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 Manual

VLT-MAN-ESO-19500-3355

Issue 4.1.12

Date Jan 30th 2025

	Name	Date	Signature
Prepared:	ESO VIMOS Pipeline Team	Jan 30th 2025	
Approved:	P. Ballester		
Released:	M. Peron		

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
2.0	30/09/2005	All	Second version
3.0	30/01/2007	All	Third version
4.0	06/06/2008	All	Fourth version
5.0	07/07/2009	All	No real changes with respect to previous version
6.0	10/10/2010	All	Added documentation on new MOS pipeline
6.1	11/11/2010	All	Documentation bugs fixed
6.2	03/03/2011	All	Further documentation bugs fixed
6.3	13/09/2011	All	Further documentation bugs fixed
6.4	22/05/2012	All	Further documentation bugs fixed
6.5	30/10/2012	6, 8	Updated IFU cubes documentation
6.6	30/10/2012	2	Include info about Reflex
6.7	01/03/2012	All	Update to the latest software versions
6.8	01/04/2013	All	–photometry parameter removed
6.9	01/11/2013	All	–qc and –check parameters removed
6.10	01/04/2014	All	Changes for version 2.9.15
7.0	01/04/2016	All	Changes for version 3.1.3
7.3	14/05/2018	All	Changes for version 3.2.4
7.4	03/05/2019	All	Changes for version 3.3.0
8.1	20/05/2021	All	Document color transformation details
			Changes for version 4.1.6
8.2	19/05/2022	All	Changes for version 4.1.7
4.1.8	19/05/2023	None	Changes for version 4.1.8
4.1.10	15/05/2024	None	Version number aligned to pipeline
4.1.12	30/01/2025	None	CDS URL updated for downloads

This page was intentionally left blank

Contents

1 Introduction **13**

1.1 Purpose . . . 13

1.2 Acknowledgements . . . 13

1.3 Reference documents . . . 14

2 Overview **16**

3 The VIMOS instrument **17**

3.1 Overview . . . 17

3.2 Direct imaging . . . 17

3.3 Multi-Object-Spectroscopy (MOS) . . . 18

3.4 Integral Field Unit (IFU) . . . 19

3.5 IFU components numbering scheme . . . 21

4 Quick start **23**

4.1 VIMOS pipeline recipes . . . 23

4.2 An introduction to Reflex, Gasgano and EsoRex . . . 25

4.2.1 Using Reflex . . . 25

4.2.2 Using Gasgano . . . 26

4.2.3 Using EsoRex . . . 33

4.3 Example of MOS data reduction using the current recipes . . . 36

4.4 Example of IFU data reduction . . . 42

4.5 Known problems . . . 45

5 VIMOS data **46**

5.1 Raw frames . . . 46

5.1.1 General frames . . . 46

5.1.2 Direct imaging frames . . . 48

5.1.3 MOS frames . . . 51

5.1.4 IFU frames . . . 53

5.2 Product frames . . . 55

5.2.1 General frames . . . 55
5.2.2 Direct imaging frames . . . 56
5.2.3 MOS frames . . . 59
5.2.4 IFU frames . . . 63

6 Static Calibration Data **67**
6.1 Detector bad pixel table . . . 67
6.2 Photometric standard stars catalog . . . 67
6.3 Grism config table . . . 68
6.4 Arc lamp lines catalog . . . 68
6.5 Master distortion table . . . 69
6.6 Atmospheric extinction table . . . 69
6.7 Standard star flux table . . . 70
6.8 Telluric contamination table . . . 70
6.9 IFU fiber identification . . . 70

7 Pipeline Recipes Interfaces **71**
7.1 vimos_bias_combine . . . 71
7.1.1 Description . . . 71
7.1.2 Input frames . . . 71
7.1.3 Recipe parameters . . . 71
7.1.4 Product frames . . . 73
7.1.5 Quality control parameters . . . 74
7.2 vimos_dark_combine . . . 74
7.2.1 Description . . . 74
7.2.2 Input frames . . . 74
7.2.3 Recipe parameters . . . 74
7.2.4 Product frames . . . 76
7.2.5 Quality control parameters . . . 77
7.3 vimos_twilight_flat_combine . . . 77
7.3.1 Description . . . 77

