* parameters are set is described above. Alternatively, if *ColorTerm* is false, the difference between the catalog magnitude and the instrumental magnitude is seen just as a function of the star airmass, and in this case a linear fit between airmass and magnitude difference is made. At least four star match tables obtained from observations made at different airmasses should be specified in input.

UseColorTerm: This parameter is only effective if both the *ColorTerm* and the *Extinction* parameters are off. In this case, before being averaged, the difference between the catalog magnitude and the instrumental magnitude is corrected for the atmospheric extinction. If the *UseColorTerm* parameter is set, the magnitude difference is also corrected for the colour index of each star. The estimates of the colour and the atmospheric extinction coefficients used for these corrections are read from the input photometric table.

A description of the algorithms used in this recipe is given in Section 7.15, page 131.

5.17 vmimpreimaging

The VIMOS pipeline recipe *vmimpreimaging* is used to apply basic reduction steps to the imaging observation that is preliminary to a MOS observation of the same field. No source detection is attempted on the image. The image WCS, together with the component describing the instrument optical distortions (see Section 6.2.2, page 114), is converted into the convention followed by VMMPS (the VIMOS mask preparation software). This set of coefficients, the so-called CO-matrix used by the SAO WCSTools package [10], is written to the header of the reduced image.

The files to be included in the input SOF are listed in Table 5.17.1.

DO category	Type	Explanation	Required
IMG_PREIMAGING	Raw frame	Preimaging exposure	√
MASTER_BIAS	Calibration	Master bias	√
IMG_MASTER_SKY_FLAT	Calibration	Master sky flat field	√
CCD_TABLE	Calibration	Bad pixel table	
PHOTOMETRIC_TABLE	Calibration	Photometric table	

Table 5.17.1: *Input files for the vmimpreimaging recipe.*

The bad pixel table needs to be specified only if the cleaning of bad pixels is requested. If a photometric table is specified, the magnitude zeropoint, the atmospheric extinction coefficient, and the colour term (see Table 5.15.2, page 72) are copied from the photometric table to the header of the reduced image.

The products of the *vmimpreimaging* recipe are indicated in Table 5.17.2.

The *vmimpreimaging* parameters are listed in Table 5.17.3.

File name	DO category	Type	Explanation
img_science_reduced.fits	IMG_SCIENCE_REDUCED	FITS	Reduced preimaging exposure

Table 5.17.2: *Product of the vmimpreimaging recipe.*

Parameter	Possible values	Explanation
BiasMethod	Master Zmaster	Bias removal with no overscan correction Bias removal with overscan correction
CleanBadPixel	true false	Clean bad pixels Do not clean bad pixels
CleanCosmic	true false	Clean cosmic rays Do not clean cosmic rays
CosmicThreshold	*float*	Sigmas above level discriminator
CosmicRatio	*float*	Peak/neighbours discriminator
TemperatureCheck	true false	Check beam and ambient temperatures consistency Do not check consistency of temperatures
TemperatureTolerance	*float* (^oC)	Max difference beam - ambient temperatures

Table 5.17.3: *vmimpreimaging parameters.*

A more complete description of the parameters is also given here:

BiasMethod: Method for bias removal from the input exposure. The bias removal procedure is described in some detail in Section 7.3. Possible settings are:

Master: After master bias subtraction, prescan and overscan regions are trimmed away from the input frame.

Zmaster: After master bias subtraction the overscan correction is applied before trimming away the overscan regions.

CleanBadPixel: Bad pixel correction on the reduced exposure. If this option is turned on, a bad pixel table should be specified in the input SOF (see Table 5.15.1). The bad pixel correction algorithm is described in Section 7.1, page 121.

CleanCosmic: Cosmic ray events removal from input raw frame. The cosmic ray rejection algorithm is described in Section 7.2, page 122.

CosmicRatio: Critical ratio for discriminating between objects and cosmic rays. This parameter is effective when *CleanCosmic* is set.

CosmicThreshold: Threshold for the selection of cosmic rays candidates. This parameter is effective when *CleanCosmic* is set.

TemperatureCheck: The Sky to CCD distortion models (see 6.2.2, page 114) may require to be corrected for thermal expansion effects on the camera and the CCD, before converting them into the CO-matrix convention. If this parameter is set, the beam temperature is compared with the temperatures of the other beams and with the ambient temperature. If the difference between the beam temperature and the ambient temperature is greater than the threshold specified by *TemperatureTolerance*, then the temperature used will be the mean of the beams temperatures that differ from the ambient temperature less than the specified threshold. If not a single beam temperature is within tolerance, then the ambient temperature is used in place of the beam temperature. This option was just introduced to prevent possible problems with the beam temperature sensors.

TemperatureTolerance: Max tolerated difference between ambient and beam temperatures. This parameter is effective only if *TemperatureCheck* is set.

A description of the algorithms used in this recipe is given in Section 7.16, page 131.

5.18 vmimobsstare

The VIMOS pipeline recipe *vmimobsstare* is used to apply basic reduction steps to one exposure made in direct imaging mode. SExtractor v2.1.6 [8] is run on the reduced image, producing a table of detected objects with their instrumental magnitudes, their celestial and image coordinates, and their stellarity index. The image WCS, together with the component describing the instrument optical distortions (see Section 6.2.2, page 114), is converted into the convention followed by VMMPS (the VIMOS mask preparation software). This set of coefficients, the CO-matrix used in the SAO WCSTools package [10], is written to the header of the reduced image.

The files to be included in the input SOF are listed in Table 5.18.1.

DO category	**Type**	**Explanation**	**Required**
IMG_SCIENCE	Raw frame	Science exposure	√
MASTER_BIAS	Calibration	Master bias	√
MASTER_DARK	Calibration	Master dark	
IMG_MASTER_SKY_FLAT	Calibration	Master sky flat field	√
CCD_TABLE	Calibration	Bad pixel table	
PHOTOMETRIC_TABLE	Calibration	Photometric table	

Table 5.18.1: *Input files for the vmimobsstare recipe.*

The bad pixel table needs to be specified only if the cleaning of bad pixels is requested. In the calibration directories there is one CCD_TABLE file for each quadrant, named `badpixel.`q`.tfits` (where q is the quadrant number). Care should be made in selecting the appropriate bad pixel tables for imaging and spectral instrument modes, that have the same names but are located in separated directories.

If a photometric table is specified, the magnitude zeropoint, the atmospheric extinction coefficient, and the colour term (see Table 5.15.2, page 72) are copied from the photometric table to the header of the reduced image. The standard photometric tables in the calibration directories are named `ipc_`fi.q`.tfits` (where f is the filter name and q the quadrant number).

The products of the *vmimobsstare* recipe are indicated in Table 5.18.2.

File name	DO category	Type	Explanation
img_science_reduced.fits	IMG_SCIENCE_REDUCED	FITS	Reduced imaging exposure
img_galaxy_table.tfits	IMG_GALAXY_TABLE	FITS	List of detected objects

Table 5.18.2: *Product of the vmimobsstare recipe.*

The galaxy table is the output of SExtractor [8] converted into FITS format, and it is shown in Table 5.18.3. The content of this table is defined within the VIMOS pipeline DRS, with the only intent of serving the purpose of the involved pipeline recipes. Some of the table columns listed contain data necessary for the computation of QC parameters (see ahead), while objects positions and magnitudes are necessary for the identification of standard photometric and astrometric stars in the *vmimstandard* and the *vmskyccd* recipes. For a customised output it is always possible to run SExtractor separately, after configuring the native SExtractor parameter files. For a complete explanation of the output parameters please refer to the SExtractor documentation (that can be found in `http://terapix.iap.fr/rubrique.php?id_rubrique=91/index.html`).

Column name	Explanation
`NUMBER`	Object number
`MAG_ISOCOR`	Corrected isophotal magnitude
`MAGERR_ISOCOR`	RMS error on corrected isophotal magnitude
`MAG_APER`	Fixed-aperture magnitude
`MAGERR_APER`	RMS error on fixed-aperture magnitude
`MAG_AUTO`	Automatic-aperture magnitude
`MAGERR_AUTO`	RMS error on automatic-aperture magnitude
`MAG_BEST`	`MAG_AUTO` if no neighbours, otherwise `MAG_ISOCOR`
`MAGERR_BEST`	Error on instrumental magnitude
`X_IMAGE`	Object X pixel position
`Y_IMAGE`	Object Y pixel position
`X_WORLD`	Object RA
`Y_WORLD`	Object Dec
`ISOAREA_WORLD`	Area of lowest isophote (arcsec^2)
`A_IMAGE`	2^{nd} order moment along the major axis (pixel)
`B_IMAGE`	2^{nd} order moment along the minor axis (pixel)
`A_WORLD`	2^{nd} order moment along the major axis (arcsec)
`B_WORLD`	2^{nd} order moment along the minor axis (arcsec)
`FWHM_IMAGE`	FWHM (pixel) of mean radial profile (gaussian fit)
`FWHM_WORLD`	FWHM (arcsec) of mean radial profile
`THETA_IMAGE`	PA of major axis (counter-clockwise from X axis)
`ERRTHETA_IMAGE`	Error on Position Angle
`ELLIPTICITY`	`1 - B_IMAGE / A_IMAGE`
`CLASS_STAR`	Stellarity index, 0.0 = galaxy, 1.0 = star
`FLAGS`	Extraction flag, different from 0 in case of error

Table 5.18.3: *Galaxy table entries.*

Parameter	Possible values	Explanation
BiasMethod	Master Zmaster	Bias removal with no overscan correction Bias removal with overscan correction
CleanBadPixel	true false	Clean bad pixels Do not clean bad pixels
CleanCosmic	true false	Clean cosmic rays Do not clean cosmic rays
CosmicThreshold	*float*	Sigmas above level discriminator
CosmicRatio	*float*	Peak/neighbours discriminator
TemperatureCheck	true false	Check beam and ambient temperatures consistency Do not check consistency of temperatures
TemperatureTolerance	*float* (^oC)	Max difference beam - ambient temperatures
StarIndex	*float*	Min stellarity index for stars selection
ComputeQC	true false	Compute QC parameters Do not compute QC parameters
SExtractor.FilterName	*file*	SExtractor convolution mask
SExtractor.SExtractor	*file*	SExtractor executable
SExtractor.StarNnwName	*file*	SExtractor neural network weights
SExtractor.Window	*file*	Image region where SExtractor is applied

Table 5.18.4: *vmimobsstare parameters.*

The *vmimobsstare* parameters are listed in Table 5.18.4. Only few parameters of the SExtractor group are listed here: almost all the entries of the SExtractor configuration file, `$PIPE_HOME/vimos/config/vimos.sex`, can be overruled by the SExtractor group parameter values specified in the `vmimobsstare.rc` configuration file. Conventionally, a SExtractor configuration parameter name matches the *vmimobsstare* configuration parameter obtained by lowercasing all of its characters with the exception of the first character of each word, and removing the underscore. For instance, the SExtractor parameter `CATALOG_TYPE` matches the *vmimobsstare* parameter `CatalogType`. When using the corresponding command line option, the SExtractor group must be explicitly specified. For instance, the command option corresponding to the parameter `CatalogType` is `--SExtractor.CatalogType`.

The `Window` parameter has no counterpart in the SExtractor configuration file, and is used to restrict the SExtractor operations to just a part of the input image. This is useful to avoid false detections along the vignetted image region.

A more complete description of the parameters not belonging to the SExtractor group is given here:

BiasMethod: Method for bias removal from the input exposure. The bias removal procedure is described in some detail in Section 7.3. Possible settings are:

Master: After master bias subtraction, prescan and overscan regions are trimmed away from the input frame.

Zmaster: After master bias subtraction the overscan correction is applied before trimming away the overscan regions.

CleanBadPixel: Bad pixel correction on the reduced exposure. If this option is turned on, a bad pixel table should be specified in the input SOF (see Table 5.15.1). The bad pixel correction algorithm is described in Section 7.1, page 121.

CleanCosmic: Cosmic ray events removal from input raw frame. The cosmic ray rejection algorithm is described in Section 7.2, page 122.

ComputeQC: If this parameter is set, *Quality Control* (QC) parameters will be computed and written partly to the header of the output galaxy table and partly to the header of the reduced standard field frame (depending on the dataset used for their computation). They will also be written to two output QC PAF files named `qc0000.paf` and `qc0001.paf`. These files are not classified as pipeline recipe products, as they are intermediate datasets that in the standard pipeline operations would be translated into new entries in the QC log file.

The QC parameters computed from the contents of the galaxy table are based only on the objects with `FLAGS` = 0. In particular, all the objects in the galaxy table having a stellarity index greater than the value specified at the parameter *StarIndex* are here called "stars" for simplicity:

`QC IMAGE QUALITY:` The image quality is computed as a robust estimate of the parameter `FWHM_WORLD` of all the stars in the galaxy table. Starting from the median value of `FWHM_WORLD`, the standard deviation from this value is used to exclude outliers, and to compute an improved estimate of the seeing. This operation is then iterated, up to four times.

`QC IMAGE QUALITY ERROR:` The error on `QC IMAGE QUALITY` is given as the population standard deviation of the `FWHM_WORLD` values contributing to the final mean.

`QC STAR COUNT:` Number of stars in galaxy table.

`QC STELLARITY MEAN:` Mean stellarity index of all objects in the galaxy table.

`QC STELLARITY RMS:` Population standard deviation of the stellarity indexes of all objects in the galaxy table.

`QC STAR STELLARITY MEAN:` Mean stellarity index of all stars in the galaxy table.

`QC STAR STELLARITY RMS:` Population standard deviation of the stellarity indexes of all stars in the galaxy table.

`QC STAR ELLIPTICITY MEAN:` Mean value of the parameter `ELLIPTICITY` of all the stars in the galaxy table.

`QC STAR ELLIPTICITY RMS:` Population standard deviation of the `ELLIPTICITY` of all the stars in the galaxy table.

`QC STAR ORIENTATION MEAN:` Mean orientation of star ellipses, from -90 to $+90$ degrees, counted counterclockwise, with 0 corresponding to the image X axis. This parameter is determined by a peak detection algorithm run on an histogram of all objects orientations. Initially, an approximate position of the most probable orientation is determined. Then a new histogram is built, centred on this position and with a number of bins dependent on the number of objects available, and a more accurate peak detection algorithm is run again. If no peak is detected, this parameter is assigned the value zero, associated to an error of ± 90 degrees (this is good for plots).

`QC STAR ORIENTATION RMS:` Sigma of mean orientation of star ellipses. This is the uncertainty on the position of the maximum of a gaussian fit to the detected peak. If no peak were detected, this parameter would be assigned the value of 90 degrees.

`QC MAGLIM`: Limiting magnitude. All the stars in the galaxy table are selected, and a histogram of their magnitudes is built. Bin sizes from 0.1 up to 1.0 magnitudes are tried, until the most populated bin contains at least 20 stars. The position of the most populated bin is taken as the limiting magnitude. If the 20 stars level for the highest bin is never reached, the limiting magnitude value is set to 0.0.

`QC MAGLIM ERROR`: The error is taken as the value of the bin size of the histogram used in the determination of the limiting magnitude. If no limiting magnitude were found, the error would be set to 0.0.

The only QC parameters computed from the reduced image are the following, and are written to the header of the reduced image itself:

`QC SKYBACK`: The sky background (in ADU/s) is evaluated dividing the central 1600x1800 region of the chip into 10x10 regions 160x180 pixels each. For each region the median level is computed. The mean of the 10 lowest values obtained is the estimation of the sky background level. This value is scaled to the unit of time.

`QC SKYBACK ERROR`: The error on `QC SKYBACK` is taken as the population RMS of the 10 values used in the estimation of the sky background level.

CosmicRatio: Critical ratio for discriminating between objects and cosmic rays. This parameter is effective when *CleanCosmic* is set.

CosmicThreshold: Threshold for the selection of cosmic rays candidates. This parameter is effective when *CleanCosmic* is set.

StarIndex: This parameter is only effective when *ComputeQC* is set. All the galaxy table objects with a stellarity index greater than the specified value are taken as stars.

TemperatureCheck: The Sky to CCD distortion models (see 6.2.2, page 114) may require to be corrected for thermal expansion effects on the camera and the CCD, before converting them into the CO-matrix convention. If this parameter is set, the beam temperature is compared with the temperatures of the other beams and with the ambient temperature. If the difference between the beam temperature and the ambient temperature is greater than the threshold specified by *TemperatureTolerance*, then the temperature used will be the mean of the beams temperatures that differ from the ambient temperature less than the specified threshold. If not a single beam temperature is within tolerance, then the ambient temperature is used in place of the beam temperature. This option was just introduced to prevent possible problems with the beam temperature sensors.

TemperatureTolerance: Max tolerated difference between ambient and beam temperatures. This parameter is effective only if *TemperatureCheck* is set.

A description of the algorithms used in this recipe is given in Section 7.17, page 132.

5.19 vmimobsjitter

The VIMOS pipeline recipe *vmimobsjitter* is used to apply basic reduction steps to a sequence of exposures made in direct imaging mode, and to combine them in a single image. Each input image is processed in the same way as by recipe *vmimobsstare*, therefore what characterises the *vmimobsjitter* is just the final combination of the input frames. The input and the output files are the same listed in the Tables 5.18.1 and 5.18.2, page 80. The *vmimobsjitter* parameters are listed in Table 5.19.1.

Parameter	**Possible values**	**Explanation**
BiasMethod	Master Zmaster	Bias removal with no overscan correction Bias removal with overscan correction
CleanBadPixel	true false	Clean bad pixels Do not clean bad pixels
Resampling	BiLinear BiCubic	Pixel value bilinear interpolation Pixel value bicubic interpolation
StackMethod	Average Median MinMax Ksigma Auto	Average combination of reduced images Median combination of reduced images Min-max combination of reduced images K-sigma clipping combination of reduced images Optimal combination of reduced images
KSigmaLow	*float* (sigma)	Low threshold for K-sigma clipping method
KSigmaHigh	*float* (sigma)	High threshold for K-sigma clipping method
MaxRejection	*int*	Number of highest values excluded in rejection stack method
MinRejection	*int*	Number of lowest values excluded in rejection stack method
TemperatureCheck	true false	Check beam and ambient temperatures consistency Do not check consistency of temperatures
TemperatureTolerance	*float* (^oC)	Max difference beam - ambient temperatures
StarIndex	*float*	Min stellarity index for stars selection
ComputeQC	true false	Compute QC parameters Do not compute QC parameters
SExtractor.FilterName	*file*	SExtractor convolution mask
SExtractor.SExtractor	*file*	SExtractor executable
SExtractor.StarNnwName	*file*	SExtractor neural network weights
SExtractor.Window	*file*	Image region where SExtractor is applied

Table 5.19.1: *vmimobsjitter parameters.*

Only few parameters of the SExtractor group are listed here: almost all the entries of the SExtractor configuration file, `$PIPE_HOME/vimos/config/vimos.sex`, can be overruled by the SExtractor group parameter values specified in the `vmimobsjitter.rc` configuration file. Conventionally, a SExtractor configuration parameter name matches the *vmimobsjitter* configuration parameter obtained by lowercasing all of its characters with the exception of the first character of each word, and removing the underscore. For instance, the SExtractor parameter `CATALOG_TYPE` matches the *vmimobsstare* parameter `CatalogType`. When using the corresponding command line option, the SExtractor group must be explicitly specified. For instance, the command option corresponding to the parameter `CatalogType` is `--SExtractor.CatalogType`.

The `Window` parameter has no counterpart in the SExtractor configuration file, and is used to restrict the SExtractor operations to just a part of the input image. This is useful to avoid false detections along the vignetted image region.

A more complete description of the parameters not belonging to the SExtractor group is given here:

BiasMethod: Method for bias removal from the input exposures. The bias removal procedure is described in some detail in Section 7.3. Possible settings are:

- **Master:** After master bias subtraction, prescan and overscan regions are trimmed away from the input frames.
- **Zmaster:** After master bias subtraction the overscan correction is applied before trimming away the overscan regions.

CleanBadPixel: Bad pixel correction on each reduced exposure, before combination. If this option is turned on, a bad pixel table should be specified in the input SOF (see Table 5.15.1). The bad pixel correction algorithm is described in Section 7.1, page 121.

ComputeQC: If this parameter is set, *Quality Control* (QC) parameters will be computed. The QC parameters are computed on the combined image, and are the same parameters that are computed in recipe *vmimobsstare* (see Section 5.18, page 80).

KSigmaHigh: Number of standard deviations above the median pixel value for rejecting a pixel value when *StackMethod* is set to *"Ksigma"*.

KSigmaLow: Number of standard deviations below the median pixel value for rejecting a pixel value when *StackMethod* is set to *"Ksigma"*.

MaxRejection: Number of highest pixel values to be rejected when *StackMethod* is set to *"MinMax"*.

MinRejection: Number of lowest pixel values to be rejected when *StackMethod* is set to *"MinMax"*.

Resampling: Method used for interpolating pixel values from a single reduced frame to the common pixelisation defined for the combined image. Possible settings are:

- **BiLinear:** Bilinear interpolation.
- **BiCubic:** Bicubic interpolation.

StackMethod: Method used for combination of reduced images. See Section 7.6 for a complete description of all the combination methods. Possible settings are:

- **Auto:** Given the number of input frames, an optimal frame combination method is selected. Currently this is always going to the method *"Average"*.
- **Average:** The combined frame is the mean of the input frames.
- **Ksigma:** The combined frame is the mean of the input frames, after K-sigma screening of pixel values. The number of sigma to be applied in the rejection is specified by the parameters *KSigmaLow* and *KSigmaHigh*.

Median: The combined frame is the median of the input frames.

MinMax: The combined frame is the mean of the input frames, after rejection of minimum and maximum values. The number of values to reject is specified by the parameters *MinRejection* and *MaxRejection*.

StarIndex: This parameter is only effective when *ComputeQC* is set. All the galaxy table objects with a stellarity index greater than the specified value are taken as stars.

TemperatureCheck: The Sky to CCD distortion models (see 6.2.2, page 114) may require to be corrected for thermal expansion effects on the camera and the CCD, before converting them into the CO-matrix convention. If this parameter is set, the beam temperature is compared with the temperatures of the other beams and with the ambient temperature. If the difference between the beam temperature and the ambient temperature is greater than the threshold specified by *TemperatureTolerance*, then the temperature used will be the mean of the beams temperatures that differ from the ambient temperature less than the specified threshold. If not a single beam temperature is within tolerance, then the ambient temperature is used in place of the beam temperature. This option was just introduced to prevent possible problems with the beam temperature sensors.

TemperatureTolerance: Max tolerated difference between ambient and beam temperatures. This parameter is effective only if *TemperatureCheck* is set.

A description of the algorithms used in this recipe is given in Section 7.18, page 132.

5.20 vmspflat

The VIMOS pipeline recipe *vmspflat* is used to create the MOS normalised master flat field from a set of MOS flat field exposures. A *not* normalised master flat field is also required by the recipe *vmspcaldisp* to determine the spectral curvature model (see Section 6.3.3, page 117). The recipe *vmspflat* does compute its own curvature model, but this is generally incompatible with the wavelength calibration and the optical distortion models computed by *vmspcaldisp*. For this reason this curvature model is just used internally in the process of flat field normalisation, where the compatibility with the Y component of the optical distortion model is irrelevant (see Sections 6.3.2 through 6.3.4).

The files to be included in the input SOF are listed in Table 5.20.1.

DO category	Type	Explanation	Required
MOS_SCREEN_FLAT	Raw frame	Flat field exposures	$\surd$
MASTER_BIAS	Calibration	Master bias	$\surd$
MASTER_DARK	Calibration	Master dark	
GRISM_TABLE	Calibration	Grism table	$\surd$
CCD_TABLE	Calibration	Bad pixel table	

Table 5.20.1: *Input files for the vmspflat recipe.*

At least one raw flat field exposure should be present in the input SOF. The acquisition of input flat fields may be done using different mask shutter settings (to avoid contamination between different spectral orders in LR grisms used with multiplexed masks, see Figure 5.21.1, page 96). The *vmspflat* recipe will properly combine all the input frames according to a specified method[6].

The bad pixel table needs to be specified only if the cleaning of bad pixels is requested. In the calibration directories there is one CCD_TABLE file for each quadrant, named `badpixel.`q`.tfits` (where q is the quadrant number). Care should be made in selecting the appropriate bad pixel tables for imaging and spectral instrument modes, that have the same names but are located in separated directories.

The grism table is required by all VIMOS pipeline MOS recipes. It contains necessary information to control the way spectra are extracted, and the determination of the spectral distortion models. The *vmspflat* recipe gets from the grism table the wavelength that should be used as reference (header entry `PRO WLEN CEN`), and the spectrum extension in CCD pixels above and below the position of the reference wavelength (header entries `PRO SPECT LLEN LO` and `PRO SPECT LLEN HI`) (see Figure 6.3.1, page 118). In the calibration directories there is one GRISM_TABLE file for each quadrant/grism combination, named `grs_`$grism$`.`q`.tfits` (where $grism$ is the grism name, and q is the quadrant number).

The products of the *vmspflat* recipe are indicated in Table 5.20.2.

File name	**DO category**	**Type**	**Explanation**
mos_master_screen_flat.fits	MOS_MASTER_SCREEN_FLAT	FITS	Master flat field
mos_combined_screen_flat.fits	MOS_COMBINED_SCREEN_FLAT	FITS	Combined flat field
vmCrvOpt-q.paf		PAF	CRV and OPT models

Table 5.20.2: *Products of the vmspflat recipe.*

The PAF file just contains the curvature and the optical distortion models computed in the process of spectra normalisation. Such models are available just for debug purposes, and are not to be used in further data reduction steps.

The *vmspflat* parameters are listed in Table 5.20.3.

A more complete description of the used parameters meaning is given here:

BiasMethod: Method for bias removal from the input spectral flat field frames. The bias removal procedure is described in some detail in Section 7.3. Possible settings are:

> **Master:** After master bias subtraction, prescan and overscan regions are trimmed away from the flat field frame.
>
> **Zmaster:** After master bias subtraction the overscan correction is applied before trimming away the overscan regions.

CleanBadPixel: Bad pixel correction on the output master spectral flat field. If this option is turned on, a bad pixel table should be specified in the input SOF (see Table 5.20.1). The bad pixel correction algorithm is described in Section 7.1, page 121.

CleanCosmic: Cosmic ray events removal from input raw spectral flat field frames. The cosmic ray rejection algorithm is described in Section 7.2, page 122.

[6] This has not yet been tested with real data.

Parameter	**Possible values**	**Explanation**
BiasMethod	Master Zmaster	Bias removal with no overscan correction Bias removal with overscan correction
CleanBadPixel	true false	Clean bad pixels Do not clean bad pixels
CleanCosmic	true false	Clean cosmic rays Do not clean cosmic rays
ComputeQC	true false	Compute QC parameters Do not compute QC parameters
FlatMethod	Average Median Polynomial	Model flat field trends by average filtering Model flat field trends by median filtering Model flat field trends by polynomial fitting
Fuzz	*int* (pixel)	Extra X pixels in spectral extraction
KSigmaLow	*float*	Low threshold for K-sigma clipping stack method
KSigmaHigh	*float*	High threshold for K-sigma clipping stack method
MaxRejection	*int*	Number of highest values excluded in rejection stack method
MinRejection	*int*	Number of lowest values excluded in rejection stack method
PolyOrderX	*float*	Degree of polynomial for X trend removal
PolyOrderY	*float*	Degree of polynomial for Y trend removal
SmoothBoxSize	*int*	Filter box size used in trend removal
StackMethod	Average Ksigma MinMax Median	Combined flat field is the average of the input frames Flat fields are combined with a K-sigma clipping method Flat fields are combined with a min-max rejection method Flat fields are combined with a median method

Table 5.20.3: *vmspflat parameters.*

ComputeQC: If this parameter is set, *Quality Control* (QC) parameters will be computed and written to the header of the output normalised master flat field and to an output QC PAF file named `qc0000.paf`. This file is not classified as a pipeline recipe product, as it is an intermediate dataset that in the standard pipeline operations would be translated into new entries in the QC log file. Currently the QC parameters computed by *vmspflat* are:

`QC MOS SLIT WIDTH:` Width (in mm) of slit closest to mask centre.