7.3.2 Input frames . . . 77
7.3.3 Recipe parameters . . . 78
7.3.4 Product frames . . . 80
7.3.5 Quality control parameters . . . 81
7.4 vimos_detector_noise . . . 81
7.4.1 Description . . . 81
7.4.2 Input frames . . . 81
7.4.3 Recipe parameters . . . 82
7.4.4 Product frames . . . 82
7.4.5 Quality control parameters . . . 82
7.5 vimos_fringe_combine . . . 82
7.5.1 Description . . . 82
7.5.2 Input frames . . . 83
7.5.3 Recipe parameters . . . 84
7.5.4 Product frames . . . 85
7.5.5 Quality control parameters . . . 86
7.6 vimos_standard_process . . . 86
7.6.1 Description . . . 86
7.6.2 Input frames . . . 86
7.6.3 Recipe parameters . . . 86
7.6.4 Product frames . . . 89
7.6.5 Quality control parameters . . . 90
7.7 vimos_science_process . . . 91
7.7.1 Description . . . 91
7.7.2 Input frames . . . 91
7.7.3 Recipe parameters . . . 92
7.7.4 Product frames . . . 95
7.7.5 Quality control parameters . . . 97
7.8 vmdet . . . 99
7.9 vmbias . . . 101

7.10 vmdark . . . 104
7.11 vmmoscalib . . . 106
7.11.1 Input files . . . 107
7.11.2 Output files . . . 107
7.11.3 Configuration parameters . . . 116
7.11.4 Quality control parameters . . . 122
7.12 vmmosscience . . . 123
7.12.1 Input files . . . 124
7.12.2 Output files . . . 127
7.12.3 Configuration parameters . . . 133
7.12.4 Quality control parameters . . . 138
7.13 vmifucalib . . . 140
7.14 vmifuscience . . . 145
7.15 vmifustandard . . . 147
7.16 vmifucombine . . . 149
7.17 vmifucombinecube . . . 149
7.18 vmspphot . . . 149

8 Geometrical distortions models 151
8.1 Polynomial models . . . 151
8.2 Optical distortions . . . 153
8.2.1 CCD to Mask transformation and its inverse . . . 153
8.2.2 CCD to Sky distortion and its inverse . . . 154
8.3 Spectral distortions . . . 156
8.3.1 Global distortion table . . . 156
8.3.2 Zero order contamination model (obsolete) . . . 158
8.3.3 Optical distortion model (obsolete) . . . 158
8.3.4 Spatial curvature model (obsolete) . . . 159
8.3.5 Inverse dispersion solution (obsolete) . . . 160
8.3.6 Slit spectra extraction (obsolete) . . . 161