`QC MOS FLAT FLUX:` The position of the slit closest to the mask centre is determined. For this slit the position of the reference wavelength is determined applying the available spectral distortion models. The total counts in the rectangular region long as the slit length in pixels, 5 CCD pixels wide, and centred at the reference wavelength position, are then bias subtracted and divided by the area of the slit and by the exposure time. The flux is given in ADU $s^{-1}mm^{-2}$.

`QC MOS FLAT FLUXERR:` The total counts in the rectangular region where the flux was determined are square-rooted, and then normalised to the unit of area and time.

FlatMethod: Method used in modeling each slit spectrum, for removing the large scale trends related to the CCD intrinsic response with changing wavelength. Possible settings are:

Average: Average filter. The smooth box is defined by the parameter *SmoothBoxSize*.

Median: Median filter. The smooth box is defined by the parameter *SmoothBoxSize*.

Polynomial: Polynomial fitting of spectrum. This method gives generally bad results, and its use is not advisable. The order of the fitting polynomial is specified by the parameters *PolyOrderX* and *PolyOrderY*.

Fuzz: Extra number of X CCD pixels, to be added at the spectra sides during the extraction. This parameter is just used when flat fields acquired with different mask shutter positions are input to the recipe (see Section 7.19, page 133).

KSigmaHigh: Number of standard deviations above the median pixel value for rejecting a pixel value when *StackMethod* is set to *"Ksigma"*.

KSigmaLow: Number of standard deviations below the median pixel value for rejecting a pixel value when *StackMethod* is set to *"Ksigma"*.

MaxRejection: Number of highest pixel values to be rejected when *StackMethod* is set to *"MinMax"*.

MinRejection: Number of lowest pixel values to be rejected when *StackMethod* is set to *"MinMax"*.

PolyOrderX: Degree of polynomial used for fitting the flat spectrum in the cross-dispersion direction. Used if *FlatMethod* is set to "Polynomial".

PolyOrderY: Degree of polynomial used for fitting the flat spectrum in the dispersion direction. Used if *FlatMethod* is set to "Polynomial".

SmoothBoxSize: Size of the running box used for smoothing, expressed as number of pixels along the dispersion direction. This parameter is ignored if *FlatMethod* is set to "Polynomial".

StackMethod: Combination method of input flat fields taken with the same mask shutters position. See Section 7.6 for a complete description of all the combination methods. Possible settings are:

Average: The combined flat field is the mean of the input frames.

Ksigma: The combined flat field is the mean of the input frames, after K-sigma screening of pixel values. The number of sigma to be applied in the rejection is specified by the parameters *KSigmaLow* and *KSigmaHigh*.

Median: The combined flat field is the median of the input frames.

MinMax: The combined flat field is the mean of the input frames, after rejection of minimum and maximum values. The number of values to reject is specified by the parameters *MinRejection* and *MaxRejection*.

A description of the algorithms used in this recipe is given in Section 7.19, page 133.

5.21 vmspcaldisp

The VIMOS pipeline recipe *vmspcaldisp* is used to determine all the spectral distortions and transformations (see Section 6.3, pages 116-120) from an arc lamp exposure and a master flat field produced by the recipe *vmspflat*.

The files to be included in the input SOF are listed in Table 5.21.1.

DO category	**Type**	**Explanation**	**Required**
MOS_ARC_SPECTRUM	Raw frame	Arc lamp exposure	$\surd$
MASTER_BIAS	Calibration	Master bias	$\surd$
MASTER_DARK	Calibration	Master dark	
MOS_MASTER_SCREEN_FLAT	Calibration	Normalised flat field	
MOS_COMBINED_SCREEN_FLAT	Calibration	Combined flat field	
GRISM_TABLE	Calibration	Grism table	$\surd$
LINE_CATALOG	Calibration	Line catalog	$\surd$
CCD_TABLE	Calibration	Bad pixel table	

Table 5.21.1: *Input files for the vmspcaldisp recipe.*

At least one raw arc lamp exposure should be present in the input SOF. In general the acquisition of input arc lamp exposures may be done using different mask shutter settings (to avoid contamination between different spectral orders in LR grisms used with multiplexed masks, see Figure 5.21.1, page 96). The *vmspcaldisp* recipe will properly combine all the input frames according to a specified method[7].

The normalised and the combined flat fields are the products of the recipe *vmspflat* run on flat field data obtained with the same mask. Neither of them is required for running *vmspcaldisp*, but if a combined flat field is not given then no spectral curvature model can be computed, and a "first guess" is used in its place. A normalised master flat field needs to be specified only if a flat field correction is requested.

The bad pixel table needs to be specified only if the cleaning of bad pixels is requested. In the calibration directories there is one CCD_TABLE file for each quadrant, named `badpixel.`q`.tfits` (where q is the quadrant number). Care should be made in selecting the appropriate bad pixel tables for imaging and spectral instrument modes, that have the same names but are located in separated directories.

The grism table is required by all VIMOS pipeline MOS recipes. It contains necessary information to control the way spectra are extracted and the determination of the distortion models. The *vmspcaldisp* recipe gets from the grism table the wavelength that should be used as reference (header entry `PRO WLEN CEN`), and the spectrum extension in CCD pixels above and below the position of the reference wavelength (header entries `PRO SPECT LLEN LO` and `PRO SPECT LLEN HI`) (see Figure 6.3.1, page 118). Other parameters, used in the construction of the extracted arc lamp slit spectra (see Table 5.21.2), are the start and the end wavelength of the image containing the extracted spectra (header entries `PRO WLEN START` and `PRO WLEN END`), and the step of the sampling along the dispersion direction (header entry `PRO WLEN INC`). In the calibration directories there is one GRISM_TABLE file for each quadrant/grism combination, named `grs_`$grism$`.`q`.tfits` (where $grism$ is the grism name, and q is the quadrant number).

[7] This has not yet been tested with real data.

The line catalogues in the calibration directories are also named `lcat_`$grism$`.`q`.tfits` although there is not a real dependency from the quadrant number.

All the products of the *vmspcaldisp* recipe are indicated in Table 5.21.2.

File name	**DO category**	**Type**	**Explanation**
extract_table.tfits	EXTRACT_TABLE	FITS	Extraction table
mos_arc_spectrum_extracted.fits	MOS_ARC_SPECTRUM_EXTRACTED	FITS	Extracted slit spectra
MOS_wavecal_$grism$_q.paf		PAF	Distortion models

Table 5.21.2: *Products of the vmspcaldisp recipe.*

Column name	**Explanation**
SLIT	Slit number
Y	Slit row coordinate in image of extracted slit spectra
CCD_X	X position of reference wavelength on CCD
CCD_Y	Y position of reference wavelength on CCD
MASK_X	x mask coordinate along the slit
MASK_Y	y mask coordinate along the slit
SPEC_NO	Number of exposures contributing to this row
CRV_POL_i	i^{th} coefficient of local spectral curvature model
CRVPOL_RMS	RMS of curvature model residuals (*unused*)
INV_DIS_i	i^{th} coefficient of local IDS
INVDIS_RMS	RMS of local IDS (CCD pixel)
DIS_QUAL	Quality of local IDS (1 = good, 0 = bad)
ZERO_X	X position of zero order contamination (*unused*)
ZERO_Y	Y position of zero order contamination (*unused*)

Table 5.21.3: *Extraction table entries.*

The primary recipe product is the extraction table, that contains information about the local modeling of spectral distortions. On Table 5.21.3 the meaning of each table column is given (IFU related columns are not listed, being the IFU data reduction not yet operative). Column Y holds the position of any spectral row (see Figure 6.3.1, page 118) in the produced image of extracted spectra. The (x, y) mask coordinates matching the (X, Y) CCD coordinates (as derived by applying the optical distortion model, see Section 6.3.2), corresponding to a point on a slit, are also listed in columns MASK_X, MASK_Y, CCD_X, and CCD_Y. The "local" curvature model coefficients are extracted from the global curvature solution. In fact, a local curvature model can be computed just where a slit spectrum edge is detected. Currently the column CRVPOL_RMS is left unused, and in future it will be likely filled just for the first and the last row of each slit spectrum (see Figure 6.3.1, page 118). The local IDS coefficients are not available in case of a failure of the local polynomial fit, as indicated in column DIS_QUAL.

A secondary product of the *vmspcaldisp* recipe is the image of extracted slit spectra, that allows a visual check of the distortion models quality. Failures of the spectral curvature model would be made apparent by widening gaps between the extracted spectra: if the slit spectra edges were followed accurately no gap should be visible at any wavelength. The stability of the wavelength calibration could also be evaluated by the alignment of the calibration lamp emission lines from all slits.

A last product is the PAF file carrying all the information related to the spectral distortions. This PAF file is copied (or moved) to the product directory, and it is identical to the IWS configuration file `MOS_wavecal_`$grism\_q$`.cmf` (where q indicates the VIMOS quadrant number, and $grism$ the grism name) that is created in the directory where *vmspcaldisp* is launched.

The *vmspcaldisp* parameters are listed in table 5.21.4.

Parameter	**Possible values**	**Explanation**
ApplyFlatField	true false	Apply flat field correction Do not apply flat field correction
ArcExtraction	Input Local Global	Extract using first guess distortion models Extract using fitted models and local IDS Extract using fitted models and global IDS
BiasMethod	Master Zmaster	Bias removal with no overscan correction Bias removal with overscan correction
CleanBadPixel	true false	Clean bad pixels Do not clean bad pixels
CleanCosmic	true false	Clean cosmic rays Do not clean cosmic rays
ComputeQC	true false	Compute QC parameters Do not compute QC parameters
ExtractionWindow	*int*	Size of arc line search window (pixel)
Fuzz	*int* (pixel)	Search radius in spectrum edge detection
ModelSlit	true false	Model wavelength solution within each slit Do not model local IDSs within each slit
ModelSlitOrder	*int*	Polynomial order for IDS modeling within each slit
RefineIDS	true false	Pretuning of the first guess IDS before line identification Do not use pretuning of the first guess IDS

Table 5.21.4: *vmspcaldisp parameters.*

A more complete description of the used parameters meaning is given here:

ApplyFlatField: If this parameter is set, the flat field correction is applied to the input arc lamp frames. In this case a normalised master flat field must be specified in the input SOF (see table 5.21.1).

ArcExtraction: This parameter is used to control the way the image of extracted slit spectra is created. Three different methods are currently available:

Global: The slit spectra extraction is performed by applying the improved distortion models computed by the recipe. The global IDS models are applied in the wavelength linearisation.

Input: The slit spectra extraction is performed by just applying the “first guess” spectral distortion models that are read from the arc lamp frame header. In this case the *vmspcaldisp* recipe does not try to fit improved distortion models based on the available data (and the output VIMOS IWS configuration file is just a copy of the input “first guess”).

Local: The slit spectra extraction is performed by applying the improved distortion models computed by the recipe. The local IDS models are applied in the wavelength linearisation.

BiasMethod: Method for bias removal from the input arc lamp exposures. The bias removal procedure is described in some detail in Section 7.3. Possible settings are:

Master: After master bias subtraction, prescan and overscan regions are trimmed away from the arc lamp frame.

Zmaster: After master bias subtraction the overscan correction is applied before trimming away the overscan regions.

CleanBadPixel: Bad pixel correction on the combined arc lamp frame used for optical distortion and inverse dispersion models determination. If this option is turned on, a bad pixel table should be specified in the input SOF (see Table 5.21.1). The bad pixel correction algorithm is described in Section 7.1, page 121.

CleanCosmic: Cosmic ray events removal from input arc lamp frames. The cosmic ray rejection algorithm is described in Section 7.2, page 122.

Parameter	**Explanation**
`QC MOS SLIT WIDTH`	Width of slit (mm)
`QC MOS HE LAMBDA`	He arc lamp line for flux determination (Angstrom)
`QC MOS HE FLUX`	Flux at chosen He wavelength (ADU $mm^{-2}s^{-1}$)
`QC MOS HE FLUXERR`	Error on flux at chosen He wavelength (ADU $mm^{-2}s^{-1}$)
`QC MOS NE LAMBDA`	Ne arc lamp line for flux determination (Angstrom)
`QC MOS NE FLUX`	Flux at chosen Ne wavelength (ADU $mm^{-2}s^{-1}$)
`QC MOS NE FLUXERR`	Error on flux at chosen Ne wavelength (ADU $mm^{-2}s^{-1}$)
`QC MOS AR LAMBDA`	Ar arc lamp line for flux determination (Angstrom)
`QC MOS AR FLUX`	Flux at reference wavelength (ADU $mm^{-2}s^{-1}$)
`QC MOS AR FLUXERR`	Error on flux at reference wavelength (ADU $mm^{-2}s^{-1}$)
`QC MOS WAVECAL COEFF`i	Median of IDS coefficient d_i (see Section 6.3.4)
`QC MOS REFWAVE MEAN`	Mean CCD offset of reference wavelength on all slits (pixel)
`QC MOS REFWAVE RMS`	RMS of CCD offsets of reference wavelength on all slits (pixel)
`QC MOS RESOLUTION1 LAMBDA`	Line used in spectral resolution determination (Angstrom)
`QC MOS RESOLUTION1`	Mean spectral resolution at red end of spectrum
`QC MOS RESOLUTION1 RMS`	RMS of spectral resolution at red end of spectrum
`QC MOS RESOLUTION2 LAMBDA`	Line used in spectral resolution determination (Angstrom)
`QC MOS RESOLUTION2`	Mean spectral resolution at centre of spectrum
`QC MOS RESOLUTION2 RMS`	RMS of spectral resolution at centre of spectrum
`QC MOS RESOLUTION3 LAMBDA`	Line used in spectral resolution determination (Angstrom)
`QC MOS RESOLUTION3`	Mean spectral resolution at blue end of spectrum
`QC MOS RESOLUTION3 RMS`	RMS of spectral resolution at blue end of spectrum
`QC MOS IDS RMS`	Global IDS rms (pixel)

Table 5.21.5: *QC parameters of the vmspcaldisp recipe.*

ComputeQC: If this parameter is set, QC parameters will be computed and written to the header of the extraction table (see Table 5.21.2) and to an output QC PAF file named `qc0000.paf`. This file is not classified as a pipeline recipe product, as it is an intermediate dataset that in the standard pipeline operations would be translated into new entries in the QC log file.

The QC parameters computed by *vmspcaldisp* are listed in Table 5.21.5. All parameters refer to the slit closest to the mask centre, unless differently specified. Fluxes are determined at the position of a chosen arc lamp line from each working lamp, and they are measured in ADU $s^{-1}mm^{-2}$. The error on fluxes is just the computed theoretical statistical noise. The spectral resolution at different wavelengths is computed as the mean of the resolutions obtained for each spectral row of each slit in the mask (see Figure 6.3.1, page 118). The spectral resolution is computed here as $\lambda/\Delta\lambda$, where $\Delta\lambda$ is the FWHM of the arc lamp line used for the measurement. Finally the RMS of the IDS model, `QC MOS IDS RMS`, is not derived from the RMSs of the single IDS local polynomial fits, but is computed as the RMS of the deviation of the observed lines peaks on the image of the extracted slit spectra from their expected positions. In this way systematic errors would not be excluded from the computation.

ExtractionWindow: Size of the arc line search window on the CCD. The window is centred on an arc lamp line expected position. If a negative value is given, then the search window size is computed by

$$\Delta Y = 3(wb_{01} + 2)$$

where ΔY is the search window size in CCD pixels, w is the slit width in millimetres, and b_{01} is the conversion factor between millimetres on the mask and pixels on the CCD along the dispersion direction, obtained from the optical distortion model (see Table 6.3.1, page 117).

Fuzz: Search radius for spectral edges, used in the determination of the spectral curvature model based on the input combined flat field. See Section 7.20, page 7.20, for more details.

ModelSlit: The trend of the local IDS coefficients within a single slit can be modeled by a polynomial fit. This may reduce the effect of outlying local solutions, before the global IDS fit is tried.

ModelSlitOrder: The order of the polynomial used in the slit IDS modeling when *ModelSlit* is set.

RefineIDS: When this parameter is set, the "first guess" IDS polynomial valid for a given spectral row is slightly modified, searching for a best match of the catalog wavelengths with the local spectrum arc lines. This pretuning operation permits a more robust arc lamp lines identification, but it should be avoided in presence of severe light contamination from reflections and/or spectral orders overlap (see Figure 5.21.1).

A description of the algorithms used in this recipe is given in Section 7.20, page 134.

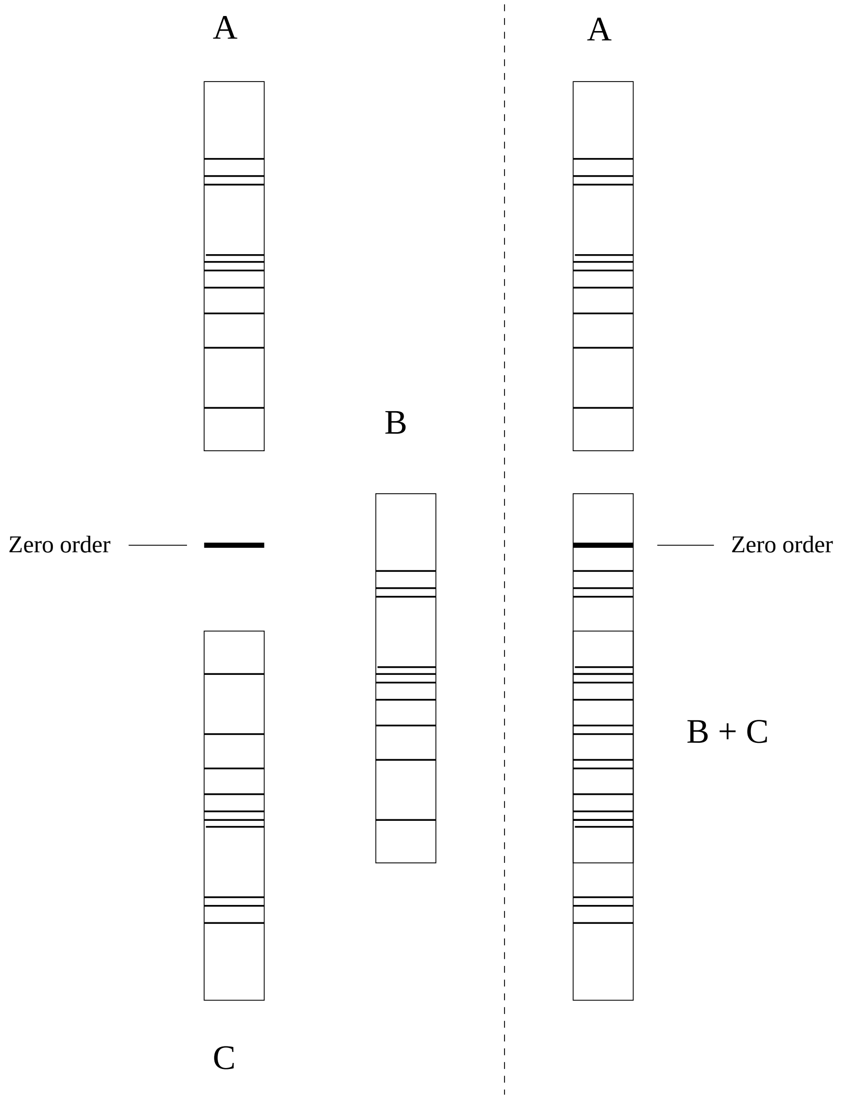


Figure 5.21.1: *Contamination from orders* 0 *and* -1 *in multiplexed spectra. On the left, the first order slit spectra A and B are shown, together with the* 0 *and the* -1 *orders of spectrum A. If spectra A and B are multiplexed, as shown on the right, spectrum B is contaminated by the* 0 *and* -1 *orders of spectrum A.*

5.22 vmmosobsstare

The VIMOS pipeline recipe *vmmosobsstare* is used to apply basic reduction steps to one exposure made in MOS mode, to locate objects, and to optimally extract their spectra.

The files to be included in the input SOF are listed in Table 5.22.1.

DO category	Type	Explanation	Required
MOS_SCIENCE	Raw frame	Science exposure	$\surd$
MASTER_BIAS	Calibration	Master bias	$\surd$
MASTER_DARK	Calibration	Master dark	
MOS_MASTER_SCREEN_FLAT	Calibration	Normalised flat field	
EXTRACT_TABLE	Calibration	Extraction table	
GRISM_TABLE	Calibration	Grism table	$\surd$
CCD_TABLE	Calibration	Bad pixel table	

Table 5.22.1: *Input files for the vmspcaldisp recipe.*

The bad pixel table needs to be specified only if the cleaning of bad pixels is requested. In the calibration directories there is one CCD_TABLE file for each quadrant, named `badpixel.`q`.tfits` (where q is the quadrant number). Care should be made in selecting the appropriate bad pixel tables for imaging and spectral instrument modes, that have the same names but are located in separated directories.

A flat field correction is applied only if a normalised master flat field (produced by the recipe *vmspflat*) is specified.

The extraction table is the product of the local spectral distortions modeling performed by the recipe *vmspcaldisp* (see table 5.21.3, page 92). If an extraction table is not specified, then the global distortion models read from the science frame header are used (see Section 6.3, pages 116-120).

The grism table contains necessary information to control the way spectra are extracted, starting from the reference wavelength (header entry `PRO WLEN CEN`), on a specific range of pixels above and below its position on the CCD (header entries `PRO SPECT LLEN LO` and `PRO SPECT LLEN HI`) (see Figure 6.3.1, page 118). Other parameters, used in the extraction of the science slit spectra (see Table 5.22.2), are the start and the end wavelength of the image of the extracted slit spectra (header entries `PRO WLEN START` and `PRO WLEN END`), and the step of the sampling along the dispersion direction (header entry `PRO WLEN INC`). Finally, the wavelengths of the sky lines used in the alignment of the spectral distortion models, necessary to keep into account the possible coordinates shifts introduced by a variation of the instrument flexures between the science and the calibration exposures, are listed in the header keywords `PRO SKY WLEN`n, with n ranging from 1 to the number specified in the keyword `PRO SKY NO`. In the calibration directories there is one GRISM_TABLE file for each quadrant/grism combination, named `grs_`$grism$`.`q`.tfits` (where $grism$ is the grism name, and q is the quadrant number).

All the products of the *vmmosobsstare* recipe are indicated in Table 5.22.2.

The slit spectra are remapped with the instrument distortions removed and at a fixed wavelength step. A sky value is estimated for each wavelength and then subtracted from the data. The result is stored in the `MOS_SCIENCE_EXTRACTED` image, while the image `MOS_SCIENCE_SKY` contains the sky model that

File name	**DO category**	**Type**	**Explanation**
mos_science_reduced.fits	MOS_SCIENCE_REDUCED	FITS	Objects spectra
mos_science_extracted.fits	MOS_SCIENCE_EXTRACTED	FITS	Sky subtracted slit spectra
mos_science_sky.fits	MOS_SCIENCE_SKY	FITS	Sky spectra
object_table.tfits	OBJECT_TABLE	FITS	Objects spectra identification
window_table.tfits	WINDOW_TABLE	FITS	Objects positions in slit

Table 5.22.2: *Product of the vmmosobsstare recipe.*

was subtracted from the resampled data. The positions of the extracted slit spectra and of the detected objects they may contain are listed in the window table, 5.22.3.

Column name	**Explanation**
`SLIT`	Slit number in ADM
`SPEC_LEN`	*unused*
`SPEC_START`	Image bottom row of slit spectrum (pixel)
`SPEC_END`	Image top row of slit spectrum (pixel)
`OBJ_START`	Object start position from slit spectrum bottom (pixel)
`OBJ_END`	Object end position from slit spectrum bottom (pixel)
`OBJ_NO`	Object counter within a slit
`OBJ_POS`	Object peak position from slit spectrum bottom (pixel)
`OBJ_WIDTH`	*unused*
`OBJ_X`	Mask x coordinate of object (mm)
`OBJ_Y`	Mask y coordinate of object (mm)
`OBJ_RA`	*unused*
`OBJ_DEC`	*unused*

Table 5.22.3: *Window table entries. The IFU entries are not listed, because IFU data reduction is not yet supported. The positions of the slit spectra are counted from the image bottom (with the first row at $Y = 0$), while the objects positions are counted from the bottom row of the slit to which they belong.*

The extracted spectra of the detected objects are stored in the rows of the `MOS_SCIENCE_REDUCED` image, that is the primary product of this recipe. The object table, shown in Table 5.22.4, indicates from what slit each spectrum was extracted.

The *vmmosobsstare* parameters are listed in Table 5.22.5.

A more complete description of the used parameters meaning is given here:

BiasMethod: Method for bias removal from the input science exposure. The bias removal procedure is described in some detail in Section 7.3. Possible settings are:

Master: After master bias subtraction, prescan and overscan regions are trimmed away from the science frame.

Zmaster: After master bias subtraction the overscan correction is applied before trimming away the overscan regions.

Column name	Explanation
SLIT	Slit number in ADM
Y	Image row of extracted spectrum
OBJ_NO	Object counter within same slit
OBJ_X	Mask x coordinate of object
OBJ_Y	Mask y coordinate of object
OBJ_RA	*unused*
OBJ_DEC	*unused*

Table 5.22.4: *Object table entries. The IFU entries are not listed, because IFU data reduction is not yet supported. The Y pixel positions of the extracted spectra are counted from the image bottom, with the first row at $Y = 0$.*

CleanBadPixel: Bad pixel correction on the input science frame. If this option is turned on, a bad pixel table should be specified in the input SOF (see table 5.22.1). The bad pixel correction algorithm is described in Section 7.1, page 121.

DetectionLevel: In the object detection task, only the pixel values that are *DetectionLevel* noise sigmas above the background level are included in the objects candidates. The object detection is tried on the mean cross-dispersion profile of the extracted slit spectra. The mean profile is computed on the wavelength interval specified by the grism table entries `PRO AVG POS` and `PRO AVG RNG`.

Fuzz: Extra number of X CCD pixels, to be added at the slit spectra sides during the slit spectra extraction as a safety margin.

HorneExtraction: If this parameter is set, the Horne's method for spectral optimal extraction (K.Horne (1986), PASP 98, p.609) is applied to the objects detected in the extracted slit spectra[8]. Otherwise a simple aperture extraction is used.

LineWidth: Search radius (in pixels along the dispersion direction) from the expected position of the sky lines in the input science frame. This parameter is only effective when *UseSkylines* is set.

MaxObjectSize: In the object detection process, all the pixel values of the slit mean cross-dispersion profile that are above the threshold specified at *DetectionLevel* are selected. If any unbroken sequence of selected pixels is longer than *MaxObjectSize,* an attempt to unblend it into sub-objects is made.

MinObjectSize: In the object detection process, all the pixel values of the slit mean cross-dispersion profile that are above the threshold specified at *DetectionLevel* are selected. Any unbroken sequence of selected pixels must be at least *MinObjectSize* pixels long to be accepted as an object candidate.

ModelSlit: If this parameter is set, and an extraction table is specified in input, the coefficients of the local IDS solutions contained in the extraction table (see Table 5.21.3, page 92) are modeled within each slit by a polynomial fit. The model values for the coefficients are replaced in the extraction table and then used in the spectral extraction process. This is a way to smooth out outlying IDS local solutions.

[8] To be rigorous, Horne's extraction is not intended to be applied to resampled data, where the noise is correlated.

Parameter	Possible values	Explanation
BiasMethod	Master Zmaster	Bias removal with no overscan correction Bias removal with overscan correction
CleanBadPixel	true false	Clean bad pixels Do not clean bad pixels
ModelSlit	true false	Model global IDS within each slit Apply wavelength solution from extraction table
ModelSlitOrder	*int*	Polynomial order for global IDS within each slit
UseSkylines	true false	Use sky lines to align wavelength calibration Do not align wavelength calibration
SkyMethod	Median Fit	Sky at object position is median of all sky values Sky at object position is interpolation of sky values
PolyOrder	*int*	Order of sky interpolating polynomial
Fuzz	*int* (pixel)	Extra X pixels in slit spectra extraction
SlitMargin	*int* (pixel)	Excluded pixels at slit ends for sky determination
LineWidth	*int* (pixel)	Pixels to read around expected sky line position
DetectionLevel	*float* (sigma)	Object detection level in terms of noise sigmas
WatershedLevels	*int*	No. of levels in watershed method for objects unblending
WatershedFraction	*float*	Minimum flux fraction of unblended objects
MinObjectSize	*int* (pixel)	Minimum allowed object size
MaxObjectSize	*int* (pixel)	Minimum interval where to attempt objects unblending
HorneExtraction	true false	Optimal (Horne's) spectral extraction Aperture spectral extraction

Table 5.22.5: *vmmosobsstare parameters.*

ModelSlitOrder: This parameter is only effective when *ModelSlit* is set and an extraction table is specified in input. The order of the polynomial used in the global IDS solutions within each slit is specified. In particular, if this parameter is set to zero the local IDS coefficients are replaced by their mean values within each slit.

PolyOrder: Order of the polynomial used in sky level modeling, when *SkyMethod* is set to *Fit*.

SkyMethod: Method used for the sky determination at the position of the detected objects. The sky level is determined for each sampled wavelength of the extracted slit spectra. The modeled slit sky spectra are written to the output `MOS_SCIENCE_SKY` image. Possible settings are:

Fit: The pixel values outside the regions where objects were detected are fitted by a polynomial. The model values are then taken as the sky level[9]. The order of the polynomial used is specified at the parameter *PolyOrder*.

Median: The median of the pixel values outside the regions where objects were detected is taken as the sky level.

[9] This method is quite sensitive to the presence of cosmic rays, and it should be avoided on uncleaned images.

SlitMargin: Number of pixels at the edges of the extracted slit spectra, to be excluded from the determination of the sky level and from the object detection task.

UseSkylines: If this parameter is set, the sky lines listed in the grism table are searched and identified in the input science exposure. The median offset from their expected positions along the dispersion direction is taken as a measure of the variation of the instrument flexure between the science exposure and the flat field and arc lamp exposures used for calibration. This offset is added to the Y component of the optical distortion model (see Section 6.3.2, page 116) before its usage in the spectral extraction task.

WatershedFraction: In the object detection process, an attempt of unblending into sub-objects an extended (*i.e.,* greater than *MaxObjectSize*) object candidate is made. A candidate sub-object must have at least the fraction of the total flux specified here, in order to be flagged as a separate object.

WatershedLevels: Number of levels used in watershed method applied to object unblending. See parameter *MaxObjectSize* about object unblending.

A description of the algorithms used in this recipe is given in Section 7.21, page 149.

5.23 vmmosobsjitter

The VIMOS pipeline recipe *vmmosobsjitter* is used to apply basic reduction steps to a sequence of exposures made in MOS mode, to combine them in a single image, to locate objects, and to optimally extract their spectra. Each input image is processed in the same way as by recipe *vmmosobsstare*, therefore what mainly characterises the *vmmosobsjitter* recipe is the task of combining the input frames. In the current pipeline release, the stacking method used is always the same, *i.e.,* the median of frames (see Section 7.6, page 123). The input and the output files are the same as listed in the Tables 5.22.1 and 5.22.2, page 97. The *vmmosobsjitter* configuration file is also identical to the one used by the *vmmosobsstare* recipe, shown in table 5.22.5, page 100.

A description of the algorithms used in this recipe is given in Section 7.22, page 150.

5.24 vmifucalib

The VIMOS pipeline recipe *vmifucalib* is used to determine the spectral extraction mask, the wavelength calibration, and the fibers relative transmission correction, from a set of flat field and one arc lamp exposures.

The files to be included in the input SOF are listed in Table 5.24.1.

At least one flat field exposure should be present in the input SOF, but if an arc lamp exposure is not given, then only the spectral extraction mask can be determined.

If an arc lamp exposure is given in input, a line catalog must also be given.

The fiber identification file is optional: it consists of intensity profiles (one for each IFU pseudo-slit) cut along the cross-dispersion direction of a flat field exposure, where the fiber spectra have been safely identified. The fibers corresponding to the peak positions of each profile are listed in the tables included in the FITS file extensions. Such safe identifications would then be transferred to the new input flat fields by cross-correlation. In the calibration directories there is one IFU_IDENT file for each quadrant/grism combination,

DO category	**Type**	**Explanation**	**Required**
IFU_SCREEN_FLAT	Raw frame	Flat field exposure	$\surd$
IFU_ARC_SPECTRUM	Raw frame	Arc lamp exposure	
MASTER_BIAS	Calibration	Master bias	$\surd$
LINE_CATALOG	Calibration	Line catalog	
IFU_IDENT	Calibration	Fiber identification	
CCD_TABLE	Calibration	Bad pixel table	

Table 5.24.1: *Input files for the vmifucalib recipe.*

named `ifu_ident_`$grism$`_`q`.fits` (where q indicates the VIMOS quadrant number, and $grism$ the grism name).

If the fiber identification file is not specified, the fiber spectra identification is still attempted, but the result is not always correct. A fiber misidentification would appear later on the reconstructed image of the field-of-view (generated by the *vmifuscience* recipe) as zig-zagging patterns breaking the generally smooth look of the intensity distribution.

The spectral distortions (coded in the extraction mask) are always recomputed from scratch by tracing the flat field spectra, and then by wavelength-calibrating the extracted arc lamp spectra. Contrary to what happens in the MOS data reduction task, the spectral distortion models contained in the data headers are ignored.

The bad pixel table needs to be specified only if the cleaning of bad pixels is requested. In the calibration directories there is one CCD_TABLE file for each quadrant, named `badpixel.`q`.tfits` (where q is the quadrant number). Care should be made in selecting the appropriate bad pixel tables for imaging and spectral instrument modes, that have the same names but are located in separated directories.

The line catalogues in the calibration directories are named `lcat_`$grism$`.`q`.tfits` (where $grism$ is the grism name, and q the quadrant number although there is no actual dependency from the quadrant number).

All the products of the *vmifucalib* recipe are indicated in Table 5.24.2.

File name	**DO category**	**Type**	**Explanation**
ifu_master_screen_flat.fits	IFU_MASTER_SCREEN_FLAT	FITS	Combined flats
ifu_arc_spectrum_extracted.fits	IFU_ARC_SPECTRUM_EXTRACTED	FITS	Extracted arc spectra
ifu_flat_spectrum_extracted.fits	IFU_FLAT_SPECTRUM_EXTRACTED	FITS	Extracted flat spectra
ifu_trace.tfits	IFU_TRACE	FITS	Extraction mask
ifu_ids.tfits	IFU_IDS	FITS	Wavelength calibration
ifu_transmission.tfits	IFU_TRANSMISSION	FITS	Transmission correction

Table 5.24.2: *Products of the vmifucalib recipe.*

The extracted spectra are stored in the output images in a conventional order, with blue on the left and red on the right side. The images have 400 rows in the case of MR and HR observations, and 1600 rows in the case of LR observations. The spectra starting from the left side of each pseudo-slit are stored starting from the bottom rows of the output images. In the case of LR observations, the first 400 spectra from the pseudo-slit 1 are at the bottom, and the last 400 spectra from the pseudo-slit 4 are at the top.

Each image row corresponds to an IFU fiber position on the IFU head. This correspondence is described in a set of 8 tables located in the calibration directories (see Table 5.24.3). A subset of 4 tables refers to LR observations, with 1600 spectra per quadrant, and they are named `ifutableLR`q`.fits` (where q indicates the VIMOS quadrant number). A second subset of 4 tables should be used for MR and HR observations, with 400 spectra per quadrant, and they are named `ifutableHR`q`.fits`.

Column name	**Explanation**
ROW	Image row, counted from the bottom starting from 1.
L	X coordinate on the IFU head, counted from left, ranging from 1 to 80.
M	Y coordinate on the IFU head, counted from bottom, ranging from 1 to 80.

Table 5.24.3: *IFU position table entries.*

The content of the calibration tables generated by the recipe *vmifucalib* is described in Tables 5.24.4, 5.24.5, and 5.24.6.

Column name	**Explanation**
Ci	i^{st} coefficient of the spectrum tracing polynomial.
RMS	Standard deviation of polynomial fit.

Table 5.24.4: *IFU extraction mask.*

In the extraction mask `ifu_trace.tfits`, there are two table extensions for each active IFU pseudo-slit. Each table includes the coefficients of 400 polynomial fits, one for each fiber spectrum, starting from the first spectrum at the left end of a pseudo-slit. The first table extension of each pair is the actual extraction mask, obtained by polynomial fitting of the tracings on the whole spectral range. The second table extension is just a linear fitting of the tracing on a short range, used in the alignment of the extraction mask to the scientific spectra.

Column name	**Explanation**
Ci	i^{st} coefficient of the inverse dispersion polynomial.
RMS	Standard deviation of polynomial fit.
NLINES	Number of identified arc lamp lines used in fit.

Table 5.24.5: *Inverse dispersion solution.*

In the inverse dispersion solution `ifu_ids.tfits` there is one table extensions for each active IFU pseudo-slit. Each table includes the coefficients of 400 polynomial fits, one for each fiber spectrum, starting from the first spectrum at the left end of a pseudo-slit.

Column name	Explanation
TRANS	Relative transmission factor of fiber.

Table 5.24.6: *Relative transmission factors for all fibers.*

The `ifu_transmission.tfits` table includes the fiber-to-fiber relative transmission correction factors of 400 (in case of MR or HR observations) or 1600 (in case of LR observations) fiber spectra, starting from the first spectrum at the left end of the first pseudo-slit.

The *vmifucalib* parameters are listed in Table 5.24.7.

Parameter	Possible values	Explanation
AllowSingleFrames	true false	A single input flat is also allowed More than one input flat is required
StackMethod	Average Median MinMax Ksigma Auto	Master flat is average of input flats Master flat is median of input flats Master flat is obtained with min-max rejection Master flat is obtained with K-sigma clipping Optimal combination of input flats
KSigmaLow	*float* (sigma)	Low threshold for K-sigma clipping method
KSigmaHigh	*float* (sigma)	High threshold for K-sigma clipping method
MinRejection	*int*	No. of lowest rejected values for rejection method
MaxRejection	*int*	No. of highest rejected values for rejection method
BiasMethod	Master Zmaster	Bias removal with no overscan correction Bias removal with overscan correction
CleanBadPixel	true false	Clean bad pixels Do not clean bad pixels
ApplyTransmission	true false	Apply transmission correction to extracted spectra Do not apply transmission correction
MaxIdsRms	*float* (pixel)	Maximum tolerated RMS of residuals in IDS fit
MaxTraceRejection	*int*	Maximum percentage of rejected positions in tracing
ComputeQC	true false	Compute QC parameters Do not compute QC parameters

Table 5.24.7: *vmifucalib parameters.*

A more complete description of the used parameters meaning is given here:

AllowSingleFrames: If this parameter is set, then a master flat field is produced also from a single input flat field exposure. In this case the *StackMethod* is ignored.

ApplyTransmission: If this parameter is set, then the computed fiber-to-fiber relative transmission correction factors are applied to all the extracted spectra.

BiasMethod: Method for bias removal from the input frames. The bias removal procedure is described in some detail in Section 7.3. Possible settings are:

Master: After master bias subtraction, prescan and overscan regions are trimmed away from the processed frame.

Zmaster: After master bias subtraction the overscan correction is applied before trimming away the overscan regions.

CleanBadPixel: Bad pixel correction on the master flat field. If this option is turned on, a bad pixel table should be specified in the input SOF (see Table 5.21.1, page 91). The bad pixel correction algorithm is described in Section 7.1, page 121.

ComputeQC: If this parameter is set, *Quality Control* (QC) parameters will be computed and written to the header of the output tables, and to three output QC PAF files named `qc0000.paf`, `qc0001.paf`, and `qc0001.paf`. These files are not classified as pipeline recipe products, as they are intermediate datasets that in the standard pipeline operations would be translated into new entries in the QC log file. Currently the QC parameters computed by *vmifucalib* are:

`QC IFU LOSTi`: Number of fibers that could not be traced on pseudo-slit i.

`QC IFU TRACEi RMS`: Mean value of the RMS of the polynomial fitting obtained on each traced IFU spectrum on pseudo-slit i.

`QC IFU IDS RMS`: RMS of the IDS residuals, evaluated on the image of extracted arc lamp spectra.

`QC IFU RESOLUTIONj LAMBDA`: Wavelength of the arc lamp line chosen for determining the spectral resolution in the red ($j = 1$), central ($j = 2$), and blue ($j = 3$) spectral regions.

`QC IFU RESOLUTIONj`: Spectral resolution in the red ($j = 1$), central ($j = 2$), and blue ($j = 3$) spectral regions, averaged on all spectra, evaluated on an arc lamp spectrum line dependent on the used grism. The spectral resolution is computed as the ratio between the arc lamp line wavelength, and its FWHM.

`QC IFU RESOLUTIONj RMS`: RMS of spectral resolution determined in the red ($j = 1$), central ($j = 2$), and blue ($j = 3$) spectral regions.

`QC IFU WAVECALj COEFFi`: Median i^{th} coefficient of the inverse dispersion solutions on pseudo-slit j, with $i = 1, 2, ..., n$ (where n is the degree of the polynomial used). In the case of MR and HR observations, this parameter is just computed for the active slit ($j = 2$).

`QC IFU TRACEj COEFFi`: Median i^{th} coefficient of the fiber spectra tracing solutions on pseudo-slit 1, with $i = 1, 2, ..., n$ (where n is the degree of the polynomial used). In the case of MR and HR observations, this parameter is just computed for the active slit ($j = 2$).

`QC IFU REFROWj`: The reference row is the Y pixel position on the CCD where, for a given pseudo-slit j, each fiber spectrum is detected, identified, and conventionally begun to be traced. This parameter is reported here, because referenced by other IFU QC1 parameters.

`QC IFU TRACEj CENTRAL`: On pseudo-slit j, this is the sequence number of the active fiber closest to the central CCD X pixel at the reference row (see `QC IFU REFROWi`).

`QC IFU TRACEj SLOPE`: On pseudo-slit j, a linear fit is made to the tracing of the central spectrum (see `QC IFU TRACEi CENTRAL`), on a 400 pixels interval centered on the reference row (see `QC IFU REFROWi`). In absence of optical distortions, a perfect grism alignment would correspond to a zero slope.

`QC IFU FLUX LAMBDAi`: The flat field flux (see `QC IFU FLUX MEAN`) is measured on a wavelength interval starting ($i = 1$) and ending ($i = 2$) at the specified values.

`QC IFU FLAT FLUX`: The mean integrated signal, per fiber, per second, within the specified wavelength interval, is computed on all active pseudo-slits.

KSigmaHigh: Number of standard deviations above the median pixel value for rejecting a pixel value when *StackMethod* is set to *"Ksigma"*.

KSigmaLow: Number of standard deviations below the median pixel value for rejecting a pixel value when *StackMethod* is set to *"Ksigma"*.

MaxIdsRms: Maximum tolerated RMS of residuals in IDS fit (pixel). In the determination of the wavelength calibration, any polynomial fit not better than indicated, will be rejected.

MaxRejection: Number of highest pixel values to be rejected when *StackMethod* is set to *"MinMax"*.

MinRejection: Number of lowest pixel values to be rejected when *StackMethod* is set to *"MinMax"*.

MaxTraceRejection: Maximum percentage of rejected positions in fiber spectra tracing. In the fiber tracing operation, a number of pixel positions may be rejected because the detected position outlays the general trend, or because the signal level is too low. When the percentage of rejected positions is more than what is specified here, then the corresponding fiber is flagged as "dead" and excluded from further processing.

StackMethod: Combination method of input flat field exposures for master flat field creation. See Section 7.6 for a complete description of all the combination methods. Note that the master flat field is the frame where the fiber spectra tracing is performed, for the definition of the extraction mask. Possible settings of *StackMethod* are:

Auto: Given the number of input flat fields, an optimal frame combination method is selected. Currently this is always going to the method *"Average"*.

Average: The master flat field is the mean of the input frames.

Ksigma: The master flat field is the mean of the input frames, after K-sigma screening of pixel values. The number of sigma to be applied in the rejection is specified by the parameters *KSigmaLow* and *KSigmaHigh*.

Median: The master flat field is the median of the input frames.

MinMax: The master flat field is the mean of the input frames, after rejection of minimum and maximum values. The number of values to reject is specified by the parameters *MinRejection* and *MaxRejection*.

A description of the algorithms used in this recipe is given in Section 7.24, page 164.

5.25 vmifuscience

The VIMOS pipeline recipe *vmifuscience* is used to extract IFU scientific spectra applying the input extraction mask, after aligning it to the brightest spectra detected on the input exposure. The extracted spectra are then resampled at a constant wavelength step, after aligning the input wavelength calibration to the positions of a set of identified sky lines. The extracted spectra are eventually corrected for the relative differences in transmission from fiber to fiber.

The files to be included in the input SOF are listed in Table 5.25.1.

DO category	**Type**	**Explanation**	**Required**
IFU_SCIENCE	Raw frame	Scientific spectra	√
MASTER_BIAS	Calibration	Master bias	√
IFU_TRACE	Calibration	Extraction mask	√
IFU_IDS	Calibration	Wavelength calibration	√
IFU_TRANSMISSION	Calibration	Transmission correction	√
CCD_TABLE	Calibration	Bad pixel table	

Table 5.25.1: *Input files for the vmifuscience recipe.*

The extraction mask, the wavelength calibration, and the relative transmission table, are those generated by the recipe *vmifucalib* (see Section 5.24, page 101).

A CCD table must be specified only if bad pixel cleaning is requested. In the calibration directories there is one CCD_TABLE file for each quadrant, named `badpixel.`q`.tfits` (where q is the quadrant number). Care should be made in selecting the appropriate bad pixel tables for imaging and spectral instrument modes, that have the same names but are located in separated directories.

All the products of the *vmifuscience* recipe are indicated in Table 5.25.2.

File name	**DO category**	**Type**	**Explanation**
ifu_science_reduced.fits	IFU_SCIENCE_REDUCED	FITS	Reduced scientific spectra
ifu_fov.fits	IFU_FOV	FITS	Reconstructed field-of-view image

Table 5.25.2: *Products of the vmifuscience recipe.*

The extracted spectra are stored in the output images in a conventional order, with blue on the left and red on the right side. The images have 400 rows in the case of MR and HR observations, and 1600 rows in the case of LR observations. The spectra starting from the left side of each pseudo-slit are stored starting from the bottom rows of the output images. In the case of LR observations, the first 400 spectra from the pseudo-slit 1 are at the bottom, and the last 400 spectra from the pseudo-slit 4 are at the top.

Each image row corresponds to an IFU fiber position on the IFU head. This correspondence is described in a set of 8 tables located in the calibration directories (see Table 5.24.3 on page 103, and its description in Section 5.24).

The *vmifuscience* parameters are listed in Table 5.25.3.

Parameter	Possible values	Explanation
BiasMethod	Master Zmaster	Bias removal with no overscan correction Bias removal with overscan correction
CleanBadPixel	true false	Clean bad pixels Do not clean bad pixels
UseSkylines	true false	Use sky lines to align wavelength calibration Do not align wavelength calibration

Table 5.25.3: *vmifuscience parameters.*

A more complete description of the used parameters meaning is given here:

BiasMethod: Method for bias removal from the input frame. The bias removal procedure is described in some detail in Section 7.3. Possible settings are:

Master: After master bias subtraction, prescan and overscan regions are trimmed away from the processed frame.

Zmaster: After master bias subtraction the overscan correction is applied before trimming away the overscan regions.

CleanBadPixel: Bad pixel correction on the scientific exposure. If this option is turned on, a bad pixel table should be specified in the input SOF (see Table 5.21.1, page 91). The bad pixel correction algorithm is described in Section 7.1, page 121.

UseSkylines: If this parameter is set, a number of sky lines are searched and identified in the input science exposure. Currently, just the four bright sky lines at 5577.338, 6300.304, 6363.780, and 8344.602 Ångstrom are used. The median offset from their expected positions along the dispersion direction is taken as a measure of the variation of the instrument flexure between the science exposure and the flat field and arc lamp exposures used for calibration. This offset is added to the constant term of the IDS polynomials (see Section 5.24, page 101), before using them in the spectral extraction task.

A description of the algorithms used in this recipe is given in Section 7.25, page 164.

5.26 vmifustandard

The VIMOS pipeline recipe *vmifustandard* is used to extract IFU scientific spectra applying the input extraction mask, after aligning it to the brightest spectra detected on the input exposure. The extracted spectra are then resampled at a constant wavelength step, after aligning the input wavelength calibration to the positions of a set of identified sky lines. The extracted spectra are eventually corrected for the relative differences in transmission from fiber to fiber, they are sky subtracted, and added together to produce the total standard star spectrum.

The files to be included in the input SOF are listed in table 5.26.1.

DO category	Type	Explanation	Required
IFU_STANDARD	Raw frame	Scientific spectra	√
MASTER_BIAS	Calibration	Master bias	√
IFU_TRACE	Calibration	Extraction mask	√
IFU_IDS	Calibration	Wavelength calibration	√
IFU_TRANSMISSION	Calibration	Transmission correction	√
CCD_TABLE	Calibration	Bad pixel table	

Table 5.26.1: *Input files for the vmifustandard recipe.*

The extraction mask, the wavelength calibration, and the relative transmission table, are those generated by the recipe *vmifucalib* (see Section 5.24, page 101).

A CCD table must be specified only if bad pixel cleaning is requested. In the calibration directories there is one CCD_TABLE file for each quadrant, named `badpixel.`*q*`.tfits` (where *q* is the quadrant number). Care should be made in selecting the appropriate bad pixel tables for imaging and spectral instrument modes, that have the same names but are located in separated directories.

All the products of the *vmifustandard* recipe are indicated in Table 5.26.2.

File name	DO category	Type	Explanation
ifu_standard_reduced.fits	IFU_STANDARD_REDUCED	FITS	Reduced fiber spectra
ifu_fov.fits	IFU_FOV	FITS	Reconstructed field-of-view image
ifu_standard_extracted.fits	IFU_STANDARD_EXTRACTED	FITS	Total standard star spectrum
ifu_science_sky.fits	IFU_SCIENCE_SKY	FITS	Sky spectrum

Table 5.26.2: *Products of the vmifustandard recipe.*

The extracted spectra are stored in the output images in a conventional order, with blue on the left and red on the right side. The images have 400 rows in the case of MR and HR observations, and 1600 rows in the case of LR observations. The spectra starting from the left side of each pseudo-slit are stored starting from the bottom rows of the output images. In the case of LR observations, the first 400 spectra from the pseudo-slit 1 are at the bottom, and the last 400 spectra from the pseudo-slit 4 are at the top.

Each image row corresponds to an IFU fiber position on the IFU head. This correspondence is described in a set of 8 tables located in the calibration directories (see Table 5.24.3 on page 103, and its description in Section 5.24).

The sky spectrum is determined as the median values of all the extracted spectra along the cross dispersion direction. The total spectrum is then computed as the sum of all the sky-subtracted spectra.

The *vmifustandard* parameters are the same as for recipe *vmifuscience*, and they are listed in Table 5.25.3.

A description of the algorithms used in this recipe is given in Section 7.26, page 165.

5.27 vmifucombine

This recipe is used to compose the reconstructed images of the IFU field-of-view from different VIMOS quadrants into a single image. Such images are created by the recipes *vmifuscience* and *vmifustandard*. The input images must belong to different quadrants, so that there cannot be more than 4. In the current implementation the images are merely summed together: in fact, all field-of-view images are always 80x80 pixels, with only the sector corresponding to one quadrant having intensities that differ from zero. No relative flux correction between quadrants is applied.

6 Geometrical distortions models

The reduction of VIMOS scientific data made by the VIMOS pipeline is based on a set of predefined models of the optical distortions affecting the instrument.

In the VIMOS pipeline, optical distortions modeling is obtained by simple polynomial fitting based on known quantities, such as celestial coordinates of astrometric stars, positions of pinholes on a calibration mask, or spectral lines wavelengths from a catalog, all compared to the positions of detected features and patterns on the CCD.

Pipeline recipes related to geometrical calibrations generate a set of IWS configuration files where the coefficients of the derived polynomials are stored. This information will be copied, when appropriate, from the IWS configuration files to the headers of any dataset generated by the VIMOS instrument.

Optical distortions are not expected to remain constant in time. Small changes are introduced by modifying the orientation of the instrument within the gravitational field. A progressive aging of the structure, and possible interventions on the instrument, may also contribute to long term changes.

For this reason, the VIMOS pipeline recipes will use the distortion models contained in the datasets headers as “first guesses”, to be used as a starting point for further improvement of the solutions, adapting them to the momentary instrument distortions.

“First guesses” will work as long as they are close enough to the real distortions, ensuring a safe identification of the appropriate reference signals contained in the data. How much is “close enough” depends on a large number of factors, mainly related to the nature of the distortion, and to the robustness of the pattern matching algorithm of the involved pipeline recipe. This will be discussed in more detail in Section 7.

6.1 Polynomial models

The geometrical distortions introduced by the VIMOS + UT optics can be distinguished into *optical* and *spectral,* mirroring the fundamental instrument setups. Each optical and spectral distortion is in its turn described by a set of polynomial models. In some cases the polynomial models encode not just a distortion (intended as a transformation within the same coordinate system), but a transformation from a coordinate system to another that may include also the geometrical distortions.

Here is an overview of the polynomials used to model each distortion:

Optical

- Mask to CCD transformation (MAS2CCD)
  - Transformation matrix (scale, shift, rotation)
  - Two bivariate polynomial fits of the residuals (for the X and the Y CCD coordinates)
- CCD to Mask transformation (CCD2MAS)
  - Transformation matrix (scale, shift, rotation)
  - Two bivariate polynomial fits of the residuals (for the x and the y Mask coordinates)
- Sky to CCD distortion (SKY2CCD)

  - Bivariate polynomial fit of the residuals of CCD positions derived applying the WCS received from the TCS
- CCD to Sky distortion (CCD2SKY)
  - Inverse of the bivariate polynomial fit modeling the Sky to CCD distortion

During the data reduction process the Sky to CCD distortion model is converted by the pipeline into the CO matrix standard, used in the SAO WCSTools package [10].