9 Algorithms **162**

9.1 Stacking . . . 162

9.2 Variance maps . . . 162

9.3 Confidence maps . . . 162

9.4 WCS fitting . . . 163

9.5 Detector readnoise and gain . . . 163

9.6 Object detection/catalogue generation . . . 163

9.6.1 Background analysis and object detection . . . 164

9.6.2 Object parameterisation . . . 164

9.6.3 Morphological classification . . . 166

9.7 Defringing . . . 166

9.8 Bad pixel cleaning for spectroscopy . . . 167

9.9 Cosmic rays removal for spectroscopy . . . 168

9.10 Bias subtraction for spectroscopy . . . 169

9.11 Dark subtraction for spectroscopy . . . 169

9.11.1 Flat normalisation for spectroscopy . . . 170

9.12 Frame combination for spectroscopy . . . 170

9.13 Blind arc lamp lines identification . . . 171

9.14 Sky fringing correction for spectroscopy . . . 172

9.15 Computation of the spectral efficiency and response . . . 173

9.16 Spectrophotometry flux calibration . . . 175

9.16.1 Flux calibration for holographic grisms . . . 176

9.17 vmdet . . . 176

9.17.1 Read-out-noise determination for spectroscopy . . . 177

9.17.2 Bias subtraction for spectroscopy . . . 177

9.17.3 Photon transfer curve and gain factor determination . . . 177

9.17.4 Bad pixels identification for spectroscopy . . . 178

9.18 vmbias . . . 178

9.19 vmdark . . . 178

9.20 vmmoscalib . . . 179

9.20.1 1D peak-detection . . . 181
9.20.2 1D pattern-recognition . . . 182
9.20.3 Determination of the spectral range . . . 185
9.20.4 Choice of a reference wavelength . . . 185
9.20.5 Position of the reference wavelength on the CCD . . . 185
9.20.6 2D pattern-recognition . . . 186
9.20.7 Optical distortion model determination . . . 188
9.20.8 Tracing slit spectra edges . . . 188
9.20.9 Spatial curvature model determination . . . 188
9.20.10 Extraction of slit spectra . . . 188
9.20.11 Line catalogs and reference spectra . . . 189
9.21 vmmosscience . . . 201
9.21.1 Align wavelength solution to sky lines positions . . . 202
9.21.2 Slit spectra extraction . . . 202
9.21.3 Object detection . . . 202
9.21.4 Sky modeling . . . 202
9.21.5 Object extraction . . . 202
9.21.6 Align and combine the processed images . . . 203
9.22 Overview of the VIMOS IFU data reduction procedure . . . 203
9.22.1 Required data . . . 203
9.22.2 Fibers identification . . . 204
9.22.3 Tracing spectra . . . 205
9.22.4 Determination of the inverse dispersion solution . . . 207
9.22.5 Background subtraction . . . 208
9.22.6 Determination of the fiber profiles . . . 210
9.22.7 Aligning traces . . . 213
9.22.8 Spectral extraction . . . 215
9.22.9 Wavelength calibration . . . 216
9.22.10 Flat field correction . . . 216
9.22.11 Transmission correction . . . 217

9.22.12 Sky subtraction . . . 217
9.23 IFU WCS information . . . 217
9.24 vmifucalib . . . 218
9.25 vmifuscience . . . 218
9.26 vmifustandard . . . 219

A Troubleshooting Guide **220**
A.1 Checking the results of recipe vmmoscalib . . . 220
A.1.1 Were all spectra detected and properly traced? . . . 220
A.1.2 Were all spectra properly calibrated in wavelength? . . . 220
A.1.3 Is the spectral resolution as expected? . . . 223
A.2 Fixing pattern-recognition failures in vmmoscalib . . . 224
A.2.1 Some arc lamp reference lines are missing . . . 224
A.2.2 Some arc lamp reference lines are very faint . . . 224
A.2.3 The reference lines in the arc lamp exposure are very broad . . . 224
A.2.4 The spectral dispersion is not what expected . . . 225
A.2.5 There are spectra at very large offsets . . . 225
A.3 Fixing other possible failures in vmmoscalib . . . 225
A.3.1 The spectra are too tightly packed . . . 225
A.3.2 The wavelength calibration residuals display systematic trends . . . 226
A.3.3 The calibrated spectra look "noisy" at their ends . . . 226
A.3.4 The flat field is not properly normalised . . . 226
A.3.5 Valid reference lines are rejected . . . 226
A.4 Checking the results of recipes vmmosscience . . . 226
A.4.1 Were all spectra properly wavelength calibrated? . . . 227
A.4.2 Is the sky background properly subtracted? . . . 228
A.4.3 Were all objects detected? . . . 229
A.4.4 Were all the detected objects properly extracted? . . . 229
A.5 Fixing failures in vmmosscience . . . 229
A.5.1 The wavelength calibration is bad . . . 229
A.5.2 The sky alignment of the wavelength solution failed . . . 230

A.5.3 The sky subtraction failed for resolved sources 230
A.5.4 The sky subtraction failed for curved or tilted slits 230
A.5.5 Cosmic rays are not removed 230
A.5.6 The sampling of the remapped scientific spectra is poor 231
A.5.7 The extracted spectra are normalised in time 231
A.5.8 There are often spurious objects detected at the slit edges 231
A.5.9 Some "obvious" objects are not detected 231

B Software Installation **232**
B.1 System Requirements 232
B.2 Installing VIMOS pipeline via `macports` 232
B.3 Installing VIMOS pipeline via `rpm/yum/dnf` 232
B.4 Installing VIMOS pipeline via `install_esoreflex` 233

C Abbreviations and acronyms **235**

1 Introduction

1.1 Purpose

This manual is a complete description of the data reduction recipes used by the the VIMOS pipeline, reflecting the status of the VIMOS pipeline as of Jan 30th 2025(version 4.1.12).

The VIMOS pipeline is a subsystem of the *VLT Data Flow System* (DFS). It is used in three different environments. First, it is used within ESO in *Data Flow Operations* (DFO) for the validation of scientific exposures, data quality control and the generation of master calibration data. Second, the *Paranal Science Operations* (PSO) uses it for quick-look assessment of data and the reduction of scientific exposures. Finally, the VIMOS pipeline recipes are made public to the user community, to allow a more fine-tuned processing of the data.

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

More detailed descriptions of the data reduction algorithms used by the individual pipeline recipes can be found in Sections 8 and 9.

In Appendix C provides a list of used abbreviations and acronyms.

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 (older) MOS pipeline recipes.

Starting with the pipeline release 2.5.0, two new MOS recipes have been added: they have been entirely developed by ESO, and they are intended to replace the older set of MOS recipes. Among several improvements they are primarily implementing a different calibration approach based on pattern recognition, which was already applied successfully on the FORS1/2 and EFOSC2 pipelines. This permits to greatly reduce the workload on software maintenance, not requiring any preliminary optical and spectral modeling of the instrument in order to work. Using the new recipes is mandatory for reducing data obtained after the VIMOS CCD mosaic upgrade (Summer 2010), and they can be used for reducing older data as well.

The MOS reduction pipeline has been extensively improved and throughly reviewd thanks to the work of Sabine Moehler (ESO Data Management & Operations Department) in the course of the years 2013-2015.

Valuable suggestions on the ESO 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). Further improvements were later provided by Dr Peter M. Weilbacher (Astrophysikalisches Institut Potsdam), and Dr Katrina Exter (Space Telescope Science Institute).

The feedback we received in numerous discussions with our “beta-testers”, Paola Popesso (ESO Office of the Director General), Piero Rosati and Martino Romaniello (ESO Data Management and Operations 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) and Stefano Cristiani (INAF – Osservatorio Astronomico di Trieste).

Gianni Marconi, Stephane Brillant, and Stefano Bagnulo (ESO Paranal Observatory) have been unvaluable for the good collaboration and the constant support in the first VIMOS years.

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

1.3 Reference documents

[1] *VIMOS Data Reduction Cookbook*. VLT-MAN-ESO-13200-4033. 16

[2] ESO/ASG, *http://www.eso.org/sci/observing/phase3/p3sdpstd.pdf*. *ESO Science Data Products Standard*, 5th edition, January 2013. GEN-SPE-ESO-33000-5335. 133

[3] ESO/SDD/DFS, *http://www.eso.org/cpl/esorex.html*. *ESOREX home page*. 16

[4] ESO/SDD/DFS, *http://www.eso.org/gasgano/*. *Gasgano User's Manual*. VLT-PRO-ESO-19000-1932. 16, 28, 55

[5] M. J. Irwin. Automatic analysis of crowded fields. *MNRAS*, 214:575–604, June 1985. 164

[6] Kepler Project, *http://code.kepler-project.org/code/kepler-docs/trunk/outreach/documentation/shipping/getting-started-guide.pdf*. *Kepler Getting Started Guide*. 25

[7] Forchì V. *Reflex User's Manual*. ESO/SDD/DFS, *http://www.eso.org/gasgano/*, 0.7 edition, 2012. VLT-MAN-ESO-19000-5037. 16

2 Overview

In collaboration with instrument consortia, the Science Operation Software Department (SOSD) of the Directorate of Engineering is implementing data reduction pipelines for the most commonly used VLT/VLTI instrument modes. These data reduction pipelines have the following three main purposes:

Data quality control: pipelines are used to produce the quantitative information necessary to monitor instrument performance.

Master calibration product creation: pipelines are used to produce master calibration products (e.g., combined bias frames, super-flats, wavelength dispersion solutions).

Science product creation: using pipeline-generated master calibration products, science products are produced for the supported instrument modes (e.g., optimally extracted spectra, bias-corrected and flat-fielded images, wavelength-calibrated spectra). The accuracy of the science products is limited by the quality of the available master calibration products and by the algorithmic implementation of the pipelines themselves. In particular, adopted automatic reduction strategies may not be suitable or optimal for all scientific goals.

Instrument pipelines consist of a set of data processing modules that can be called from opportune front-end applications, such as the automatic data management tools available on Paranal.

ESO offers three front-end applications for launching pipeline recipes, *Gasgano* [4] *Esorex* [3] and *Reflex* [7] These applications can be downloaded separately from the ESO web pages (see [4], [3], [7]). An illustrated introduction to Gasgano is provided in the "Quick Start" Section of this manual (see page 23).