Spectral

- Zero Order Contamination (ZERO)
  - Two bivariate polynomials (separately for the X and the Y CCD coordinate) of mask coordinates *vs* CCD positions
- Optical Distortion (OPT)
  - Two bivariate polynomials (separately for the X and the Y CCD coordinate) of mask coordinates *vs* CCD positions
- Spectral Curvature (CRV)
  - *Local CRV*: Simple polynomial fits of local curvatures
  - *Global CRV*: Bivariate polynomial fits of the coefficients of local CRV *vs* CCD positions
- Inverse Dispersion Solution (IDS)
  - *Local IDS*: Simple polynomial fits of wavelengths *vs* CCD positions
  - *Global IDS*: Bivariate polynomial fits of the coefficients of local IDS *vs* CCD positions

The so-called "optical distortion model" is really a transformation from Mask to CCD coordinates valid for the spectral instrument setup, that includes the optical distortions at a conventional reference wavelength. The choice of a reference wavelength λ_o is in principle arbitrary, being just a conventional zero-point for all the spectral distortion models and transformations. In practice λ_o is chosen roughly in the middle of the valid spectral range of a given grism, possibly matching the wavelength of a bright and isolated line of the arc lamp catalog used for spectral calibrations.

The coefficients of all the polynomial fits are written to the appropriate VIMOS IWS configuration files, with the exception of the local CRV and IDS, whose coefficients are written instead into another pipeline product, the *extraction table* (see Section 5.21, page 92) in the case of MOS observations, and into the trace and IDS tables (see Section 5.24, pages 103 and 103) in the case of IFU observations.

Details on the algorithms applied by the relevant pipeline recipes can be found in Section 7. In the present section just a description of the geometrical distortion models is given.

6.2 Optical distortions

We include in this section any transformation between different coordinate systems while the instrument is configured in direct imaging mode.

Three fundamental coordinate systems can be considered:

- Celestial (Sky)
- Telescope focal plane (Mask)
- Instrument focal plane (CCD).

Only the transformations from CCD to Sky and from CCD to Mask (together with their inversions) are used and supported by the VIMOS pipeline.

6.2.1 CCD to Mask transformation and its inverse

The transformation from CCD to Mask coordinates is described by a two-layer model, consisting of a linear transformation containing rotation, shift, and scaling, to which a bivariate polynomial fit of the residuals is added.

The linear transformation can be expressed in the form

$$\begin{cases} x = a_{xx}X + a_{xy}Y + x_o \\ y = a_{yx}X + a_{yy}Y + y_o \end{cases}$$

where (X, Y) are CCD coordinates (pixels), and (x, y) the corresponding mask coordinates (millimetres).

If the mask were perfectly aligned with the CCD, only the diagonal elements of the matrix, a_{xx} and a_{yy}, would differ from zero, and they would correspond to the scale factor between mask and CCD (about 0.119 mm/pixel).

The coefficients of the linear transformation for quadrant q are written to the entries of the `IMG_mask2ccd_`q`.cmf` IWS configuration file indicated in Table 6.2.1.

CCD to Mask linear transformation	
IMG_mask2ccd_q.cmf	coefficient
PRO CCD MASK X0	x_o
PRO CCD MASK XX	a_{xx}
PRO CCD MASK XY	a_{xy}
PRO CCD MASK Y0	y_o
PRO CCD MASK YY	a_{yy}
PRO CCD MASK YX	a_{yx}

Table 6.2.1: *CCD to Mask linear transformation coefficients.*

The residuals to this linear transformation are modeled by a bivariate polynomial, that accounts for the optical distortions of the instrument:

$$\begin{cases} \Delta x = \sum_{i,j} x_{ij} X^i Y^j & \text{with } 0 \leq i \leq m, 0 \leq j \leq m \\ \Delta y = \sum_{i,j} y_{ij} X^i Y^j & \text{with } 0 \leq i \leq m, 0 \leq j \leq m \end{cases}$$

The coefficients of the distortion for quadrant q, and the max degree of each variable of the bivariate polynomial, are written to the entries of the `IMG_mask2ccd_`q`.cmf` IWS configuration file indicated in Table 6.2.2.

CCD to Mask distortion model	
`IMG_mask2ccd_`q`.cmf`	coefficient
`PRO CCD MASK XORD`	m
`PRO CCD MASK YORD`	m
`PRO CCD MASK X_`i`_`j	x_{ij}
`PRO CCD MASK Y_`i`_`j	y_{ij}

Table 6.2.2: *CCD to Mask distortion model coefficients.*

Currently m must be kept equal to 3, for compatibility with the VMMPS. The complete transformation from CCD to Mask is given by the sum of the linear transformation with the distortion model.

The RMS (in millimetres) of the residuals of the complete transformation is also written to the IWS configuration file, at the entries `PRO CCD MASK XRMS` and `PRO CCD MASK YRMS`, together with the assigned temperature and time tag, written to `PRO CCD MASK TEMP` and `PRO CCD MASK DAYTIM`.

The inverse transformation, from Mask to CCD, is completely analogous to the CCD to Mask transformation.

6.2.2 CCD to Sky distortion and its inverse

For transforming CCD pixel coordinates to Sky coordinates and back, a WCS is written by the TCS to the FITS header of the observation data. This transformation is performed by the pipeline calling the appropriate functions of the SAO WCSTools package [10].

Once a WCS is established, the contribution of the optical distortions needs to be modeled. This is a distortion, meaning that the transformation is performed within the same coordinate system (in this case, the CCD). It is modeled by a two-branches bivariate polynomial analogous to the one used for the Mask to CCD transformations:

$$\begin{cases} X_v = \sum_{i,j} \alpha_{ij} X^i Y^j & \text{with } 0 \leq i \leq m, 0 \leq j \leq m \\ Y_v = \sum_{i,j} \beta_{ij} X^i Y^j & \text{with } 0 \leq i \leq m, 0 \leq j \leq m \end{cases}$$

We describe here for simplicity just the CCD to Sky model. This model is not converting image pixels into celestial coordinates (RA and Dec), but converts pixel positions (X, Y) on the CCD into virtual pixel positions (X_v, Y_v), that are corrected for distortions and temperature effects. These virtual pixel positions can then be converted into celestial coordinates using the WCS information present in the data header (see Figure 6.2.1).

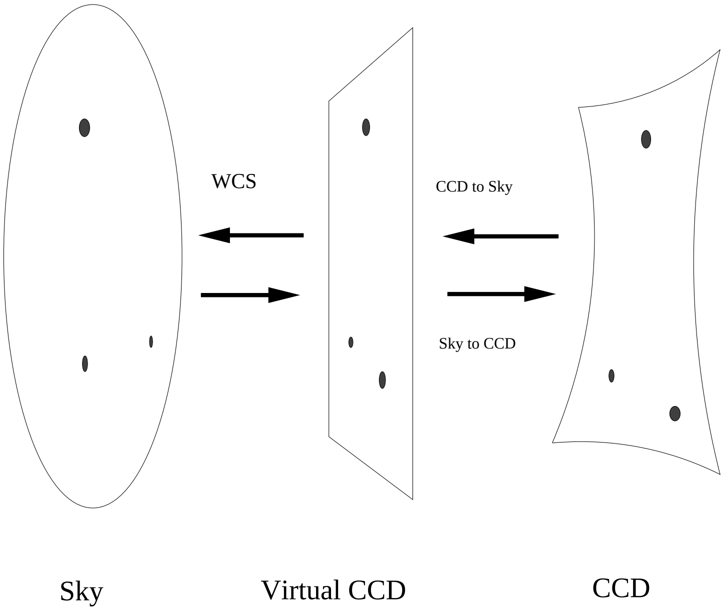


Figure 6.2.1: *Transformations and distortions between sky and CCD.*

The coefficients of the distortion for quadrant q, and the max degree of each variable of the bivariate polynomial, are written to the entries of the `IMG_sky2ccd_`q`.cmf` IWS configuration file indicated in Table 6.2.3.

CCD to Sky distortion model	
`IMG_sky2ccd_`q`.cmf`	coefficient
`PRO CCD SKY XORD`	m
`PRO CCD SKY YORD`	m
`PRO CCD SKY X_`i`_`j	α_{ij}
`PRO CCD SKY Y_`i`_`j	β_{ij}

Table 6.2.3: *CCD to Sky distortion model coefficients.*

For m a value of 3 is currently chosen.

The RMS of the residuals of the models are also written to the IWS configuration file, at the entries `PRO CCD SKY XRMS` and `PRO CCD SKY YRMS`, together with the assigned temperature and time tag, written to `PRO CCD SKY TEMP` and `PRO CCD SKY DAYTIM`.

The inverse model would simply produce the (X, Y) coordinates of the real CCD from the (X_v, Y_v) virtual coordinates obtained by applying the WCS to (RA, Dec) positions.

The pipeline converts these distortion models into the CO-matrix convention that is then written to the FITS headers of the reduced science images.

6.3 Spectral distortions

We include in this section any transformation between different coordinate systems while the instrument is configured in spectral (MOS or IFU) mode.

Four fundamental coordinate systems can be considered:

- Celestial (Sky)
- Telescope focal plane (Mask)
- Instrument focal plane (CCD).
- Spectral wavelength (Ångstrom).

Only the transformations from Mask to CCD and from CCD to wavelength are currently used and supported by the VIMOS pipeline.

6.3.1 Zero order contamination model

Currently not implemented

6.3.2 Optical distortion model

The optical distortion model of the grism is really a direct transformation from (x, y) mask coordinates to (X, Y) CCD coordinates, valid for a conventional reference wavelength λ_o.

The model can be expressed in the form

$$\begin{cases} X = \sum_{i,j} a_{ij} x^i y^j & \text{with } 0 \leq i \leq m, 0 \leq j \leq m \\ Y = \sum_{i,j} b_{ij} x^i y^j & \text{with } 0 \leq i \leq m, 0 \leq j \leq m \end{cases}$$

where (X, Y) are CCD coordinates (pixels), and (x, y) the corresponding mask coordinates (millimetres).

The coefficients of the distortion for quadrant q, and the max degree of each variable of the bivariate polynomial, are written to the entries of the `MOS_wavecal_`*grism_name_q*`.cmf` IWS configuration file indicated in Table 6.3.1.

For m a value of 3 is currently chosen.

The RMS of the residuals of the model is also written to the IWS configuration file, at the entries `PRO OPT DIS XRMS` and `PRO OPT DIS YRMS`, together with the assigned temperature and time tag, written to `PRO OPT DIS TEMP` and `PRO OPT DIS DAYTIM`.

This model provides the reference on which both the spectral curvature and the inverse dispersion models are based.

In the specific case of IFU data reduction, a global optical distortion model is not really computed, and a zeropoint for all other distortion models is defined independently for each fiber.

Optical distortion model	
MOS_wavecal_$grism_name_q$.cmf	coefficient
PRO OPT DIS XORD	m
PRO OPT DIS YORD	m
PRO OPT DIS X_i_j	a_{ij}
PRO OPT DIS Y_i_j	b_{ij}

Table 6.3.1: *Optical distortion model coefficients.*

6.3.3 Spectral curvature model

The spectrum corresponding to the position (x, y) on the mask traces a curve on the CCD. The spectral curvature model is used to determine this curve as a function of the mask coordinates.

The modeled quantity is the deviation ΔX as a function of the distance ΔY from the (X, Y) CCD coordinates obtained applying the optical distortion model to the given (x, y) mask coordinate (see Section 6.3.2 and Figure 6.3.1).

This is the *local* curvature model, that can be expressed in the form

$$\Delta X = \sum_i c_i \Delta Y^i$$

with $0 \leq i \leq m$ (with m currently set to 2).

The coefficients of the local curvature models, defined for each detected spectral edge on a flat field exposure, are written to the *extraction table* (see Section 5.21). It should be noted that the coefficient c_o is always equal to zero (for any (x, y)), as it is implied by the curvature model definition.

The coefficients c_i depend on the (x, y) mask coordinates, and can be modeled by the m bivariate polynomials:

$$c_i = \sum_{j,k} \Gamma_{i,jk} x^j y^k$$

with $0 \leq j \leq n$ and $0 \leq k \leq n$ (with n currently set to 2).

The set of polynomials modeling the coefficients of the local curvature models is known as the *global* curvature model. All the coefficients for quadrant q and grism *grism_name*, with the max degree for each variable of all the simple and the bivariate polynomials, are written to the entries of the MOS_wavecal_*grism_name*_q.cmf IWS configuration file indicated in Table 6.3.2.

Since all the c_i are zero for $i = 0$, it immediately follows that all the $\Gamma_{o,jk}$ (*i.e.*, all the PRO CRV MOD_0_j_k) are also zero. They are written nevertheless to the data headers for consistency in the description of the polynomial models produced by the VIMOS pipeline recipes.

The temperature and the time tag assigned to the curvature model are identical to the ones of the inverse dispersion solution (see next section). The CRV and the IDS models are always derived from flat field and arc lamp exposures obtained (almost) simultaneously, to ensure that they are compatible with each other.

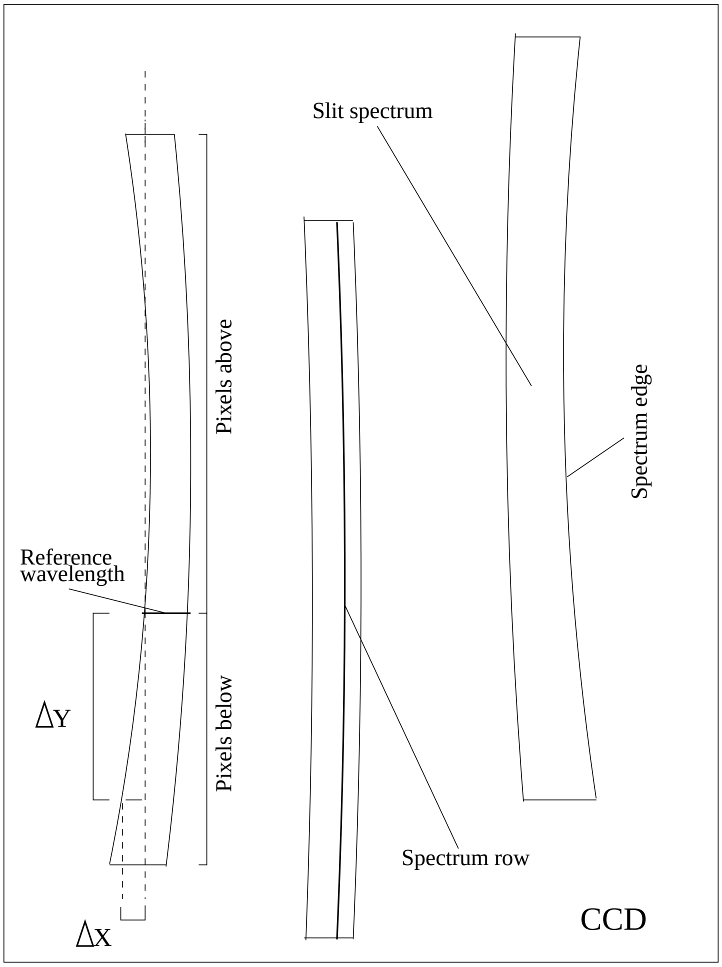


Figure 6.3.1: *MOS slit spectra on a CCD.*

In the specific case of IFU data reduction, a global curvature model is not really computed. The local curvatures are modeled independently for each fiber by a direct polynomial fit of absolute X *vs* Y CCD coordinates obtained from the fiber tracing task.

6.3.4 Inverse dispersion solution

As seen in Section 6.3.2, the optical distortion model is used to determine the position (X, Y) on the CCD corresponding to a position (x, y) on the mask, valid for a conventional reference wavelength λ_o.

In VIMOS the light is dispersed by the grism along the Y CCD coordinate, and therefore the wavelength calibration consists of a relation between the wavelength and the ΔY distance from the Y position obtained applying the optical distortion model to (x, y).

The modeled quantity is the deviation ΔY as a function of the wavelength difference $\Delta\lambda = \lambda - \lambda_o$, expressed as usual with a polynomial fit that represents the *local* inverse dispersion solution (IDS):

Global curvature model	
MOS_wavecal_$grism_name_q$.cmf	coefficient
PRO CRV POL ORD	m
PRO CRV MOD XORD	n
PRO CRV MOD YORD	n
PRO CRV MOD_i_j_k	$\Gamma_{i,jk}$

Table 6.3.2: *Global curvature model coefficients.*

$$\Delta Y = \sum_i d_i \Delta\lambda^i$$

with $0 \leq i \leq m$ (with m currently set to 3 for LR grisms, and to 4 for MR and HR grisms, being the lowest possible polynomial degree at which the residuals of the fit display a random distribution). The coefficients of the local IDS models, defined for each point corresponding to a different X CCD pixel for each slit of the mask, are written to the *extraction table* (see Section 5.21, page 92).

The coefficients d_i depend on the (x, y) mask coordinates, and can be modeled by the $m + 1$ bivariate polynomials:

$$d_i = \sum_{j,k} \Lambda_{i,jk} x^j y^k$$

with $0 \leq j \leq n$ and $0 \leq k \leq n$ (with n currently set to 3).

The set of polynomials modeling the coefficients of the local IDS models is known as the *global* IDS. All the coefficients for quadrant q and grism *grism_name*, with the max degree for each variable of all the simple and the bivariate polynomials, are written to the entries of the MOS_wavecal_*grism_name_q*.cmf IWS configuration file indicated in Table 6.3.3.

Global inverse dispersion solution	
MOS_wavecal_$grism_name_q$.cmf	coefficient
PRO IDS REL ORD	m
PRO IDS MAT XORD	n
PRO IDS MAT YORD	n
PRO IDS MAT_i_j_k	$\Lambda_{i,jk}$

Table 6.3.3: *Global inverse dispersion solution coefficients.*

The RMS of the residuals of the model is also written to the IWS configuration file, at the entry PRO IDS MAT YRMS (the entry PRO IDS MAT XRMS is unused, for obvious reasons). The temperature and time tag assigned to the model are written to PRO IDS MAT TEMP and PRO IDS MAT DAYTIM.

In the specific case of IFU data reduction, a global inverse dispersion solution is not really computed. Just the local wavelength calibration described above is computed separately for each fiber.

6.3.5 Slit spectra extraction

Probably the best way to summarise the complete modeling of spectral distortions is to see it applied to the problem of extracting a slit spectrum from a raw image.

This is equivalent to finding what CCD coordinates correspond to a given position on the mask and to a given wavelength.

Let's indicate the spectral distortion models defined in the previous sections using the symbols OPT, CRV, and IDS. Then the (X, Y) CCD coordinate corresponding to (x, y, λ) are given by:

$$\begin{cases} X = OPT_x(x, y) + CRV(IDS(\lambda)) \\ Y = OPT_y(x, y) + IDS(\lambda) \end{cases}$$

This results in a wavelength calibrated slit spectrum, corrected for the spectral and the spatial curvatures.

7 Algorithms

In this section the data reduction procedures applied by the 20 pipeline recipes currently in use (see Section 5.1) are described in some detail. Common algorithms, as cosmic rays removal or bad pixel cleaning, are described separately.

7.1 Bad pixel cleaning

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

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

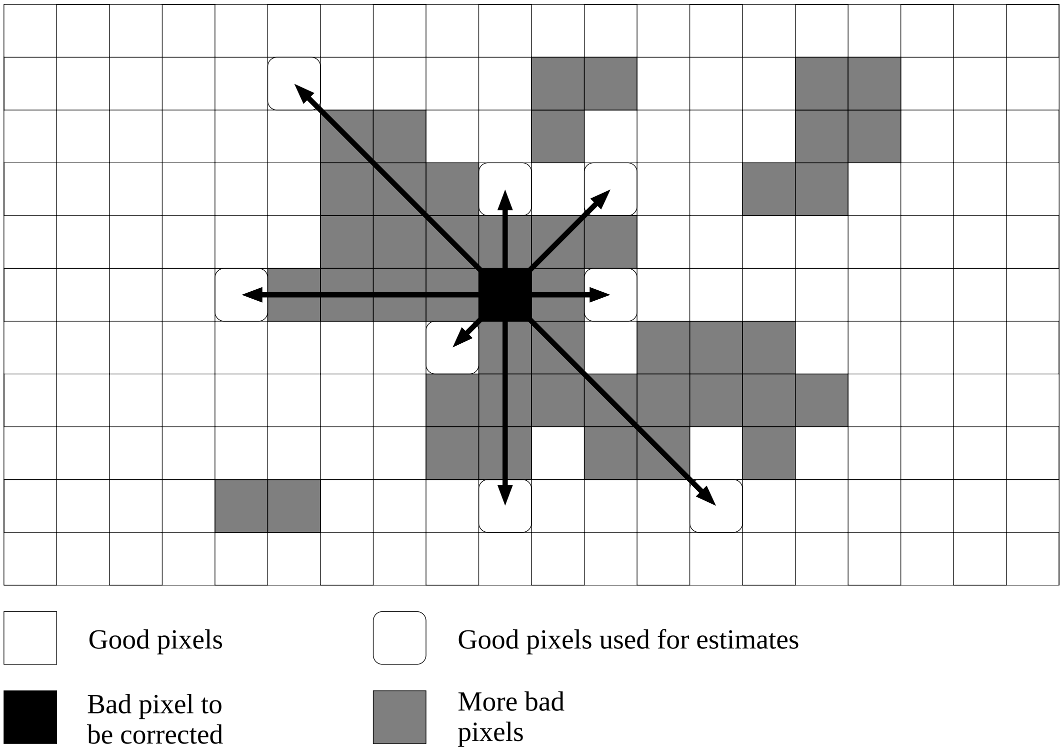


Figure 7.1.1: *Good pixels to be used in the estimate of a given bad pixel are searched along the indicated directions.*

Any bad pixel is given a new value, computed as follow: the closest good pixels along the vertical, the horizontal, and the two diagonal directions are found (see Figure 7.1.1). This search is done within a distance of 100 pixels.

If no good pixel is found within this range, then the bad pixel is not corrected. All the good pixels found within range will be used to compute the bad pixel value.

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

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

7.2 Cosmic rays removal

The core of a cosmic rays removal procedure is to determine what is and what is *not* a cosmic ray. The algorithm used for this purpose by the VIMOS pipeline recipes is the same applied by the MIDAS command `FILTER/COSMIC`, with some extensions.

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

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

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

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

with k the number of noise sigmas used in thresholding.

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

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

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

Once all the pixels affected by cosmic ray events has been located and listed in a cosmic ray events table, their values are interpolated using the procedure described in Section 7.1. If a bad pixel table is also given to a recipe, then the bad pixels are avoided in the interpolation procedure.

7.3 Bias subtraction

Removing the bias from any raw frame is a relatively simple process, but not simple enough to avoid a description on its own.

A master bias frame (`MASTER_BIAS`) is used to remove the bias level (and, if present, the fixed-pattern-noise related to the bias) from a raw frame. Typically a master calibration is produced with its overscan regions trimmed, and if this is the case with the master bias used then its missing overscan regions are extrapolated by repeating the signal contained in its border regions with equal size.

The master bias is subtracted from the raw data frame, whose overscan regions are then trimmed away. Optionally (when the *BiasMethod* recipe parameter is set to *"Zmaster"*) the residual signal in the overscan regions is averaged along the X CCD coordinate, and the obtained mean Y values are modeled with a second order polynomial fitting. This model is then subtracted from the rest of the image.

7.4 Dark subtraction

Subtracting the dark current component from any raw frame consists of multiplying an input master dark frame by the exposure time (in seconds) of the frame to be corrected, and then subtract such rescaled dark frame from it. The dark level is quite low for VIMOS CCDs (about 5 $e^{-} \cdot \mathrm{h}^{-1} \cdot \mathrm{pixel}^{-1}$), so this operation would be in most cases superfluous.

7.5 Flat field correction

The flat field correction merely consists of dividing the frame to be corrected by a given master sky flat field frame produced by the recipe *vmimflatsky* for direct imaging observation (see Section 7.11, page 127), or produced by the recipe *vmspflat* for MOS observations (see Section 7.19, page 133).

7.6 Frame combination

A common task to many of the VIMOS pipeline recipes is the combination of several frames of the same kind. Currently four basic frame combination methods are available:

Average of frames:

> Each combined frame pixel is the average of all the corresponding pixel values in the input frames. In this case at least two input frames are required.

Median of frames:

> Each combined frame pixel is the median of all the corresponding pixel values in the input frames. In this case at least three input frames are required. In case of an even number of input frames, the median value is taken as the mean of the two central values.

Rejection of minimum-maximum values:

Each combined frame pixel is the average of all the corresponding pixel values in the input frames, after rejecting a given number of minimum and maximum values from the set. In this case at least three input frames are required.

K-sigma rejection:

The median value of each pixel of the input frames is computed, and the standard deviation of all the pixel values from the median is evaluated. Each combined frame pixel is the average of all the corresponding pixel values in the input frames, after rejecting any pixel value deviating more than a given number of standard deviations. In this case, at least four input frames are required.

Combination methods different from averaging should, in principle, only be applied to the special case of statistically homogeneous set of frames. However, as a preventive measure taken in the VIMOS pipeline recipes, before applying those combination methods the input images are rescaled to a common level, chosen as the median level of the first input frame to which all the other frames median levels are aligned. This is still not a statistically valid way to proceed, since the noise level of each image is different, and methods like the K-sigma rejection would lose any real meaning.

7.7 vmdet

This recipe carries out the following fundamental steps:

1. Determining the read-out-noise.
2. Bias subtraction from all input flat fields.
3. Creating photon transfer curve, determining the gain factor.
4. Bad pixels identification.

A description of each step is given in the following sections.

7.7.1 Read-out-noise determination

Before subtracting the bias from the input flat field frames, the *read-out-noise* (RON) is evaluated from the flat fields overscan regions. Each overscan region is subtracted from itself shifted by 1x1 pixels, and the variance V of the difference image is determined. The RON is estimated as:

$$r = \sqrt{\frac{V}{2}}$$

The mean value of the RON values obtained from each overscan region is the estimated RON of the CCD (in ADU). At a later step, after the determination of the gain factor, the RON will be converted into electrons and written to the output bad pixel table header keyword `ESO DET OUT1 RON`.

7.7.2 Bias subtraction

The master bias is subtracted from each of the flat field using the *"Zmaster"* method, described in detail in Section 7.3.

7.7.3 Photon transfer curve and gain factor determination

The gain determination is based on the paper by L. Mortara and A. Fowler [12]. The photon transfer curve is the relation between the observed signal level and the observed variance of this signal. In principle, both RON and gain can be obtained by least squaring the relation

$$V = \left(\frac{r}{g}\right)^2 + \frac{S}{g}$$

where V is the variance of the bias subtracted signal S (in ADU), g is the unknown gain factor in e^-/ADU (corresponding to the data header keyword `ESO DET OUT1 CONAD`), and r is the read-out-noise in e^-.

Since the relation between the signal and its variance is linear, we can build the photon transfer curve from the average signal and variance determined on just a portion of the chip.

From the central 200 x 200 pixel region of the CCD four different photon transfer curves are derived, one from each quarter of this region. This is a way to obtain independent determinations of the gain factor, and allow an estimate of the statistical error on the final result.

For each pair of flat fields with equal exposure time the median value of the signal within the selected regions is computed, while the variance is evaluated from the difference of the pair of frames.

The final gain factor is determined as the mean

$$g = \frac{\sum_i g_i}{4}$$

where g_i are the gain values obtained from the linear fitting of the four independent photon transfer curves. The error on the gain is estimated as

$$\Delta g = \frac{1}{2}\sqrt{\frac{\sum_i (g - g_i)^2}{3}}$$

(the factor $1/2$ is converting the population standard deviation into error on the mean — dividing by the square root of the number of values contributing to the mean itself). The value of the gain is written to the bad pixel table header keyword `ESO DET OUT1 CONAD`, and its inverse to its header keyword `ESO DET OUT1 GAIN`.