The VIMOS, pipeline comes with three Reflex workflows:

- IMG workflow. This should be used for imaging data. It calibrates the images and produces stacks of jittered images.
- MOS workflow. This is used for the MOS mode. This workflow has been extensively reviewed.
- IFU workflow. This should be used for IFU data. It is a basic worflow provided for convinience.

In order to use each of these workflows, please refer to the corresponding Reflex tutorial that can be downloaded from the pipelines page (*http://www.eso.org/sci/software/pipelines/*.

The VIMOS, instrument and the different types of raw frames and auxiliary data are described in Sections 3 and 5.

A brief introduction to the usage of the available reduction recipes using *Gasgano* or *Esorex* is presented in Section 4. It is strongly suggested to read also the Troubleshooting Guide in the Appendix (page 220), and the pipeline related sections in the VIMOS, Data Reduction Cookbook [1], which go even deeper into that.

More details on what are inputs, products, quality control measured quantities, and controlling parameters of each recipe are given in Section 7.

More detailed descriptions of the data reduction algorithms used by the individual pipeline recipes can be found in Section 9.

In Appendix C a list of used abbreviations and acronyms is given.

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 1st, 2003.

A new set of four holographic HR_red grisms was installed in VIMOS on October 5, 2005. Compared to the previous set, the efficiency increased by about 50%.

In the context of the VIMOS Improvement Project, on August 7, 2010 VIMOS was back on sky equipped with a new CCD mosaic with higher red sensitivity (a factor of two for wavelengths longer than 8000 Å) and less interference fringing (not detectable on LR_red grism spectra, and about 2% peak-to-valley on HR_red grism spectra). Also a new holographic HR_blue grism was installed.

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. After August 7, 2010, the mosaic was replaced by a e2v CCD 44-82-1-D42, deep depletion, backside illuminated, double layer coating chips.[1].

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/sci/facilities/paranal/instruments/vimos/inst/imaging.html.

The colour transformations used for the conversion between the VIMOS-IMG photometric system and the system used for photometric calibration (e.g. APASS) can be found in Appendix A of the VIMOS User Man-

[1]See `http://www.eso.org/sci/facilities/paranal/instruments/vimos/` for detector design and performance reports of the four VIMOS CCD systems

ual, or in the on-line F.A.Q: *http://www.eso.org/sci/data-processing/faq/are-there-any-colour-transformations-available-for-the-vimos-imaging-filters.html*

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".

There are 6 grisms available, all operating in first order. Their spectral characteristics are given in Table 3.1.

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	GG475	7000	4800 - 10000	580 - 720	2.5
MR	OS_red	7000	5500 - 9700	580 - 720	2.5
HR_red	GG475	7400	6500 - 8650	2500 - 3100	0.6
HR_orange	GG435	6310	5150 - 7600	2150 - 2650	0.6
HR_blue	none	5100	4100 - 6300	2050 - 2550	0.5
HR_blue_holog	none	5100	4100 - 6300	2050 - 2550	0.5

Table 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 transmission curves for the four grism/filter units are available at* `http://www.eso.org/sci/facilities/paranal/instruments/vimos/`.

With LR grisms, a spectrum will typically span less than 600 pixels along the dispersion direction. This allows a spectral multiplexing factor up to 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 GG475 filter. It is therefore possible to stack up to 2 spectra along the dispersion direction, provided that half of 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 only spatial multiplexing is possible. 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 0th, -1st and 2nd 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 -1st order spectrum at the same resolution of the 1st 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 -1st contamination on figure 3.1, page 19). For this reason spectrally 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 0th, 2nd and -1st contaminations would then be removed by the sky subtraction procedure.