An estimate of the RON could also be obtained from this linear fit, but while with this method the gain determination is accurate, the RON determination turns out to be very poor. For this reason in the *vmdet* recipe the RON is evaluated by directly measuring the variance of the signal within the overscan regions, as shown in Section 7.7.1.

7.7.4 Bad pixels identification

The representative exposure level of each pair of flat fields with the same exposure time is determined as the median level of the 200 x 200 central region of the images. Then each pixel value from the same images is compared to the corresponding exposure levels. A pixel is flagged as "bad" when the slope of the linear fit of this table of values deviates from the expected slope by more than a given threshold.

The specified threshold is expressed in standard deviations from the mean value of the slopes. In order to apply this threshold correctly, and to determine what the expected slope is, the effects of the non-uniform illumination of the CCD are kept into account: an empirical (polynomial) model of the relation between the local illumination level with both the expected slope and the expected variance of the slope is determined by the recipe before applying the specified threshold.

The detected bad pixels are written to the bad pixel table columns and, if requested, as pixels of value 1 in a 0-filled image having the same size of the CCD (overscans are removed). For debug purposes, an error image containing the uncertainties on the fitted slopes can also be created.

It should be clear that with the described method any pixel that is not exposed (*e.g.*, because it belongs to a vignetted part of the CCD) would also be classified as "bad", even if it is capable of a regular response.

7.8 vmbias

This recipe carries out the following fundamental steps:

1. Optional cosmic rays removal (see Section 7.2).
2. Combination of input bias frames (see Section 7.6).
3. Optional bad pixel cleaning from output master bias frame (see Section 7.1).

The details of each step are explained in the specified sections.

7.9 vmdark

This recipe carries out the following fundamental steps:

1. Bias subtraction (see Section 7.3).
2. Optional cosmic rays removal (see Section 7.2).
3. Combination of input dark frames (see Section 7.6).
4. Optional bad pixel cleaning from output master dark frame (see Section 7.1).

The details of each step are explained in the specified sections. The product master dark is divided by the total exposure time of all input darks (in seconds).

7.10 vmimflatscreen

This recipe carries out the following fundamental steps:

1. Bias subtraction (see Section 7.3).
2. Optional dark subtraction (see Section 7.4).
3. Optional cosmic rays removal (see Section 7.2).
4. Combination of input screen flat field frames (see Section 7.6).
5. Creation of master flat field as normalisation of the combined flat field.
6. Optional bad pixel cleaning from output flat field frames (see Section 7.1).

Beyond the standard reduction steps, described in some detail in the indicated sections, only the flat field normalisation procedure needs to be outlined here.

7.10.1 Screen flat field normalisation

The screen flat field is just used to determine the small-scale fixed-pattern-noise. Any possible large scale trend should be modeled and removed from the result frame, because it not only just reflects the characteristics of the instrument detector and optics, but also the typically non-uniform illumination of the screen by the calibration lamp. A good determination of the large scale trends would be obtained from a *sky* exposure, where a uniform field is actually observed — see Section 7.11.

A model of the large scale trends is obtained by first cleaning the bad pixels from the combined screen flat field (see Section 7.1, page 121) and then heavy smoothing the result. Next, the original combined flat field is divided by this smoothed version of itself, generating the master screen flat field. In symbols,

$$F = \frac{C}{smooth(C)}$$

where C is the combined screen flat field, F the master screen flat field, and $smooth$ the smooth operator.

7.11 vmimflatsky

This recipe carries out the following fundamental steps:

1. Bias subtraction (see Section 7.3).
2. Optional dark subtraction (see Section 7.4).
3. Optional cosmic rays removal (see Section 7.2).
4. Combination of input sky flat field frames (see Section 7.6).

5. Optional modeling of the large scale trends in the combined sky flat field, to be applied to the input master screen flat field.

6. Creation of the master sky flat field as simple normalisation of the combined flat field to its median value.

7. Optional bad pixel cleaning from output master sky flat field (see Section 7.1).

Beyond the standard reduction steps, described in some detail in the indicated sections, only steps 5 and 6 need to be outlined here.

7.11.1 Modeling large scale trends

This optional step is carried out when a master screen flat field was specified in the input SOF.

In general, the combination of the input frames is sufficient to produce an acceptable master sky flat field for use in the flat field correction of imaging science frames. However, in order to eliminate the contributions of field stars on the jittered sequence of sky flat fields, the frame combination method must be based on a rejection algorithm (rather than on a simple average). As a consequence, the combined master sky flat field tends to be noisier than a combined master flat field.

To solve this problem it is possible to use the small-scale information contained in a master screen flat field (see Section 7.10, page 127) and apply it to the more reliable large-scale trend of the combined sky flat field.

First, the combined sky flat field is divided by the master screen flat field, in order to eliminate the small-scale fixed-pattern-noise; next the bad pixels are cleaned (see Section 7.1, page 121), and a heavy smoothing is applied. The model of the large-scale trend obtained in this way is finally multiplied by the master screen flat field, resulting in a better quality combined sky flat field. In symbols,

$$S' = F \cdot smooth\left(\frac{S}{F}\right)$$

where S is the combined sky flat field, F the master screen flat field, $smooth$ the smooth operator, and S' the improved combined sky flat field.

7.11.2 Combined flat field normalisation

The combined sky flat field, whatever way produced, is divided by its median level. The output is defined as the master sky flat field to be used in the reduction of science frames.

7.12 vmmasktoccd

This recipe carries out the following fundamental steps:

1. Source detection on input pinhole image.

2. Matching mask pinholes with sources detected on the pinhole image.

3. Determination of the Mask to CCD transformation and its inverse.

A description of each step is given in the following sections.

7.12.1 Source detection

After applying the standard reduction steps (bias subtraction, flat fielding, cosmic rays and bad pixel cleaning if requested, etc.), the detection of the pinholes images on the CCD is made applying SExtractor [8] in PSF fitting mode. SExtractor is run twice: the first run is meant to determine the instrumental PSF (or, more precisely, the convolution of the pinholes shapes with the instrumental PSF) as a function of the CCD coordinates. A second SExtractor run is performed taking into account this PSF model.

7.12.2 Matching mask pinholes with their CCD images

The pinhole coordinates are read from the ADM included in the input image header, and transformed into CCD expected positions using the “first guess” Mask to CCD transformation. The actual position of a pinhole image on the CCD is searched within a given radius, and if more than one match is found then the brightest is chosen.

7.12.3 Determination of Mask to CCD transformations

A bivariate polynomial fit (see Section 6.2.1, page 113) is performed on pinholes mask positions *vs* detected positions on the CCD. First just a linear fit is tried, determining scale, offset, and rotation. Next a bivariate polynomial fit of the residuals is iterated a given number of times, rejecting at each iteration any detected position deviating more than $4\cdot$ RMS from the model, until the Mask to CCD transformation is determined. The inverse transformation, from CCD to Mask, is obtained by inverse fit of the selected points.

7.13 vmskyccd

This recipe carries out the following fundamental steps:

1. Object detection on input astrometric image.

2. Matching stars with astrometric catalog entries.

3. Determination of Sky to CCD distortion.

A description of each step is given in the following sections.

7.13.1 Object detection

After applying the standard reduction steps (bias subtraction, flat fielding, cosmic rays and bad pixel cleaning if requested, etc.) the input astrometric field exposure is processed by SExtractor [8]. A table with the detected objects and their parameters is produced.

7.13.2 Star matching

All the stars are selected from the detected objects. Next, their CCD coordinates are transformed into celestial coordinates using the WCS and the "first guess" CCD to sky distortion model found in the header of the astrometric image (see Figure 6.2.1, page 115).

The applied CCD to sky distortion, used to transform real CCD pixels into virtual ones (see Section 6.2.2, page 114), is also corrected for temperature effects. Indicating with $C_X(X,Y)$ the X component of the CCD to sky distortion model, the coordinate X_v of the virtual pixel is computed as

$$X_v = S(T) \cdot (C_X(X,Y) - X_c) + X_c$$

where X and Y are the real CCD coordinates, X_c is the central X coordinate of the CCD, and $S(T)$ is the temperature correction factor given by

$$S(T) = k(T - T_m) + 1$$

where $k = 6.0 \cdot 10^{-4}/^oC$ is the thermic expansion coefficient, T the beam temperature, and T_m the temperature at which the used "first guess" CCD to sky distortion model was evaluated. In the same way the

$$Y_v = S(T) \cdot (C_Y(X,Y) - Y_c) + Y_c$$

is applied.

After the objects celestial coordinates are computed, a match is made with the entries of the specified astrometric catalog, leading to a list of detected astrometric stars.

7.13.3 Determination of Sky to CCD distortions

Applying the WCS in the astrometric image header, the catalog celestial coordinates of the detected astrometric stars are transformed into CCD coordinates, and matched to their actual (*i.e.*, derived by SExtractor) coordinates on the CCD. A bivariate polynomial fit (see Section 6.2.2, page 114) is performed on theoretical positions *vs* real positions on the CCD, and vice versa, leading respectively to the Sky to CCD and the CCD to Sky distortion modeling.

The quality of the distortion modeling can be judged by the RMS of the model fit residuals, but most critical is the distribution of the astrometric stars on the field-of-view: it may happen that too few astrometric and badly distributed stars are available for a reliable fit. Such cases may be screened by a visual examination of the astrometric image, but the computation of a uniformity index of the astrometric stars distribution on the field-of-view is helpful. (*To be done*).

7.14 vmimstandard

After applying the standard reduction steps (bias subtraction, flat fielding, cosmic rays and bad pixel cleaning if requested, etc.), this recipe carries out the following fundamental steps:

1. Object detection on input standard stars field image.
2. Matching stars with photometric catalog entries.

The object detection and the matching of the detected stars with the entries of a photometric catalog are carried out as in the recipe *vmskyccd* (see Section 7.13, page 129). This recipe produces a table of the identified standard stars, with their positions, their catalog magnitudes in all the available bands, and their instrumental magnitudes (determined by SExtractor [8]). Further processing of an appropriate set of this kind of table (see for instance the recipe *vmimcalphot*, Section 5.16, page 77) would make possible to determine night zeropoints, atmospheric extinction coefficients, and colour terms.

7.15 vmimcalphot

This recipe is used to determine night zeropoints, atmospheric extinction coefficients, and colour terms from a set of star match tables, produced by the recipe *vmimstandard* (see Section 5.15, page 72). The input star match tables must all be derived from exposures made with the same filter.

In general this problem is solved by a robust minimisation of the linear system

$$Z - E \cdot A - C \cdot C_i = \Delta M_i$$

where, for a given star i, ΔM_i is the measured difference between the catalog magnitude and the instrumental magnitude in the appropriate band, C_i is the star colour index, and A is the airmass. The unknown terms are the atmospheric extinction coefficient E at the considered band, the colour term C, and the zeropoint Z.

However, it is also possible to derive a solution by freezing the values of either or both of the unknowns C and E; while Z is always determined, freezing C would permit the determination of just the best coefficient E and vice versa. Freezing both C and E would determine Z as a simple estimate of ΔM_i (plus an offset). Such cases, and in particular the last one, would typically be preferred in the case that just a small number of stars were available.

7.16 vmimpreimaging

This recipe is used to convert the instrument distortions (see Section 6.2.2, page 114) into the CO-matrix convention followed by the VIMOS mask preparation software (VMMPS). The distortions encoded in the computed CO-matrix include the temperature effects (see Section 7.13.2, page 130).

The conversion from the internal VIMOS convention to the CO-matrix convention is made using a regular 10x10 grid of CCD coordinates that are first transformed into celestial coordinates applying the distortion and transformation models found in the image header. The direct transformation from celestial to distorted CCD coordinates is then fitted by the CO-matrix model, using the appropriate functions of the SAO WCSTools package [10].

This recipe also applies to the input image the same data reduction steps applied by the recipes *vmimstandard* and *vmimobsstare*, excluding the source detection task [8]. The magnitude zeropoint from a given photometric table might also be added to the reduced image header.

7.17 vmimobsstare

This recipe is used to reduce a single scientific exposure in direct imaging mode, and it carries out the following fundamental steps:

1. Bias subtraction (see Section 7.3).
2. Optional dark subtraction (see Section 7.4).
3. Flat field correction (see Section 7.5).
4. Optional bad pixel cleaning (see Section 7.1).
5. Optional cosmic rays removal (see Section 7.2).
6. Source detection (running SExtractor).

The details of each step are explained in the specified sections. About the source detection task, please refer to the SExtractor documentation [8], that can be found at

`http://terapix.iap.fr/rubrique.php?id_rubrique=91/index.html`.

7.18 vmimobsjitter

This recipe is used to reduce a sequence of scientific exposures obtained in direct imaging mode, and to align and sum the results in a combined frame. The data reduction steps applied to each frame are the same applied in recipe *vmimobsstare*. The only difference lays in the final frames combination, that consists of the following fundamental steps:

1. Determine a common pixelisation and coordinate system for the combined frame.
2. Resample the single reduced frames to the new pixelisation.
3. Combination of the resampled frames (see Section 7.6).

A description of the first two steps is given in the following sections.

7.18.1 Determination of common coordinate system and pixelisation

The common coordinate system and pixelisation for the combined frame are simply those of the first frame of the input sequence. The range of celestial coordinates covered by the union of all the input images is determined, and the pixel range of the first frame is expanded accordingly.

7.18.2 Resampling of reduced frames

For each single reduced image, an output image with the common coordinate system and pixelisation defined above is allocated. The coordinates of each pixel of the output frame are converted to celestial coordinates using the common WCS, and then converted to the pixel coordinates of the input image, using its specific WCS[10]. The interpolated value of the nearby pixels — 4 pixels for bilinear and 16 for bicubic interpolation — is computed, and written to the running pixel position of the output image. Coefficients and formulae for the bi-cubic interpolation are taken from "Numerical Recipes", II Ed., page 119.

7.19 vmspflat

This recipe carries out the following fundamental steps:

1. Creation of the combined flat field from the input raw flat field exposures.
2. Creation of the normalised master flat field from the combined flat field.

A description of each step is given in the following sections.

7.19.1 Creation of the combined flat field

After applying the standard reduction steps (bias subtraction, bad pixel cleaning if requested, etc.), the input MOS flat field images are stacked together according to the mask shutter position used for each exposure. In this way one image for each used mask shutter position is produced.

The combined flat field image is constructed by extracting from the stacked images a rectangle centred on each slit spectrum. This rectangle is positioned using the first guess of the optical distortion model found in the input flat fields headers: applying the optical distortion model to a slit central position (x, y) on the mask, the CCD position (X, Y) of the reference wavelength of the spectrum is found (see Section 6.3.2). The extraction rectangle is given an X size equal to the slit length (converted into CCD pixels) plus a number of extra pixels, and a Y size equal to the number of CCD pixels to be extracted above and below the central wavelength position, as specified in the input grism table (see Figure 6.3.1, page 118).

All such rectangular subframes are extracted from the stacked images and inserted in the output combined flat image.

[10] It is well known that an alignment of the input images based solely on the images WCSs is not sufficiently accurate, and the described algorithm still needs to be improved in this respect. Preliminary tests have also indicated that the image alignments obtained with this recipe tend to be more inaccurate at larger offset between images. Work is in progress to solve these problems.

Obtaining exposures with different mask shutter positions is a way to eliminate contaminations from other multiplexed spectra (see Figure 5.21.1, page 96). This is never necessary for high resolution grisms, and in that case the combination of input flat fields becomes trivial.

7.19.2 Normalisation of the combined flat field

The combined flat field normalisation is carried out in four steps. First, each slit spectrum is extracted and remapped removing the spectral curvature; then it is heavily smoothed (with a filter or with a polynomial fit); the smoothed result is mapped back to its original reference; and finally the combined flat field image is divided by the image of smoothed spectra.

The result is the master flat field, that may be used by other data reduction recipes for applying a flat field correction.

The spectral optical and curvature models are necessary for a correct remapping of the spectra during the smoothing operation. For this purpose the "first guess" of the x component of the optical distortion model is improved, together with the "first guess" of the spectral curvature model, by tracing the edges of the slit spectra and fitting the found positions (see Section 6.3.3). The y component of the optical distortion model cannot be determined, because flat field spectra do not offer any landmark along the dispersion direction. For this reason the optical and the curvature distortion models derived with *vmspflat* cannot be considered consistent with the wavelength calibration determined with *vmspcaldisp*, and therefore cannot be used for any other purpose (including writing them to the VIMOS IWS configuration files).

A more detailed description of the spectral curvature determination is given in Section 7.20.3.

7.20 vmspcaldisp

This recipe carries out the following fundamental steps:

1. Creation of the combined arc lamp frame from the input raw arc lamp exposures.
2. Determination of the spectral optical distortion model from the combined arc lamp exposure.
3. Determination of the spectral curvature from the combined master flat field (if given: see Section 5.21).
4. Determination of the inverse dispersion solution.

A description of each step is given in the following sections.

7.20.1 Creation of the combined arc lamp frame

This operation is performed in the same way as for the creation of the combined flat field described in Section 7.19.1.

7.20.2 Determination of the optical distortion model

The x and the y components of the optical distortion models are determined by a bivariate polynomial fit of a list of corresponding mask and CCD positions (see Section 6.3.2).

To determine the CCD position of the reference wavelength λ_o corresponding to the coordinates (x, y) of the centre of a slit on the mask, a template of the expected pattern of lines in the close neighbourhoods of the reference wavelength position is built (see Figure 7.20.1).

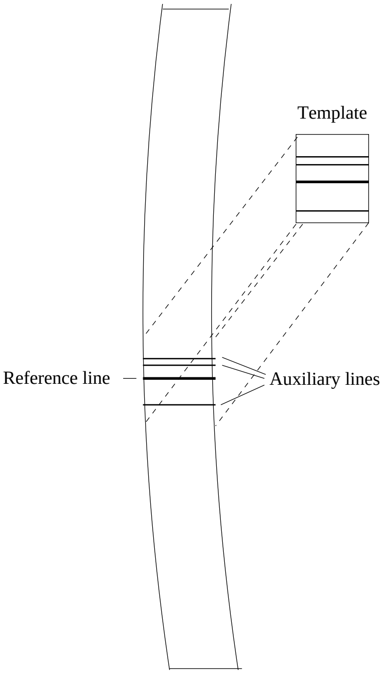


Figure 7.20.1: *Pattern matching for optical distortion model determination.*

The template (typically consisting of a pattern of 3–5 lines) is tried at different CCD positions around the expected position derived from the "first guess" optical distortion model, searching for a maximum match. To avoid flat maxima in the search of the best X coordinate, the pattern is also made as wide as the slit width on the CCD. The search is repeated at slightly different values of the dispersion (around the one expected at the position of the reference wavelength), to make the search robust against possible slight changes of the instrument distortions.

The pattern matching search is repeated for each slit of the mask, and when the list of matching mask and CCD coordinates is completed a new optical distortion model can be fit.

An optical distortion model fit is not performed if there are not enough slits: the number of slits must be at least twice the number of free model parameters. In addition to that, the slits coordinates on the mask must span at least 50 mm both in the x and in the y direction. If these conditions are not met, then the "first guess" optical distortion model is left untouched, and used "as is" in the remaining reduction steps.

7.20.3 Determination of the spectral curvature model

If a combined flat field is passed to *vmspcaldisp*, then the spectral curvature "first guess" model can be improved. The procedure is identical to the one applied in the *vmspflat* recipe, but the curvature model derived with *vmspcaldisp* is different because it is based on a better optical distortion model determination.

The spectral curvature model is refined starting from its "first guess", used in combination with the just derived optical distortion model. The edges of all the slit spectra are located and traced in the combined master flat, and are fit by a polynomial shape. Currently a second degree polynomial is used for this purpose. The *global* spectral curvature model is then determined by fitting the values of all the coefficients of all the local polynomials, as a function of the position (x, y) of the slits ends on the mask. The bivariate polynomial used in the fit has currently 2x2 free coefficients.

For details on the polynomials used in the spectral curvature modeling, see Section 6.3.3.

A spectral curvature model fit is not performed if there are not enough slits: the number of slits must be at least twice the number of free model parameters. In addition to that, the slits coordinates on the mask must span at least 50 mm both in the x and in the y direction. If these conditions are not met, then the "first guess" curvature model is left untouched, and used "as is" in the remaining reduction steps.

7.20.4 Determination of the inverse dispersion solution

Each CCD pixel laying on the image of a slit main axis (corresponding to the spectrum reference wavelength) is used as a starting point in extracting a 1D-spectrum from each row of each slit. Such extraction is performed following the spectral curvature model, that for each CCD pixel defines a *row* on the slit spectrum (see Figure 6.3.1, page 118).

As an *optional* preliminary step in the determination of the inverse dispersion relation, the spectrum corresponding to the central row of the slit is extracted. Next, a template of the arc lamp spectrum is created from the line catalog: the template consists of a list of wavelength intervals, each centred around a catalog wavelength and with a size depending on the slit width. When different windows overlap, they are merged into a single wavelength interval.

The extracted central spectrum is then compared to the arc lamp template, attempting to pre–tune the local solution derived from the "first guess" global IDS. The pre-tuning operation is performed in the following steps:

1. The logarithm of the extracted spectrum is computed.
2. The first guess polynomial relation between λ and Y CCD pixel is extracted from the "first guess" global IDS. A grid of sampling values is defined for the coefficients of this polynomial (currently the only modified coefficients are the constant term and the dispersion, *i.e.*, the coefficients d_o and d_1 defined in Section 6.3.4).
3. Using the polynomial corresponding to each node of the grid of coefficients values, the arc lamp template is transformed into pixel intervals on the extracted spectrum.
4. The logarithm of the extracted spectrum is integrated in the pixel intervals obtained at point 7.23.10. The result of this integral is used as a *match index*.

5. The polynomial having the coefficients corresponding to the the highest match index is selected, replacing the header "first guess" for the examined slit.

Once the "first guess" polynomial is pre–tuned, it is used to correctly identify the arc lamp lines in all slit spectra, and to determine their accurate position.

The line identification is done by selecting on the extracted spectrum the peak which is closest to its expected position. The position of the identified line is determined by a peak search algorithm run within a window of given size (see Section 5.21).

Once a table of identified lines positions is completed for the extracted spectrum, a polynomial transformation from catalog wavelength to pixel position is determined. Typically the polynomial transformation is of 3^{rd} order for low dispersion grisms, and 4^{th} order for higher dispersion grisms (see Section 6.3.4). This process is repeated for each row of each slit of the mask, and all the solutions are written to the output extraction table (see Table 5.21.3, page 92).

The global IDS model is determined by the bivariate polynomial fitting of the coefficients of the local polynomials, as described in Section 6.3.4. Currently this model has 3x3 free parameters.

An inverse dispersion solution is not determined if there are not enough slits: the number of slits must be at least twice the number of free model parameters. In addition to that, the slits coordinates on the mask must span at least 50 mm both in the x and in the y direction. If these conditions are not met, then the "first guess" global IDS is left untouched, and used "as is" in the remaining reduction steps.

Plots of arc lamp spectra from different grisms and lamps combinations are given in Figures 7.20.2–7.20.10, where the spectral lines used in the wavelength calibration are marked.

A list of all the used calibration lines available from the Helium, Argon and Neon lamps within the spectral range of all the VIMOS grisms is given in tables 7.20.1 and 7.20.2.

Wavelength (Å)	**Element**	**Note**
3888.650	He	
4026.190	He	
4471.480	He	
4713.143	He	
4921.929	He	
5015.675	He	
5400.560	Ne	
5764.418	Ne	
5852.488	He	
5875.618	He	
5944.834	Ne	
6029.997	Ne	
6074.338	Ne	
6096.163	Ne	
6143.062	Ne	
6163.594	Ne	
6217.281	Ne	
6266.495	He	
6304.789	Ne	
6334.428	Ne	
6382.991	Ne	
6402.246	Ne	
6506.528	Ne	
6532.882	Ne	
6598.953	Ne	
6678.150	He	
6678.200	He+Ne	blend 6678.150 + 6678.280
6678.280	Ne	
6717.043	Ne	
6929.468	Ne	
6965.430	Ar	
7032.413	Ne	
7065.188	He	
7173.939	Ne	
7245.167	Ne	

Table 7.20.1: *Arc lamp calibration lines available in the VIMOS spectral range (continued in Table 7.20.2).*

Wavelength (Å)	Element	Note
7272.936	Ar	
7281.349	He	
7383.980	Ar	
7438.899	Ne	
7488.872	Ne	
7503.867	Ar	
7507.000	Ar	blend 7503.867 + 7514.651
7514.651	Ar	
7535.770	Ne	
7544.046	Ne	
7635.105	Ar	
7723.800	Ar	blend 7723.76 + 7724.21
7948.175	Ar	
8006.156	Ar	
8012.000	Ar	blend 8006.156 + 8014.786
8014.786	Ar	
8103.692	Ar	
8110.000	Ar	blend 8103.692 + 8115.311
8115.311	Ar	
8264.521	Ar	
8300.326	Ne	
8377.607	Ne	
8408.210	Ar	
8415.000	Ar	blend 8408.210 + 8424.647
8424.647	Ar	
8495.360	Ne	
8521.441	Ar	
8634.648	Ne	
8654.383	Ne	
8667.943	Ar	
8780.622	Ne	
8853.866	Ne	
9122.966	Ar	
9224.500	Ar	
9657.780	Ar	

Table 7.20.2: *(Continued from Table 7.20.2) Arc lamp calibration lines available in the VIMOS spectral range.*

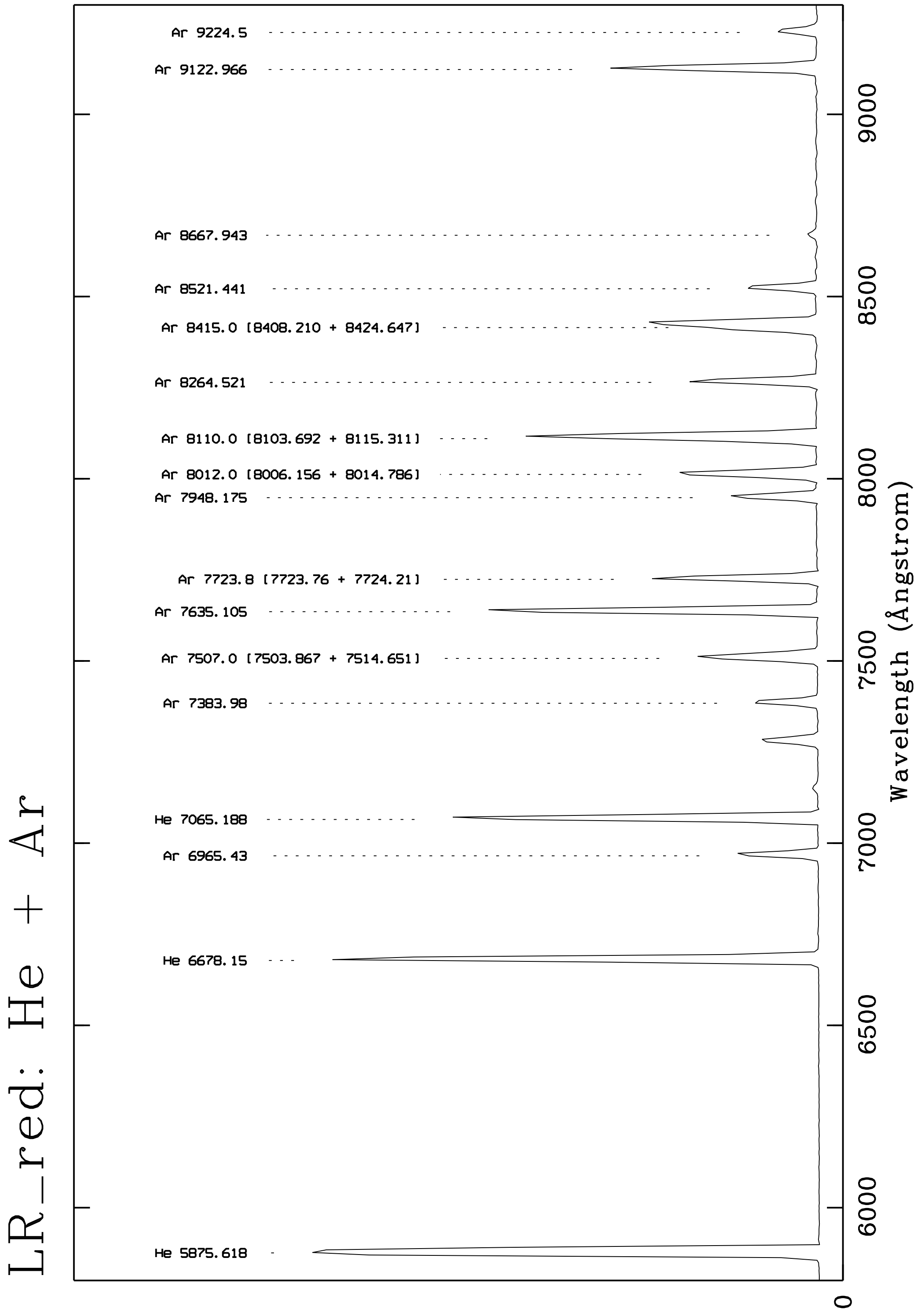


Figure 7.20.2: *LR red arc line spectrum from 5800 to 9300 Ångstrom. The lines used for calibration are indicated.*

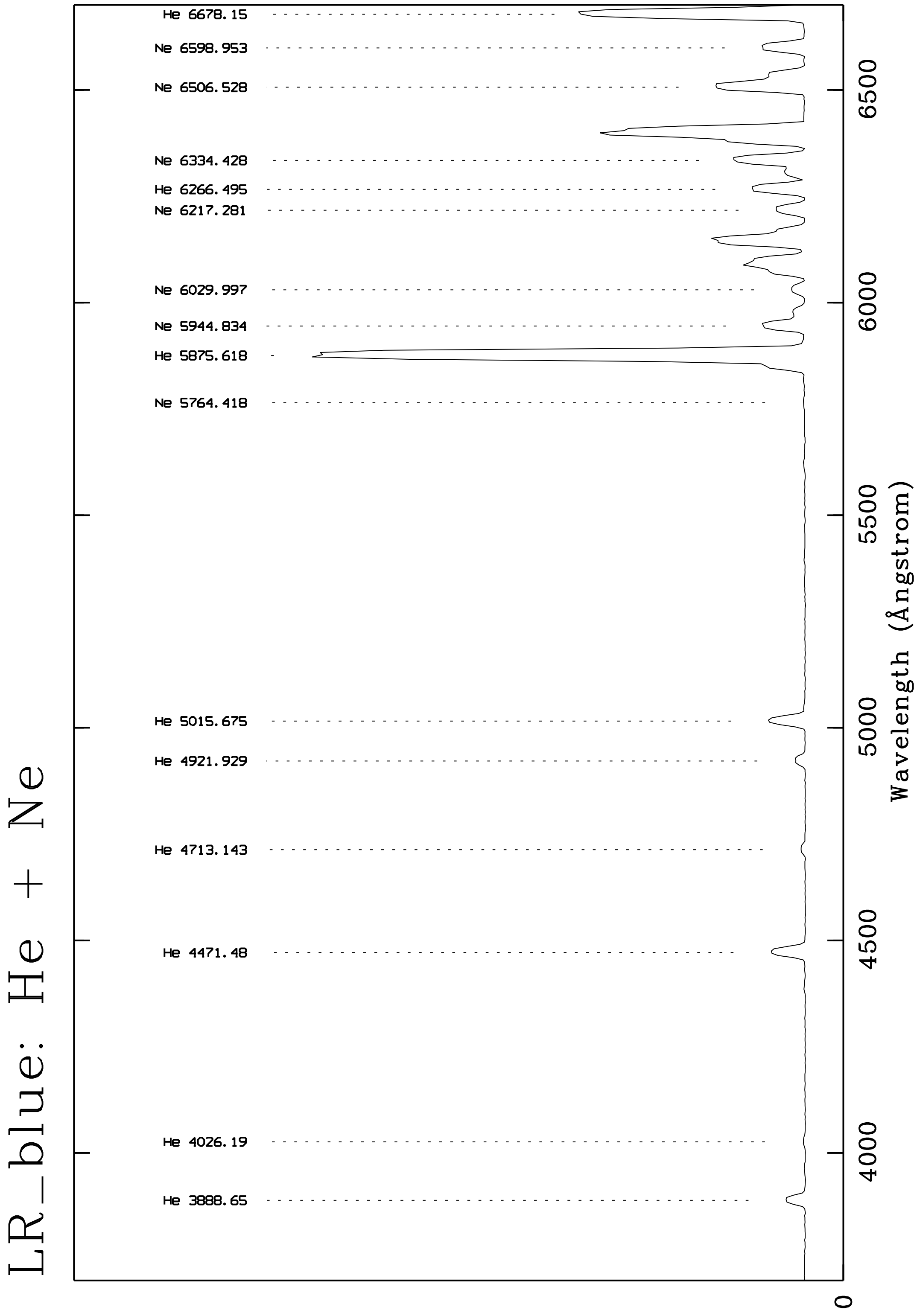


Figure 7.20.3: *LR blue arc line spectrum from 3800 to 6750 Ångstrom. The lines used for calibration are indicated.*

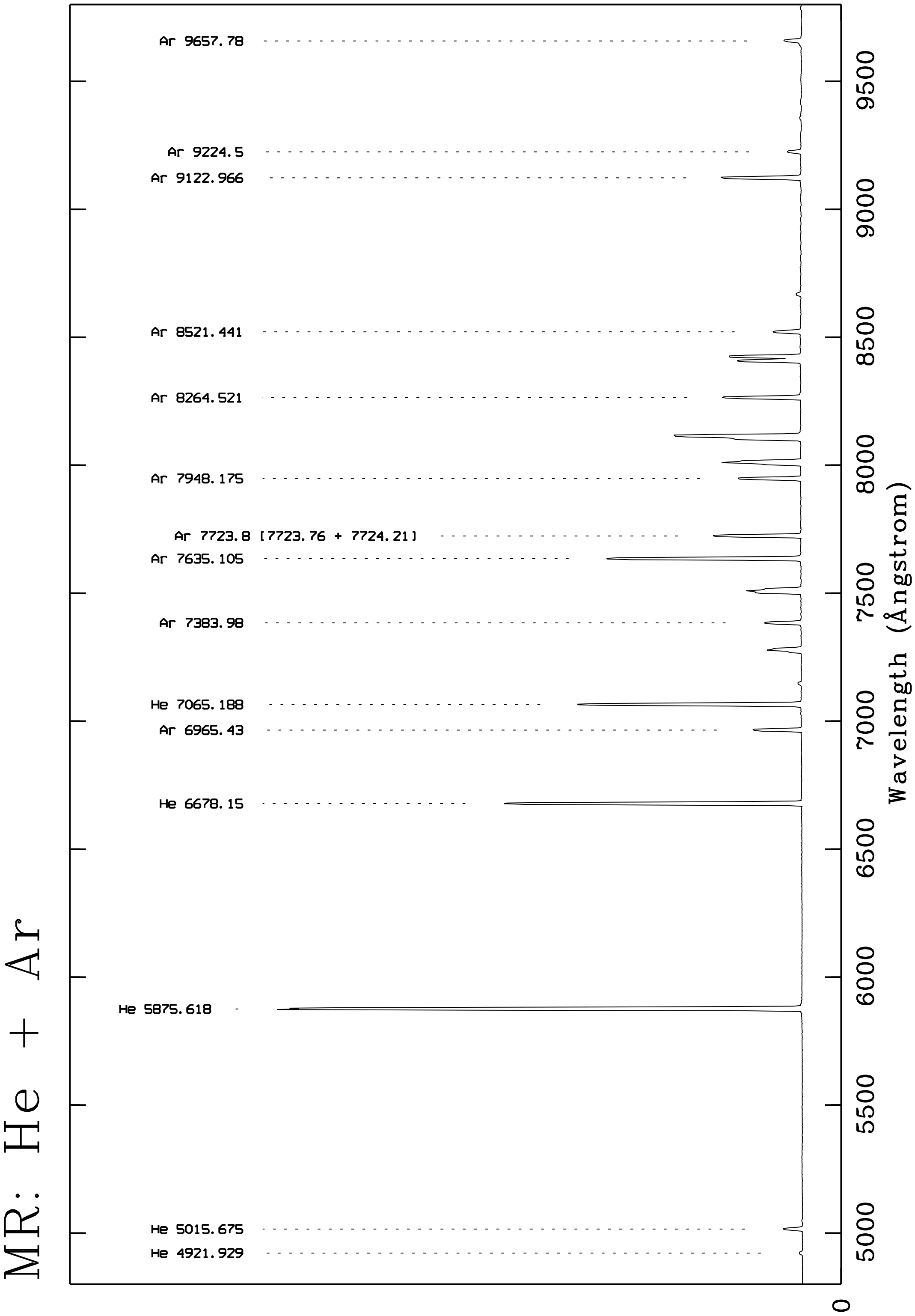


Figure 7.20.4: *MR arc line spectrum from 4800 to 9800 Ångstrom. The lines used for calibration are indicated.*

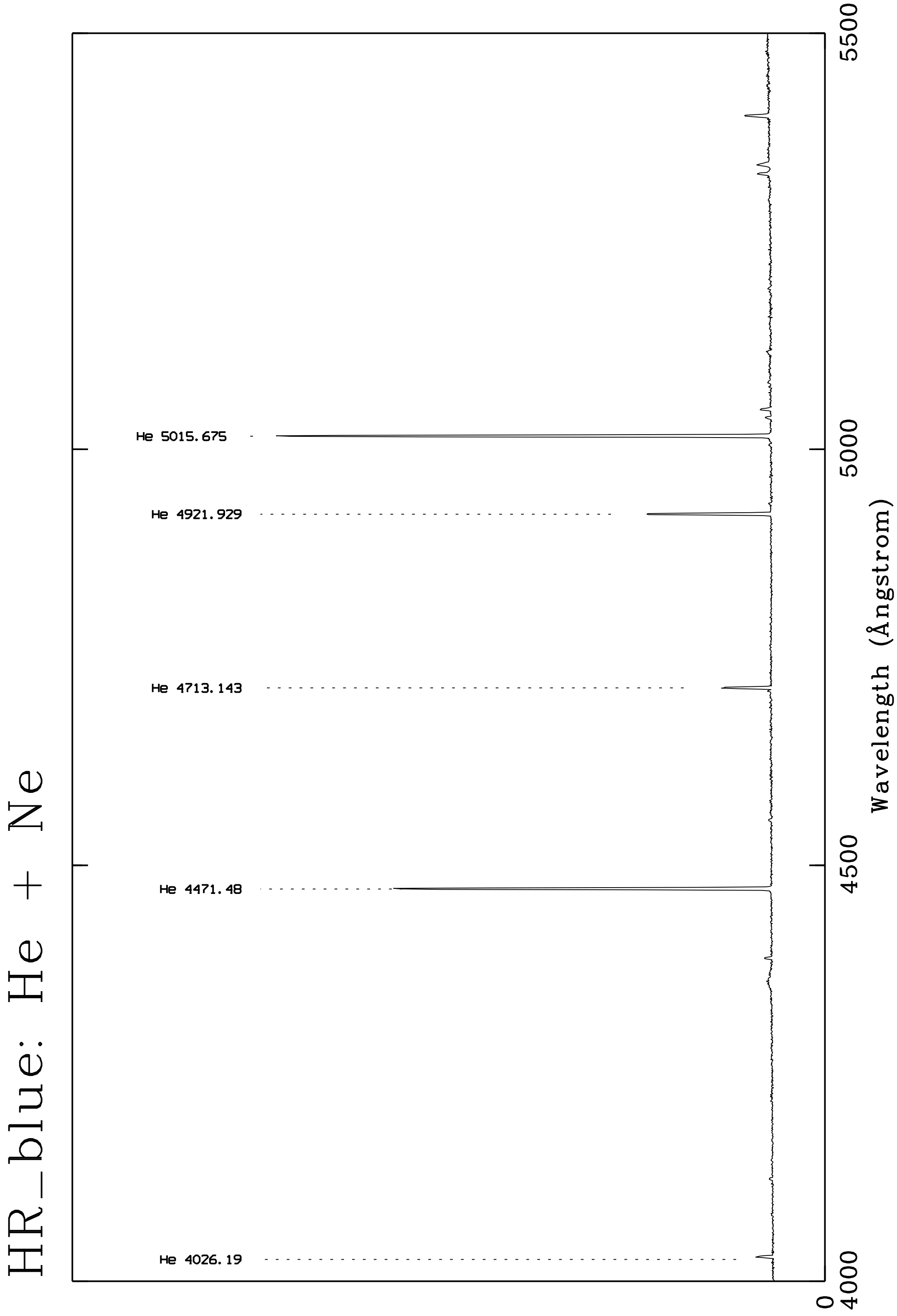


Figure 7.20.5: *HR blue arc line spectrum from 4000 to 5500 Ångstrom. The lines used for calibration are indicated.*

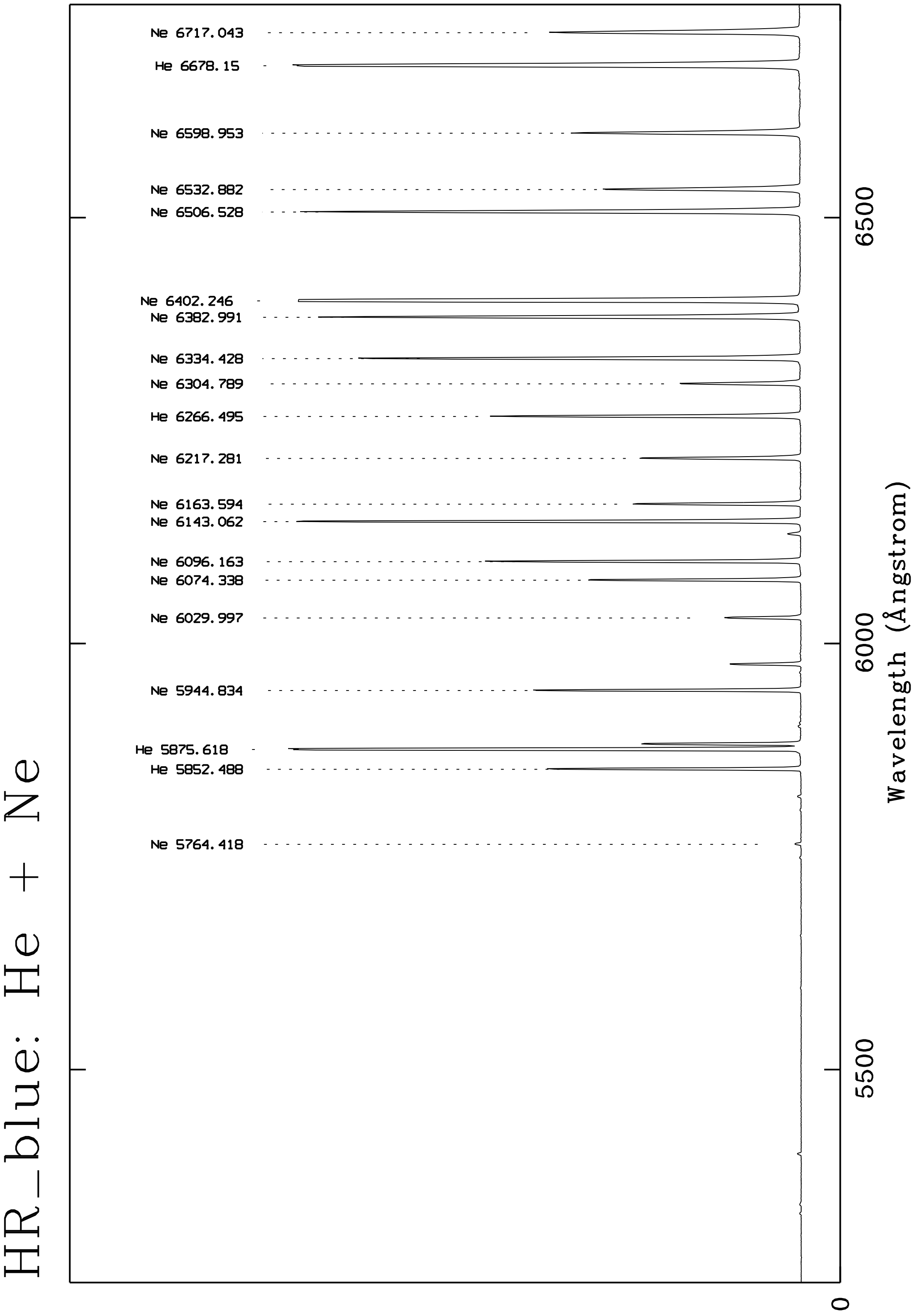


Figure 7.20.6: *HR blue arc line spectrum from 5250 to 6750 Ångstrom. The lines used for calibration are indicated.*

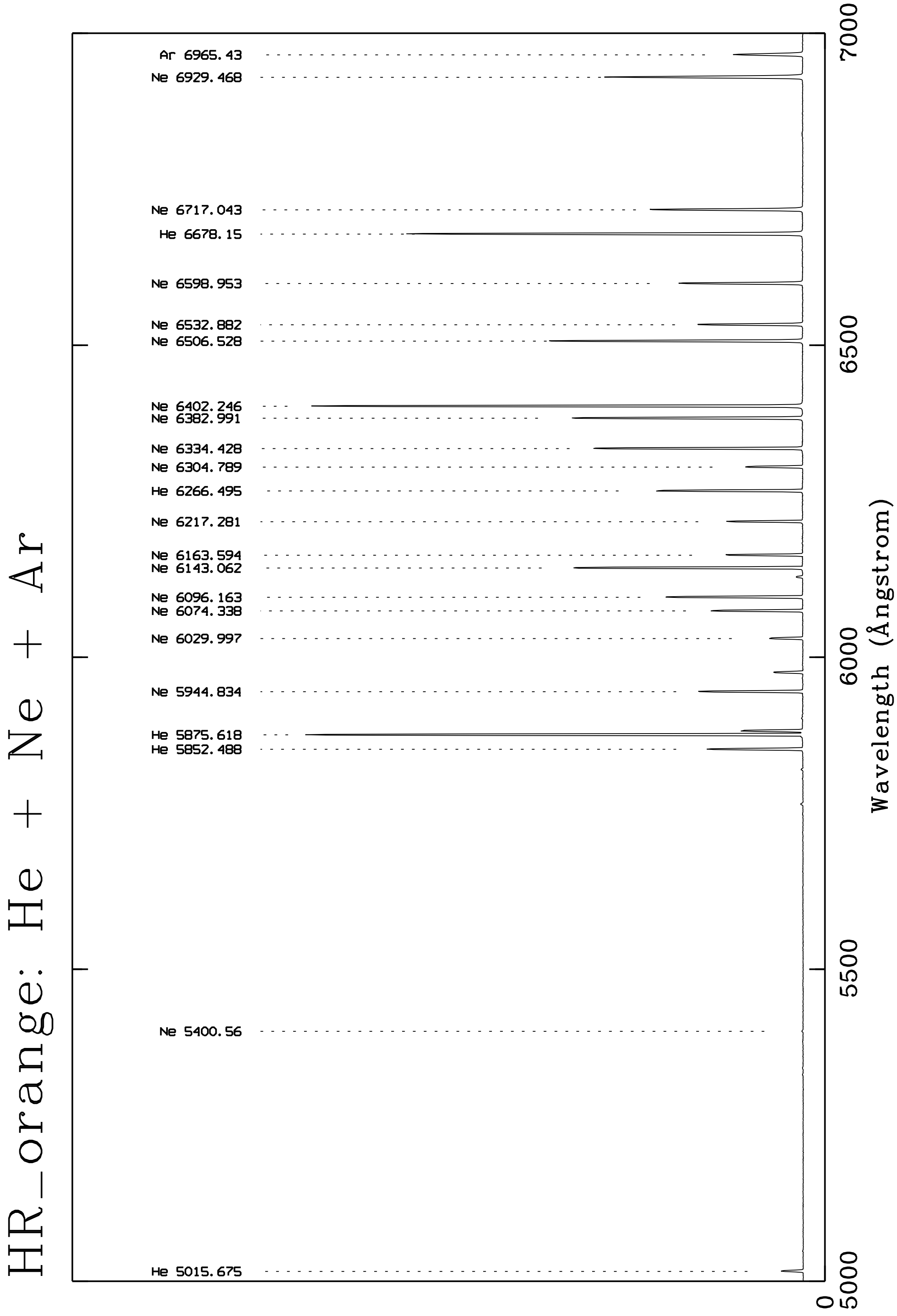


Figure 7.20.7: *HR orange arc line spectrum from 5000 to 7000 Ångstrom. The lines used for calibration are indicated.*

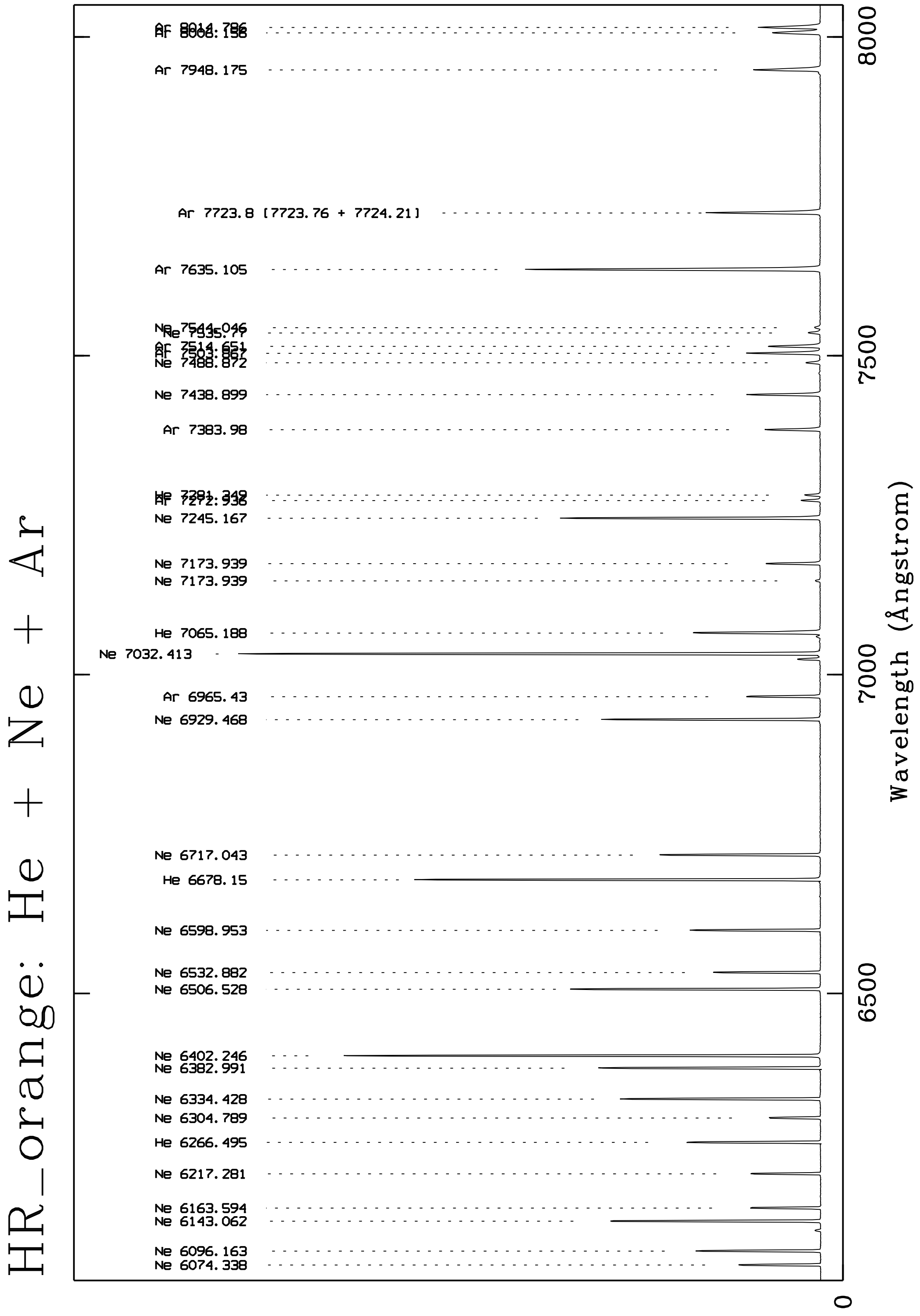


Figure 7.20.8: *HR orange arc line spectrum from 6050 to 8050 Ångstrom. The lines used for calibration are indicated.*

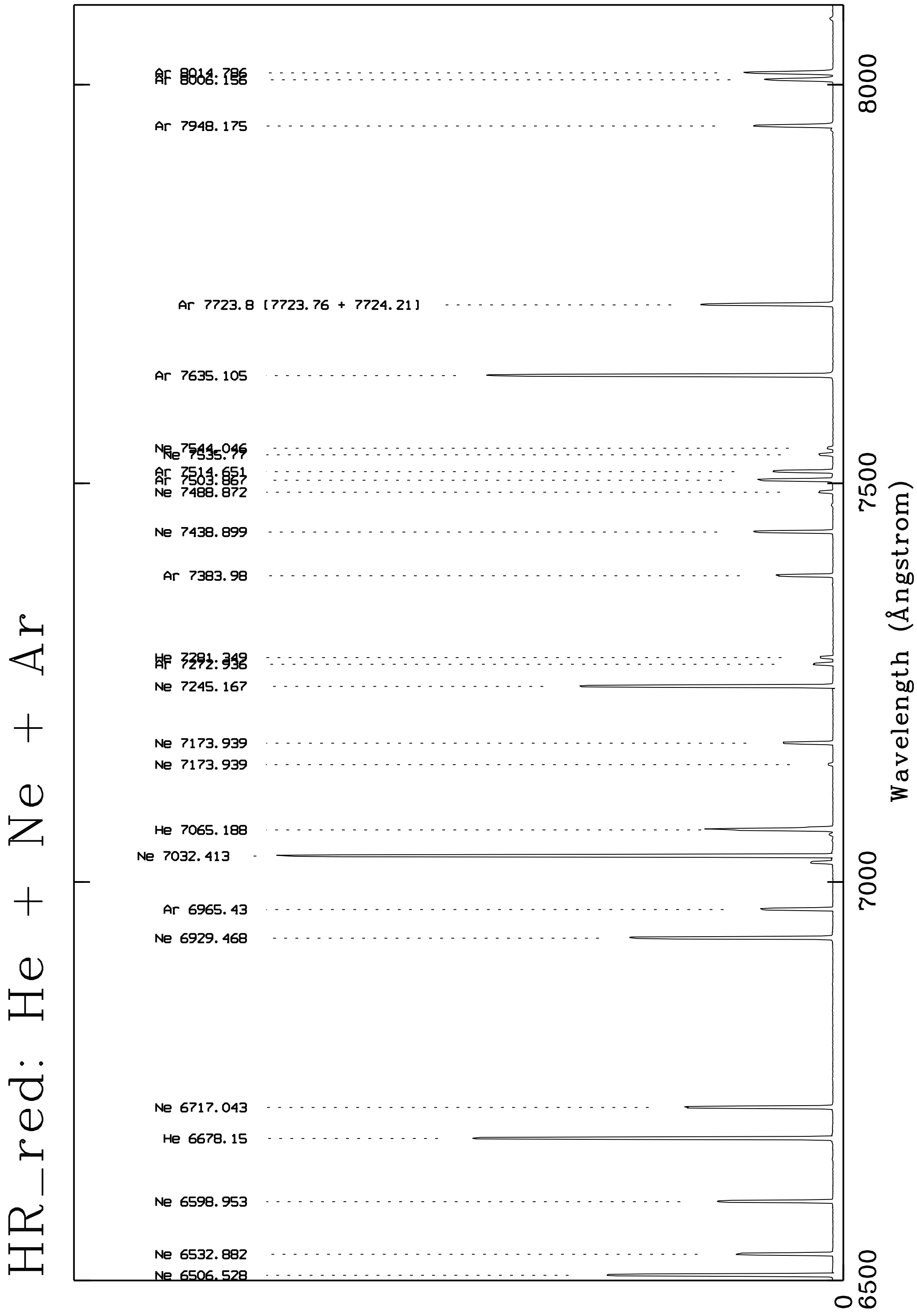


Figure 7.20.9: *HR red arc line spectrum from 6500 to 8100 Ångstrom. The lines used for calibration are indicated.*

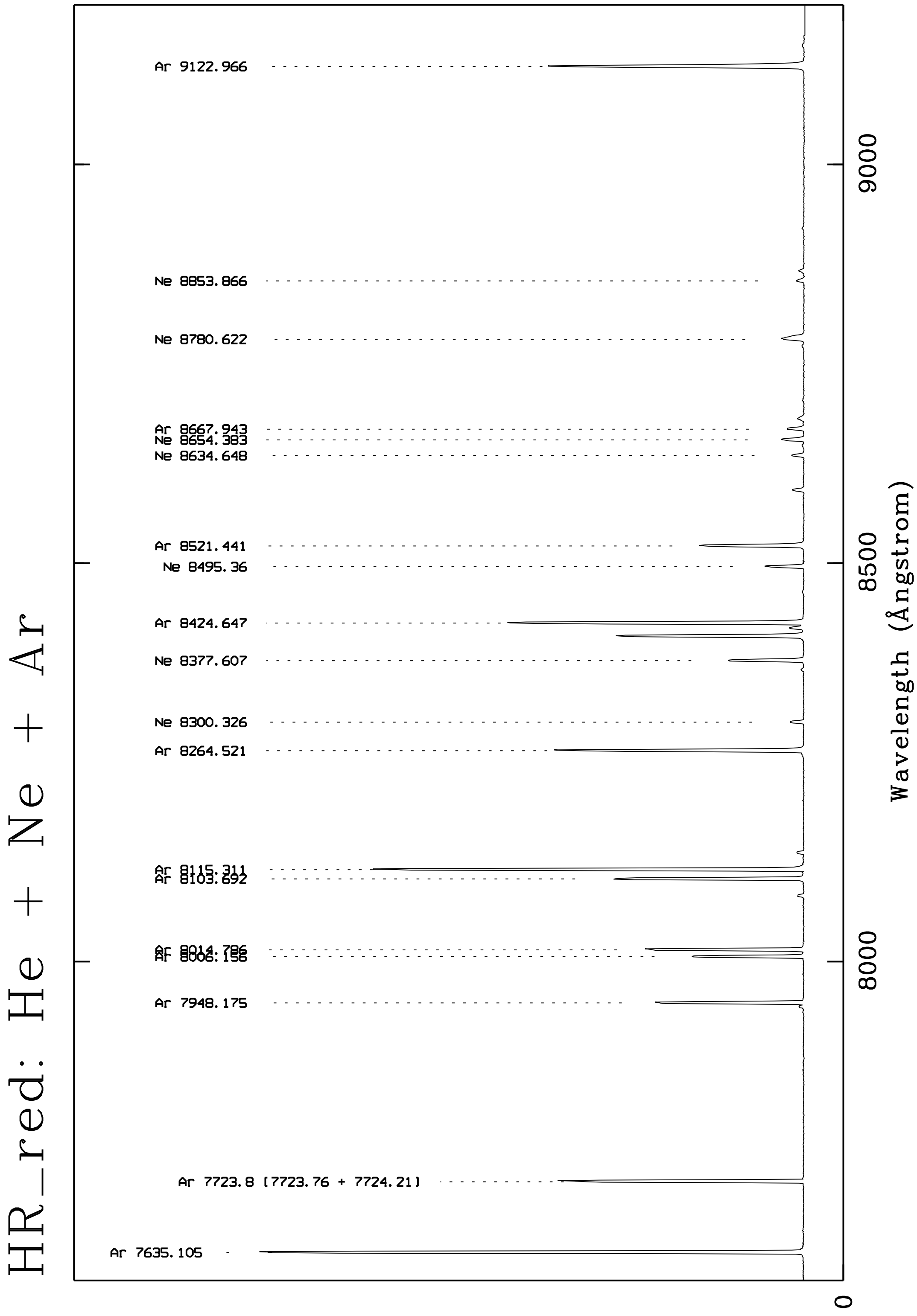


Figure 7.20.10: *HR red arc line spectrum from 7600 to 9200 Ångstrom. The lines used for calibration are indicated.*

7.21 vmmosobsstare

This recipe is used to reduce one exposure made in MOS mode. The following fundamental steps are carried out:

1. Bias subtraction (see Section 7.3).
2. Optional dark subtraction (see Section 7.4).
3. Optional flat field correction (see Section 7.5).
4. Optional bad pixel cleaning (see Section 7.1).
5. Align distortion models to sky lines positions.
6. Slit spectra extraction.
7. Object detection.
8. Sky modeling and subtraction from extracted slit spectra.
9. Object extraction.

Beyond the standard reduction steps, described in some detail in the indicated sections, only the last five steps need to be outlined here.

7.21.1 Align distortion models to sky lines positions

A set of sky lines, dependent of the spectral range covered by the grism in use, is taken as reference for aligning the spectral distortion models to a possible variation of the instrument mechanical flexures. For each slit spectrum, the distortion models derived from the calibration data are used to look for the reference sky lines around their expected positions along the spectral dispersion direction. The median offset between the expected and the observed positions is added to the Y component of the optical distortion model (see Section 6.3.2, page 116), that will be then used in the extraction of the slit spectra.

7.21.2 Slit spectra extraction

Each slit spectrum is read from the pre-processed input image, following the shapes of the modeled spectral distortions (see Section 6.3.5). The slit spectra are always resampled along the cross-dispersion direction at a 1-pixel step, and along the dispersion direction at the wavelength step defined by the keyword `PRO WLEN INC` of the used grism table, ensuring that the flux is conserved. The resampled values are written to an output image that contains all the rectified slit spectra aligned in wavelength (with slit 1 at bottom).

7.21.3 Object detection

After the slit spectra are all extracted and rectified, a detection algorithm is run to locate the emission of possible objects. The mean spatial profile of each slit spectrum is computed in the wavelength range specified in the used grism table (see Section 5.22, page 97), excluding cosmic ray events. The profile is analysed, looking for signal significantly above the background noise.

The background level and its associated noise are determined empirically, computing iteratively a robust estimate of the most probable value in the profile, and excluding at each iteration all values with positive residuals greater than their population RMS. The final population RMS is taken as the noise level.

All the profile values exceeding the background level by a specified number of noise sigmas are selected. Each unbroken sequence of selected profile values is taken as an object candidate. If a sequence of values is shorter than a specified amount (see parameter *MinObjectSize*, Table 5.22.5, page 100) it is rejected. If a sequence of values is longer than another specified amount (see parameter *MaxObjectSize*), an attempt is made to unblend the object into sub-objects, that will be then extracted separately. The unblending operation is made using a watershed algorithm to analyse the structure of the sub-profile and to reassign portions of the sequence to the identified sub-objects.

7.21.4 Sky modeling

Once the positions of the detected objects on the extracted and rectified slit spectra are determined, it is possible to analyse the values of the sky pixels to estimate the sky level within the regions containing the objects. For each wavelength, the median sky level may be computed and replaced in the objects regions or, alternatively, a polynomial fit may be tried on the sequence of available sky values, and the interpolated values used at the object positions. In all cases, the sky values — real or computed — are subtracted from the extracted slit spectra.

7.21.5 Object extraction

The object spectra are extracted from the rectified and sky subtracted slit spectra. The method used may be a simple aperture extraction, where all the signal included in the object region is integrated, or, alternatively, an optimal extraction, *i.e.* an average of the signal optimally weighted by a function of the signal noise. The optimal extraction takes also care of removing the cosmic rays contamination, and resolves the possible effects of a residual spectral curvature. The algorithm used is the one described by K.Horne (1986), in PASP vol.98, p.609. It should be noted that this method is not suitable for extended or blended objects, or for objects only consisting of emission lines (with no continuum), where the object profile might not be modeled properly.

While the aperture extraction is applied to the regions obtained by the object detection task and specified in the window table (see table 5.22.3, page 98), the optimal extraction is applied to a region having twice that size, provided that this wouldn't exceed the slit boundaries.

7.22 vmmosobsjitter

This recipe is used to reduce a sequence of exposures made in MOS mode. The following fundamental steps are carried out for each input frame:

1. Bias subtraction (see Section 7.3).
2. Optional dark subtraction (see Section 7.4).
3. Optional flat field correction (see Section 7.5).
4. Optional bad pixel cleaning (see Section 7.1).
5. Align distortion models to sky lines positions (see Section 7.23.6).
6. Slit spectra extraction (see Section 7.21.2).
7. Object detection (see Section 7.21.3).
8. Sky modeling and subtraction from extracted slit spectra (see Section 7.21.4).

The following steps involve all the images containing the extracted and sky subtracted slit spectra:

1. Align and combine the processed images.
2. Iterate the object detection on the combined image (see Section 7.21.3).
3. Object extraction from the combined image (see Section 7.21.5).

Beyond the reduction steps described in some detail in the indicated sections, only the image alignment needs to be outlined here.

7.22.1 Align and combine the processed images

The object detection task applied separately to each processed frame is expected to lead to the detection of a number of objects (at least the brightest ones). Their positions are then correlated between images, leading to the determination of a median offset valid for each exposure relatively to the first exposure of the jittered sequence. The images are aligned and resampled accordingly, before being combined using a median combination method (see Section 7.6).

7.23 Overview of the VIMOS IFU data reduction procedure

In this section, an overview of the IFU data reduction procedure is given. This procedure is not yet completed in the current pipeline release, and it needs further testing for evaluating its robustness and reliability. The distributed IFU pipeline recipes should be considered as a beta release offered for evaluation to a wider community of users.

7.23.1 Required data

For the described data reduction three different types of exposures are required:

- Flat field lamp exposure
- Arc lamp exposure
- Science exposure

where it is assumed that all the datasets have been bias subtracted (see Section 7.3, page 123).

The data reduction strategy is based on the idea of aligning the tracings of the flat field spectra directly to the tracings of the brightest science spectra. This alignment would compensate the traslation and the rotation of the spectra caused by the instrument mechanical instabilities, making it possible to optimally extract all the scientific spectra. In general the flexure component along the dispersion direction would not be accurately determined in this way (for geometrical reasons), but this is then solved by aligning the wavelength calibration obtained from the arc lamp exposure to the sky lines in the science exposure, or by applying a model of the instrument flexures along the dispersion direction.

This method is advantageous because it does not require any calibration to be taken during the night. Data reduction would be possible with flat field and arc lamp exposures obtained at daytime, provided that the following conditions are fulfilled:

1. The instabilities of the instrument (flexures) and of the IFU mask (mechanical play) would never introduce offsets larger than 2 pixels on the spectra positions on the CCD[11].
2. At least one spectrum of the science exposure should be traceable (with the current method a spectrum begins to be safely traceable when the signal reaches 50 e^-/pixel).

A second strategy may be applied in the unfortunate case that no science spectra are traceable. In such a situation the alignment of the flat field tracings to the science could only be based on other sources. For instance, an arc lamp exposure may be obtained before and after the scientific observation. The two exposures would be correlated to obtain the differential flexure, and then the standard tracing solution from the day flat field calibration would be aligned to the mean flexure position obtained. In the current data reduction system this method is not yet implemented, and when no traceable scientific spectra are available the extraction mask obtained from the flat field exposure is used without alignment.

[11] As of today (July 2004) this condition is not yet fulfilled, and therefore night calibrations are still mandatory.

7.23.2 Fibers identification

Please refer to Section 3.5, page 18, for the conventions used in numbering the IFU components.

By fibers identification we mean here the correct association of a fiber position on the IFU mask to a corresponding position on the CCD. If a conventional Y_o coordinate (*i.e.*, measured along the dispersion direction) is chosen on the CCD, the fiber identification would consist of assigning to each fiber its X position along this reference line (see Figure 7.23.1).

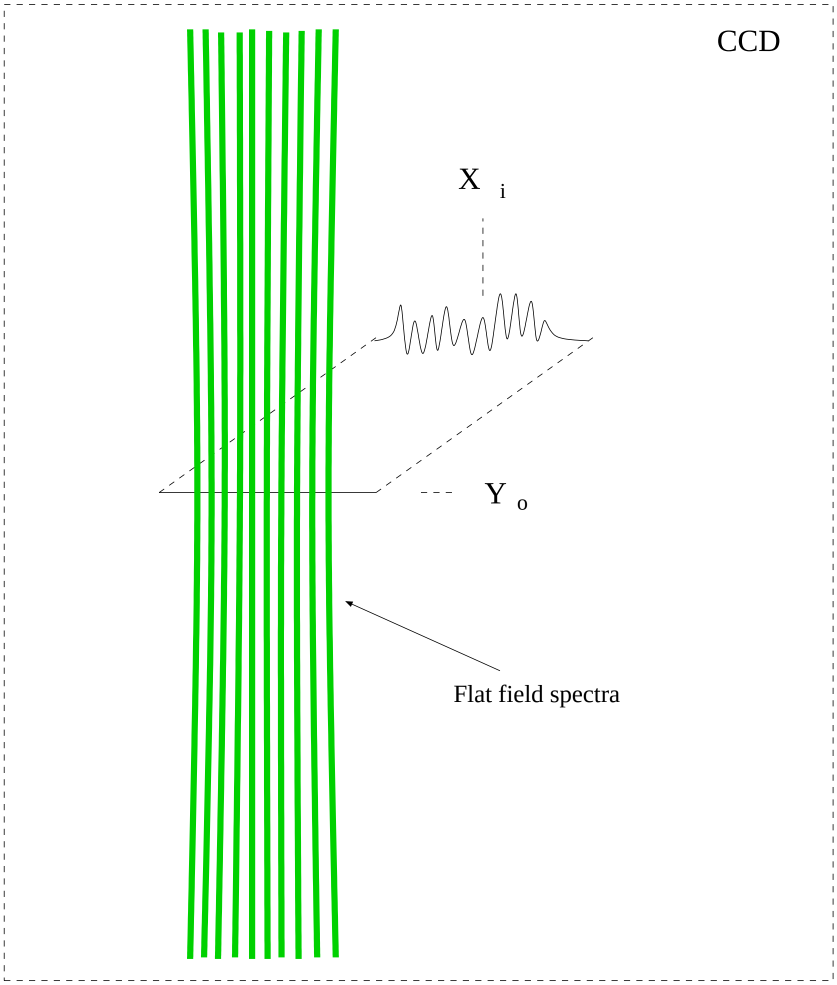


Figure 7.23.1: *The reference Y_o coordinate in the fiber identification procedure is conventionally fixed.*

Such an identification is given even in the case a fiber is not visible, either because it is damaged, or lost in the vignetted part of the CCD, or even purposefully masked.

The fiber spectra identification is always carried out on a flat field exposure. Preliminary identifications, performed manually on a set of reference flat field exposures (one for each grism/quadrant combination), are available in the calibration directory. Such safe identifications are used as reference and transferred to any other flat field by cross-correlation. With this method it is possible to safely identify fibers even in presence of major instabilities of the instrument (currently the correlation radius is set to 10 pixels).

Alternatively fibers may be even identified without any first-guess on the CCD position of the fiber signal. This blind fibers identification is based on a folding analysis on the reference row of the input IFU flat field exposure, to roughly determine the position of the gaps between blocks. This method safely rejects false gaps due to IFU head masking, dead fibers, and bad CCD columns. After the positions of the gaps are determined, a correlation of each 80-fibers block with a grid of 80 5-pixel-spaced points is performed, leading to the final identification of the fibers within each block.

The blind fiber identification would not be bullet-proof in the case that either the first or the last fiber block is cut by the vignetted part of the CCD. Rather, the ambiguity introduced by the possible loss of fibers at the other end of the block would be inherent to the data, and there would be no way to safely identify the visible fibers (but by a judgment *a posteriori* on the quality of the reduced data).

Both the described methods are applicable by recipe *vmifucalib* (see Section 5.24, page 101).

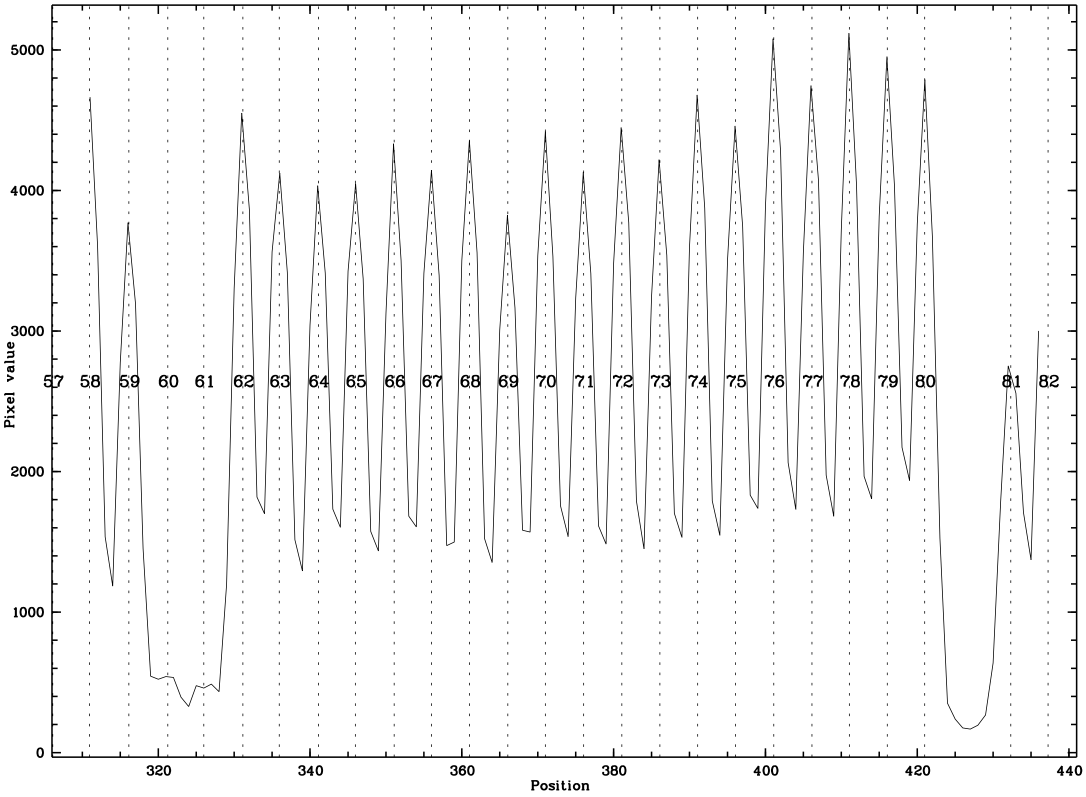


Figure 7.23.2: *Fiber identification along the cross-dispersion direction. The gap between the first and the second blocks of fibers is visible on the right. On the left the positions of two fibers lost to the IFU head shutter are marked.*

7.23.3 Tracing spectra

The tracing of each fiber flat field spectrum is a relatively simple matter, given the typically high S/N ratio reached in flat field exposures. The start X positions for the tracing, at a conventional Y_o coordinate on the CCD, are those obtained by the fiber identification task (see Section 7.23.2, page 153). Such positions are then used as first-guesses for the peak positions at the $Y_o + 1$ and $Y_o - 1$ coordinates, that are respectively used as first-guesses for the $Y_o + 2$ and $Y_o - 2$ coordinates, and so on, till some predefined limits set for spectral extraction are met.

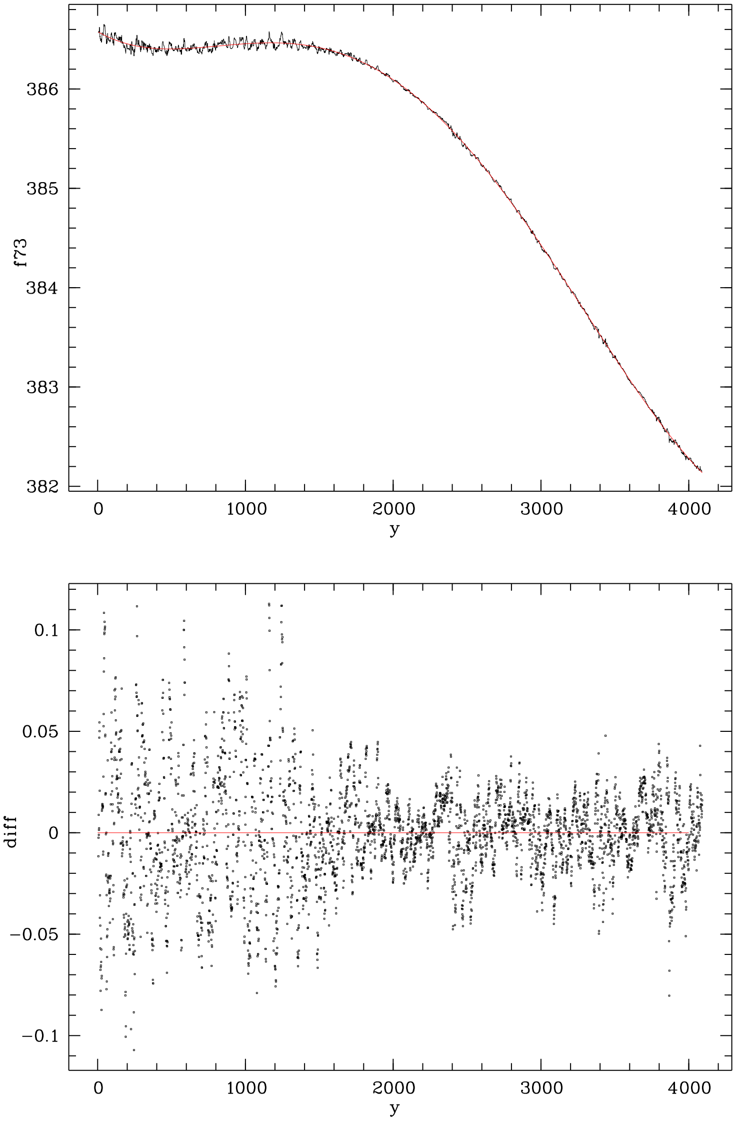


Figure 7.23.3: *Tracing of fiber spectrum 73 of HR_orange flat field in quadrant 3. Both axis are in pixels.*

The list of X positions obtained for each fiber at each Y coordinate would then be modeled by a low degree polynomial, with the intent of eliminating the outliers and increasing the tracing accuracy. In the following this polynomial will be indicated with $X_i(Y)$, where i is the fiber sequence number.

During the spectral tracing operation, the presence of damaged or lost fibers will also be determined. If for a given fiber the search for a peak will fail beyond a given predefined rate, then the corresponding fiber will be flagged as "dead", and will not be treated in the science spectra extraction.

The trace operation is carried out on a median filtered image of a flat field exposure (currently the smoothing box is set to 1x15 pixels), to avoid the derailment of the tracing by cosmic rays, bad pixels, or zero-order contamination in the case of LR observations. The accuracy of the tracing is typically of 0.01 pixel (see Figure 7.23.3).

The tracing of all the detected flat field spectra, performed by the recipe *vmifucalib* (see Section 5.24, page 101) leads to the definition of the extraction mask that will be used in the extraction of the scientific spectra associated to the flat field.

7.23.4 Background subtraction

It can be seen that straylight is absent, or negligible, in VIMOS IFU scientific observations.

However, this is not the case for flat field exposures, where a straylight apparently correlated with the strong spectral signal can be observed. The background level is about 10% of the illumination level. This has no negative effect on the spectral tracing, but it may introduce a non-negligible bias in the determination of the relative transmission correction factors for each fiber.

If a complete tracing solution is available for a given exposure, it is possible to precisely locate the regions where the background level can be evaluated, *i.e.*, along the gaps between the 80-fibers blocks, and in any other portion not containing spectra. The background pixel values are fitted by a low degree bivariate polynomial, and this model values are interpolated and subtracted from the original image [12].

7.23.5 Determination of the fiber profiles

On a background subtracted flat field exposure, the first and the last spectra of each 80-fibers blocks are considered. The 10 half-profiles facing the background regions can be used to determine the fiber profile down to the zero signal level.

Preliminarily, following the accurate tracing of each fiber, the interpolated maximum value of the fiber signal at each coordinate along the dispersion direction is determined. Such values will be used in the normalisation of the pixel values obtained at each Y coordinate.

All the pixel values of the half profile are assigned to their distance from the profile centroid derived from the tracing solution (see Figure 7.23.4) up to a distance of 5 pixels where the ground level is reached.

Thanks to the spectral curvature, different pixelisations of the fiber profile are available at different positions along the dispersion direction. The fiber flat spectrum is cut into intervals each of the order of hundreds of pixels, and for each one of these intervals all the pixel values contributing to the profile are normalised to the value of

[12] This is not implemented in the current release.

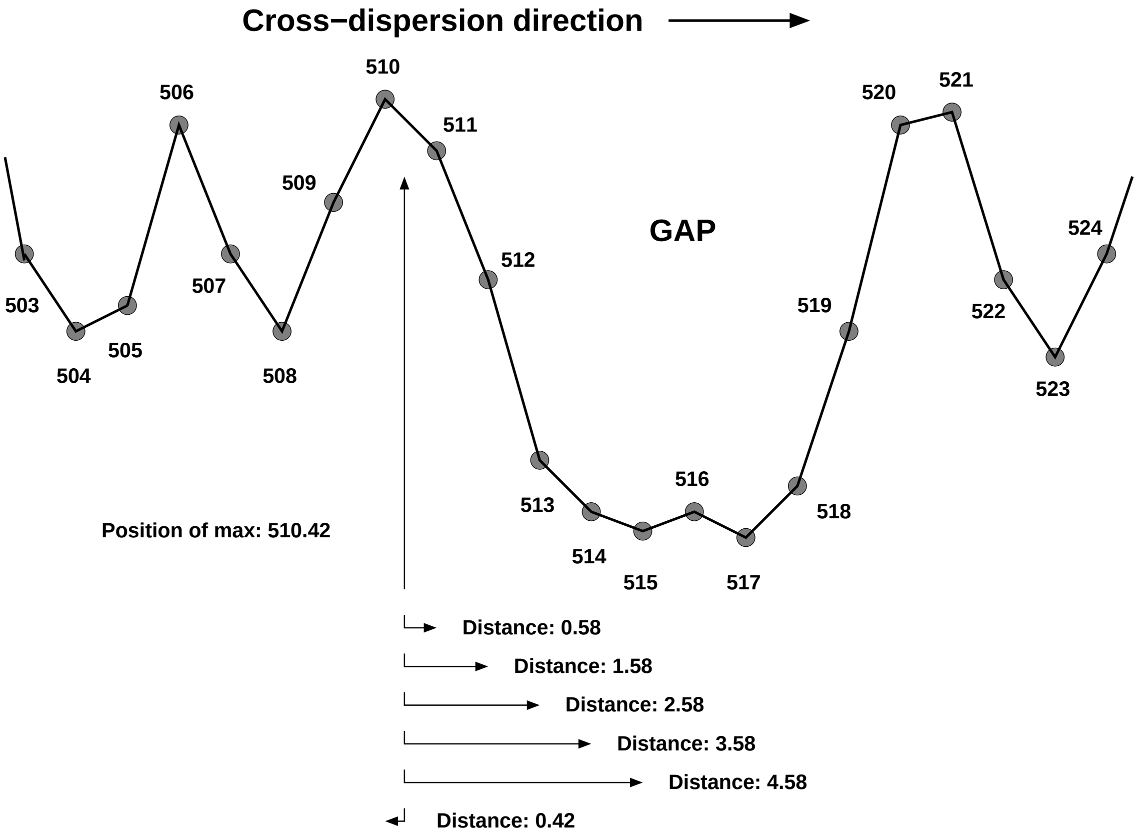


Figure 7.23.4: *Reading the fiber profile. In this example, some fiber profiles around a gap between two fibers blocks are shown. The last fiber profile at the left of the gap peaks at the pixel position 510.42 (obtained from the tracing solution). The pixels distances from the peak position are assigned to each measured pixel value. Also, the value of the pixel immediately before the position of the maximum is used, since the contamination from the previous fiber is shown to be negligible (at about 1/3000 of the peak intensity).*

the peak maximum computed as described above. If the contributions from the 10 different fiber profiles are shown to be consistent with each other they may be merged into a single dataset.

In this way an interval of about 5 pixels is populated with direct evaluations of the empirical profile of the fiber spectrum. These values are then averaged within a grid of predefined bins, and tabulated.

Any change in the profile shape as a function of the chosen interval along the dispersion direction can also be modeled. Whether or not this will turn out to be necessary in the VIMOS IFU case will be decided as soon as a complete study of the reconstructed profiles is realised.

As a preliminary result it seems that there is negligible colour dependency of the reconstructed profile (less than 2% between profiles reconstructed from red and blue regions), and similarly negligible differences between the $\sim$ 100 studied fibers (with few exceptions, probably due to actual physical differences between fibers).

Examples of reconstructed fiber profiles are given in Figures 7.23.5 and 7.23.6. Each plot contains about 24,000 pixel values covering the entire spectral range. The increase of noise in portions of the HR_blue plot is due to pixel values coming from the faintest part ($< 50e^-$/pixel) of the spectrum. Similar plots from LR grisms are perfectly consistent with the ones obtained from HR data, but contain less points. It should be noted that the slightest error in the tracing solution (> 0.05 pixel) would introduce major discontinuities in plots like the ones in Figure 7.23.5.

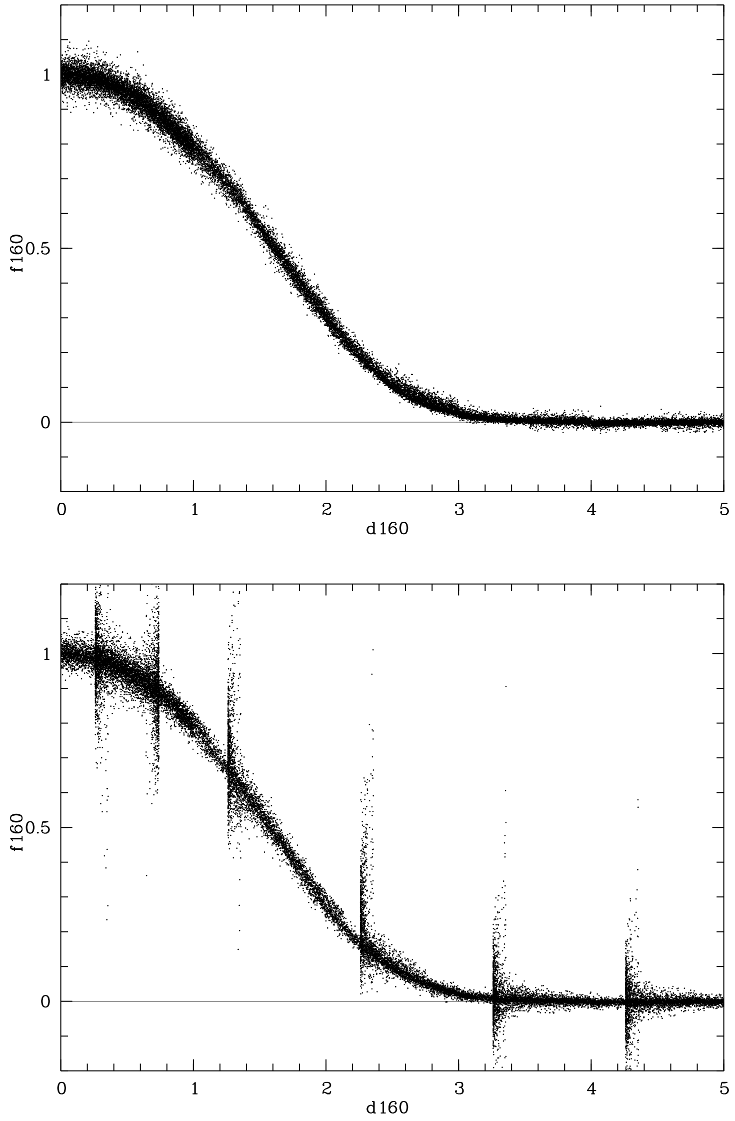


Figure 7.23.5: *Observed cross-dispersion profiles of spectra from fiber 160 on the central slit of quadrant 3.*

The modeled curve is the convolution of the true fiber profile with the pixel box. This is exactly what is needed in the spectral extraction process: any point of this curve is the value that a pixel at that distance from the profile centroid would have. This makes the reconstruction of the profile to be used in the spectral extraction task at each pixel position along the dispersion direction trivial.

It should be noted that, if such a curve is reconstructed properly, the sum of all of its values sampled at a 1-pixel step should be approximately constant (for the curves in Figure 7.23.6 this constant is ~ 3.26). For convenience, the fiber profile model may be normalised to make this constant equal to 1, that is:

$$\sum_i P(x_o + i) = 1$$

where $P(x)$ is the normalised fiber profile having the maximum at $x = 0$, x_o an arbitrary distance, and i any integer number between $-\infty$ and $+\infty$. It can be shown that $P(x)$ is not a gaussian, and it's not even the convolution of a gaussian with a box profile.

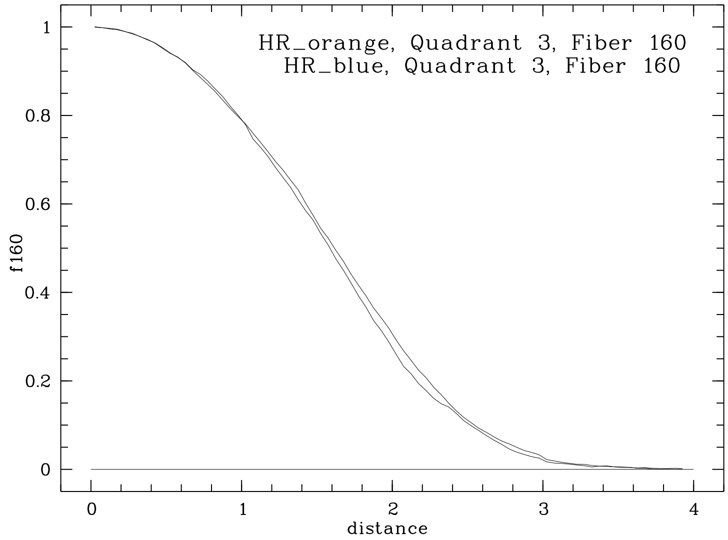


Figure 7.23.6: *The smoothed profiles obtained from Figure 7.23.5 are compared. The fiber profile from the HR_blue spectrum lays systematically below the fiber profile from the HR_orange spectrum, but the difference is* $< 2\%$.

In the current system, the same tabulated model profile is always used in the spectral extraction task (see Table 7.23.1). This profile was chosen as the median profile of a sample obtained from about 100 different fibers from the pseudo-slits of all quadrants.

Distance from centroid (pixel)	**Profile intensity**	**Distance from centroid (pixel)**	**Profile intensity**
0.025	1.00000	0.075	0.99996
0.125	0.99600	0.175	0.99363
0.225	0.99024	0.275	0.98371
0.325	0.97754	0.375	0.96946
0.425	0.95954	0.475	0.94931
0.525	0.93719	0.575	0.92335
0.625	0.90941	0.675	0.89560
0.725	0.87707	0.775	0.86130
0.825	0.84158	0.875	0.82178
0.925	0.80195	0.975	0.77927
1.025	0.76022	1.075	0.73276
1.125	0.70892	1.175	0.68236
1.225	0.65778	1.275	0.63209
1.325	0.60109	1.375	0.57527
1.425	0.54045	1.475	0.51107
1.525	0.48302	1.575	0.45345
1.625	0.42497	1.675	0.39587
1.725	0.36674	1.775	0.34252
1.825	0.31339	1.875	0.28750
1.925	0.26153	1.975	0.24057
2.025	0.21730	2.075	0.19604
2.125	0.17451	2.175	0.15626
2.225	0.13760	2.275	0.12041
2.325	0.10540	2.375	0.09260
2.425	0.07954	2.475	0.06847
2.525	0.05968	2.575	0.05272
2.625	0.04530	2.675	0.04062
2.725	0.03373	2.775	0.02871
2.825	0.02387	2.875	0.02167
2.925	0.01669	2.975	0.01503
3.025	0.01247	3.075	0.00993
3.125	0.00758	3.175	0.00669
3.225	0.00526	3.275	0.00301
3.325	0.00289	3.375	0.00123
3.425	0.00065		

Table 7.23.1: *Fiber profile model, normalised to its maximum intensity. A factor 3.0175 should be applied to the profile intensities to fulfill the relation* $\sum_i P(x_o + i) = 1$ *(see text).*

7.23.6 Aligning traces

In order to compensate for any difference introduced by instrument instabilities, the extraction mask, based on the fiber tracing $X_i(Y)$ made on a flat field exposure, should be aligned to the tracings made on the science exposure.

The alignment is possible if at least one scientific spectrum is traceable. At long exposure times (more than about 30 minutes) the sky emission itself becomes traceable, while a short exposure on a completely dark field may rightfully be rejected as invalid. Only in the case of a short exposure on a pure emission line object no spectrum may be traceable, making an accurate data reduction impossible with this method.

The flat field tracing solutions are traslated and rotated to a best match with the available science tracings. A single traced science spectra is sufficient to get alignment accuracies that are better than 0.1 pixels for all fibers. If however the instrument instabilities introduce signal displacements greater than 2 pixels, the scientific spectra will not be correctly identified, and the extraction mask alignment will be off by an integer number of fibers. The consequences might be disastrous, because the wrong tracings would be used to extract the scientific spectra, and the wrong relative transmission factors would be applied to them. In addition each scientific spectrum would be assigned to an offset position on the IFU head (see Figure 3.5.3, page 19), giving to the objects on the reconstructed field-of-view a typical zig-zagged appearance.

7.23.7 Spectral extraction

With a fiber profile model $P(x)$ and the (aligned) trace $X_i(Y)$ from each flat field fiber spectrum, it is now possible to extract all the scientific spectra. Let $S(X,Y)$ be the value of a pixel of coordinates (X,Y) in the science frame, and $F_i(Y)$ the (still unknown) total flux from the i-th fiber at the Y CCD coordinate. $S(X,Y)$ will be the sum of all contributions from all the fibers to the pixel (X,Y) (cross-talk):

$$S(X,Y) = \sum_i F_i(Y) \cdot P(X - X_i(Y))$$

In practice, it is known that the contribution from fibers that are far from the (X,Y) position can be neglected. Taking into consideration just the 3 closest fibers to the (X,Y) pixel we may write

$$S(X,Y) = F_{j-1}(Y) \cdot P(X - X_{j-1}(Y)) + F_j(Y) \cdot P(X - X_j(Y)) + F_{j+1}(Y) \cdot P(X - X_{j+1}(Y))$$

where j is the number of the fiber having the minimum difference $|X - X_i(Y)|$. With 400 spectra laying along the cross–dispersion direction, and with each spectrum about 5 pixels wide, the above formulation is *for each* Y a redundant system of 2000 equations in the 400 unknowns $F_i(Y)$. This system should be resolved for each Y pixel value (*i.e.*, more than 3000 times).

With such figures, this may turn out to be a computationally heavy method for the determination of the spectra $F_i(Y)$. Moreover, even if this extraction method would completely eliminate the effects of the cross-talk between fibers, we could not consider this as an *optimal* extraction.

Formulas like Robertson's (Robertson, J.G., 1986, PASP, 98, 1220), that are meant to optimally extract mutually contaminating nearby spectra, redefine the optimal weights used in Horne's extraction (Horne, K., 1986, PASP, 98, 609) to the practical effect of entirely rejecting the spectral signal that turns out to be too contaminated by the other spectrum. In the IFU case, where the contamination is overall and systematic, this implies the loss of a lot of signal that could instead be recovered with the solution of the linear system shown above.

At any rate, applying Robertson's formulation may turn out to be the only practically applicable choice. A first approximation of the spectral profile of the j-th fiber along the cross–dispersion direction must first be defined:

$$S_j(X,Y) = S(X,Y) - M_{j-1}(Y) \cdot \frac{P(X - X_{j-1}(Y))}{P(0)} - M_{j+1}(Y) \cdot \frac{P(X - X_{j+1}(Y))}{P(0)}$$

This approximation consists of subtracting from the observed profile $S(X,Y)$ the contributions from the two nearby spectra, here modeled with the PSF rescaled to their observed peak values $M_{j-1}(Y)$ and $M_{j+1}(Y)$. The following weights are defined:

$$W_j(X,Y) = \frac{S_j(X,Y)}{r^2 + gS(X,Y)} \cdot P(X - X_j(Y))$$

where r is the read-out-noise in electrons and g the gain factor in $e^-/$ADU. The optimally extracted spectrum (in electrons) would then be given by

$$F_j(Y) = g \sum_X W_j(X,Y) \cdot S_j(X,Y)$$

From the definition of the weights it is clear how abruptly they are set to zero as soon as the total signal $S(X,Y)$ is not balanced by the estimate of the single fiber profile, $S_j(X,Y)$.

In the current implementation of the extraction task used by the IFU pipeline recipes, the spectral flux is estimated from the values of the 3 pixels closest to the centroid $X_i(Y)$, normalised by the tabulated fiber profile model (see Table 7.23.1, page 160), and then simply averaged. It can be shown that, within a distance of 1.5 pixels, the cross-talk contamination between nearby fibers is always less than one part in a thousand (thanks to the arrangement of the fibers along the pseudo-slits, that never places fibers that are spatially far apart close to each other).

7.23.8 Wavelength calibration

The wavelength calibration is derived from an arc lamp exposure. The arc lamp spectra are extracted according to the procedure described in the previous section.

A rough optical distortion model would be initially used to unambiguously identify the brightest features of the extracted spectra. The search window for such features is wide, guaranteeing that even strong signal displacements would not prevent the identification of the reference lines. This prevents mistaking contaminations with the brightest arc lamp lines used as references. A rough wavelength calibration would then be used to search for the arc lines to be identified. In the particular case of LR observations, the expected positions of the zero order contamination on the CCD is determined, permitting to avoid regions that are too close to the zero order contamination. This search is required to return a number of lines at least twice the number of degrees of freedom of the fitting polynomial, and the model residuals should have an RMS always less than a specified threshold. If such requirements are not met, the whole result of the search is rejected, and the search is repeated anew, using wider and wider search windows - up to a certain limit. The first solutions found is then re-used as improved "first guesses", filling the gaps that are typically left behind after the first iteration.

The typical accuracy reached for the wavelength calibration is of about 0.2 pixels.

Once a wavelength λ is assigned to each Y pixel of each extracted spectrum, the positions of a number of predefined sky lines is determined on the extracted scientific spectra, and their median offset from their expected position is used to align the arc wavelength calibration to the scientific spectra. At this point the wavelength calibration is completed and the extracted scientific spectra are resampled in the wavelength space at a constant wavelength step.

7.23.9 Flat field correction

For IFU data, two alternative kinds of flat field correction can be considered:

The "classical" flat field correction, i.e., dividing the bias subtracted raw data by a map of the fixed-pattern-noise of the CCD, before further reduction steps are applied to the data. The fixed-pattern-noise map may be obtained by averaging several IFU flat field exposures, and removing the large scale trends (including the fibers pattern). This is technically difficult to obtain with the necessary accuracy, and it has not yet been done.

The extracted flat field correction, i.e., dividing the extracted scientific spectra by the extracted, normalised, transmission corrected, and large-scale-trend removed flat-lamp spectra. It should be noted that if this correction is applied, then the "classical" flat field correction (described at point 1) should not be applied. Moreover, dividing the extracted scientific spectra by the extracted flat-lamp spectra is just an approximate correction. The approximation is completely invalidated when we consider that the instrument flexures may have displaced the scientific spectra by a number of pixels with respect to the flat-lamp spectra. If the flat-lamp spectra were not extracted exactly from the same CCD regions as the scientific ones, the fixed-pattern noise would not be removed from the data, in fact it would be worsened[13].

In the current system, no flat fielding correction is applied to the scientific data. The recipe *vmifucalib* (see Section 5.24, page 101) produces an image of extracted flat field spectra that may be used for an approximate flat fielding correction (using any interactive data reduction system, as MIDAS or IRAF).

7.23.10 Transmission correction

The spectral extraction procedure (see Section 7.23.7, page 161) is applied to the flat field exposure itself. Assuming that the flat lamp uniformly illuminates the IFU head, it is straightforward to obtain the relative transmission factors corresponding to each extracted spectrum.

The wavelength calibrated flat field spectra are integrated along a fixed wavelength interval, chosen where the spectra are brighter, and away from possible zero order contaminations from multiplexed spectra (in case of LR grism observations). The obtained integrals are normalised to their median value. The normalised values are what is currently used for the fiber-to-fiber relative transmission correction applied to the scientific spectra after their extraction. This is valid, under the assumption that the absorption law maintains the same shape for all fibers.

[13] When a flat field correction is applied, a price is paid in terms of the increased variance of the processed signal, equal to the sum of the variances of the flat field and of the signal to be corrected; this price is only acceptable under the assumption that the removed fixed-pattern noise is greater than the noise added to the data by the flat fielding operation itself.

7.23.11 Sky subtraction

Sky subtraction is probably the toughest problem in the VIMOS IFU data reduction process, because it will not be uncommon that the observed objects completely fill the IFU field. The only possibility is to select from all the reduced spectra the ones with the lowest signal, and classify them as sky spectra. Assuming that the transmission correction and the wavelength calibration had already been applied, their median spectrum would simply be subtracted from all the other extracted spectra. It is clear though that this way of proceeding is extremely risky. In the current implementation no sky subtraction is applied to the reduced data (with the exception of standard star spectra reduction, see recipe *vmifuscience*, Section 5.25, page 107).

7.24 vmifucalib

In this section the basic steps of the data reduction procedure applied by the recipe *vmifucalib* (see Section 5.24, page 101) are described. Please refer to Section 7.23, page 152, for more details about the basic operations.

The input flat field and arc lamp exposures are processed in the following way:

1. If more than one, the flat field exposures are combined with a specified stacking method, and the master bias is removed from the result. The master bias is removed also from the arc lamp exposure.
2. The flat field spectra are identified and traced as described in Section 7.23.3, page 155. This operation is performed for each illuminated IFU pseudo-slit (just one pseudo-slit in the case of HR and MR grism data, and four pseudo-slits in the case of LR grism data).
3. The obtained traces are fit with a 4th degree polynomial in the case of HR and MR grism data, or with a 3rd degree polynomial in the case of LR data. The accuracy reached is better than 0.04 pixels.
4. The tracing polynomial models are used to extract the flat field spectra and the arc lamp spectra. The extraction is based on the values of the three pixels that are closer to the trace of each fiber (see Section 7.23.7, page 161).
5. The wavelength calibration is obtained for each extracted arc lamp spectra (see Section 7.23.8, page 162). A 4th degree polynomial is used for relating wavelengths to CCD positions for HR and MR grism data, while a 3rd degree polynomial is used for LR grism data. The obtained accuracy is better than 0.3 pixels.
6. The extracted flat field spectra are resampled at a constant wavelength step (applying a flux conservation correction, and slightly oversampling the signal so that the wavelength step is a bit smaller than a CCD pixel), and used to determine the fiber-to-fiber relative transmission correction (see Section 7.23.10, page 163).

7.25 vmifuscience

In this section the basic steps of the data reduction procedure applied by the recipe *vmifuscience* (see Section 5.25, page 107, are described. Please refer to Section 7.23, page 152, for more details about the basic operations.

This recipe receives a science exposure, a master bias, and all the calibrations produced by the recipe *vmifucalib*, processing them in the following way:

1. The master bias is removed from the science exposure.
2. The science exposure is analysed, in order to locate traceable spectra. This operation doesn't identify the detected spectra — it just determines their positions on the CCD.
3. If no traceable spectra are found, go to step 6.
4. The detected scientific spectra are traced, and the traces are modeled with a polynomial shape (see Section 7.23.3, page 155).
5. The extraction mask obtained from the flat field is aligned to the available tracing solutions on the science (see Section 7.23.6, page 161).
6. The science spectra are extracted along the (possibly modified) flat field tracings (see Section 7.23.7, page 161).
7. The wavelength calibrations for each fiber are used to determine the offset of a number of chosen sky lines from their expected positions on the CCD. The wavelength calibrations polynomials are corrected according to this offset. Currently, just the sky-lines at 5577.388, 6300.304, 6363.780, and 8344.602 Å are used.
8. The science spectra are resampled at a constant wavelength step (applying a flux conservation correction, and slightly oversampling the signal so that the wavelength step is a bit smaller than a CCD pixel).
9. The scientific spectra are divided by the relative transmission correction factors obtained with recipe *vmifucalib*.
10. The calibrated science spectra are integrated along a predefined wavelength range, chosen where the spectra are brighter, and away from possible zero order contaminations from multiplexed spectra. The obtained values are used in the reconstruction of the IFU field-of-view. The reconstructed field-of-view fills the region of an 80x80 image that corresponds to the reduced quadrant. In this way, the reconstructed image from 4 reduced quadrants can be easily obtained by the sum of the reconstructed images from different quadrants (see recipe *vmifucombine*, Section 5.27, page 110).

7.26 vmifustandard

The data reduction steps applied by this recipe are the same as those applied by the *vmifuscience* recipe. The only extra operations are: an evaluation of the sky spectrum (see Section 7.23.11, page 164), and the determination of the total standard star spectrum from all the single fiber spectra.

A Installation

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

A bundled version of the VIMOS pipeline with all the required tools and an installer script is available from `ftp://ftp.eso.org/pub/cpl/vimos`, for users who are not familiar with the installation of software packages.

A.1 Supported platforms

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

- HP-UX 11.00 or later,
- Linux (glibc 2.1 or later),
- Sun Solaris 2.8 or later,

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

A.2 Building the VIMOS pipeline

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

A.2.1 Requirements

To compile and install the VIMOS pipeline one needs:

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

An installation of the Common Pipeline library (CPL) must also be available on the system. Currently the CPL version 1.0.1 or newer is required. The CPL distribution can be obtained from `http://www.eso.org/cpl`.

Please note that CPL itself depends on an existing qfits installation. The qfits sources are available from the CPL download page or directly from the qfits homepage at `http://www.eso.org/projects/aot/qfits`. In conjunction with CPL 1.0.1, qfits 4.3.5 must be used.

Important Note: Qfits contains a fixed sized pointer buffer which needs to be enlarged for the VIMOS recipes. For qfits 4.3.5 the only way to adjust the buffer size is to edit the file `config.make` which will be created when the qfits `configure` program in the distributions top-level directory is executed.

Edit this file and remove all xmemory related entries (entries starting with '-DXMEMORY') from the line starting with 'CFLAGS'. Then add `-DXMEMORY_MAXPTRS=32768` to the end of this line and continue with the qfits installation.

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

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

A.2.2 Downloading the VIMOS pipeline source distribution

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

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

A.2.3 Compiling and installing the VIMOS pipeline

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

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

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

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

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

3. Run the configuration script `configure` which is located in the top-level directory of the VIMOS pipeline source tree.

If the CPL or qfits have not been installed into one of the system's standard directories, the configuration script must be told where the dependent libraries (CPL, qfits, etc.) can be found. This is either done using the command-line options `--with-cpl`, `--with-cext` and `--with-qfits` of the script or by defining the two environment variables `CPLDIR` and `QFITSDIR`.

As an example, if the CPL has been installed into `/somewhere`, *i.e.*, the CPL header files are located in `/somewhere/include` and the libraries are in `/somewhere/lib`. In this case the argument of the option `--with-cpl` or the path assigned to the variable `CPLDIR` must be `/somewhere`, *i.e.*, the root directory of the CPL installation. The same is true for the option or environment variable corresponding to qfits.

If command-line options are used, the argument of `--with-cext` is usually the same as the one passed to `--with-cpl`, since both packages install together. This is also reflected by the fact that there is no separate environment variable for the `cext` package.

By default the configuration script will setup the installation path of the package as `/usr/local`. This means that root privileges would be required to complete the installation. If one does not have sufficient privileges, or the default installation path is just not appropriate, it is possible to change this default. There are two possibilities: using the script's `--prefix` option, or defining the environment variable `PIPE_HOME`. Using the option will override a possibly defined variable.

In either case the provided path is used for the installation of the VIMOS pipeline. All its components will be located in the directory tree rooted at this path.

The following example assumes that the variables `CPLDIR` and `QFITSDIR` are properly set or the CPL and qfits have been installed into one of the system's standard directories. This should keep the following example commands as simple as possible.

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

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

```
./configure --prefix=/usr/local/pipeline
```

Alternatively, if using the environment variable is preferred, `PIPE_HOME` may be defined in the shell's startup file. If the *Bourne* or a compatible shell (i.e. *sh*, *bash*, *ksh*, *zsh*, etc.) is used, to the file `.profile` (or `.bashrc` if `bash` is used) it should be added:

```
PIPE_HOME=/usr/local/pipeline
export PIPE_HOME
```

If the the C-shell (i.e. *csh* or *tcsh*) is used the commands above translate into:

```
setenv PIPE_HOME /usr/local/pipeline
```

and should be added to the C-shell startup file `.cshrc`.

To activate these settings one may either logout and login again, or source the startup script manually.

4. To compile and install the VIMOS pipeline on the system, run the following sequence of commands from the source-tree's top-level directory:

```
$ make
$ make install
```

The `configure` script provides a variety of command-line options to customise the VIMOS pipeline installation. The list of available options can be obtained by running `./configure --help`. Using a command line option always takes precedence over any previously set environment variable.

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

A.3 Configuring the pipeline recipe front-end applications

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

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

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

A.3.1 Setting up the EsoRex command-line tool

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

```
esorex [esorex_options] recipe_name [recipe_options] set_of_frames
```

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

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

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

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

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

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

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

```
esorex --recipes
```

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

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

A.3.2 Setting up Gasgano

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