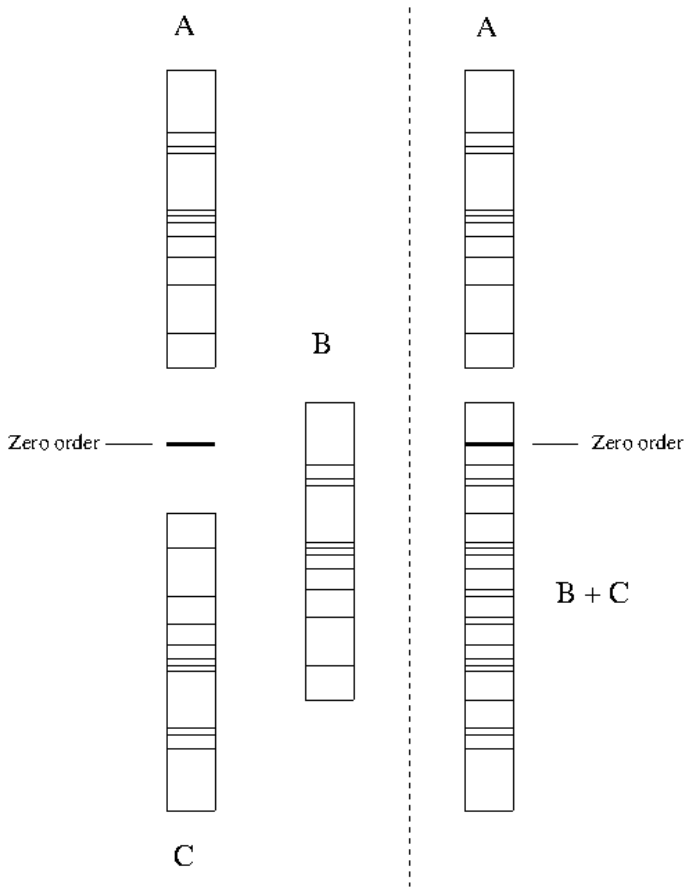

Figure 3.1: *Contamination from orders* 0 *and* -1 *in multiplexed spectra. On the left, the first order slit spectra A and B are shown, together with the* 0 *and the* -1 *orders of spectrum A. If spectra A and B are multiplexed, as shown on the right, spectrum B is contaminated by the* 0 *and* -1 *orders of spectrum A..*

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 3.3 and 3.4, pages 21–22).

At medium and high resolution just the 4 central bundles on the IFU head are illuminated (see Figure 3.4, page 22). 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 3.1). 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 common spectral range is reduced by about 160 Å leading to Table 3.2.

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

Table 3.2: *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 3.2).

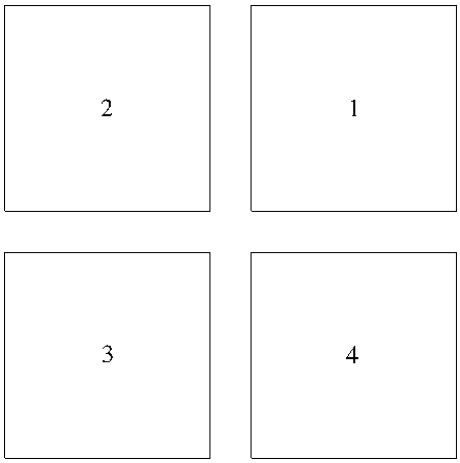

Figure 3.2: *Counting VIMOS quadrants.*

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 3.3).

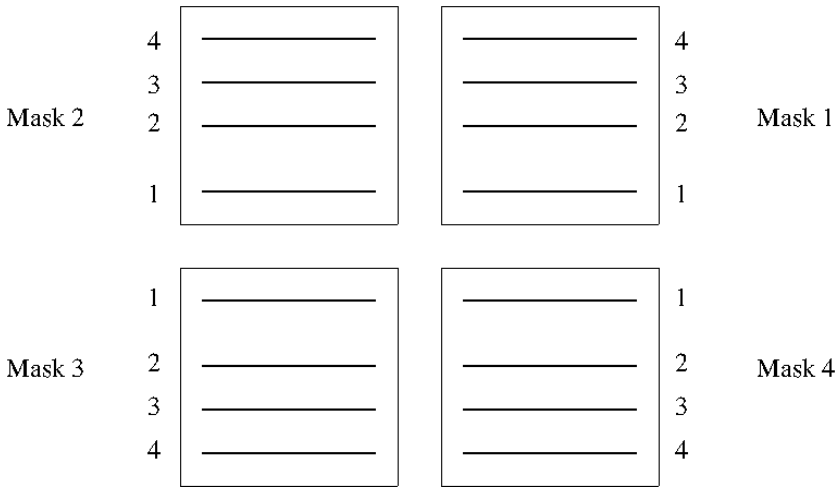

Figure 3.3: *Counting IFU pseudo-slits.*

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 3.4).

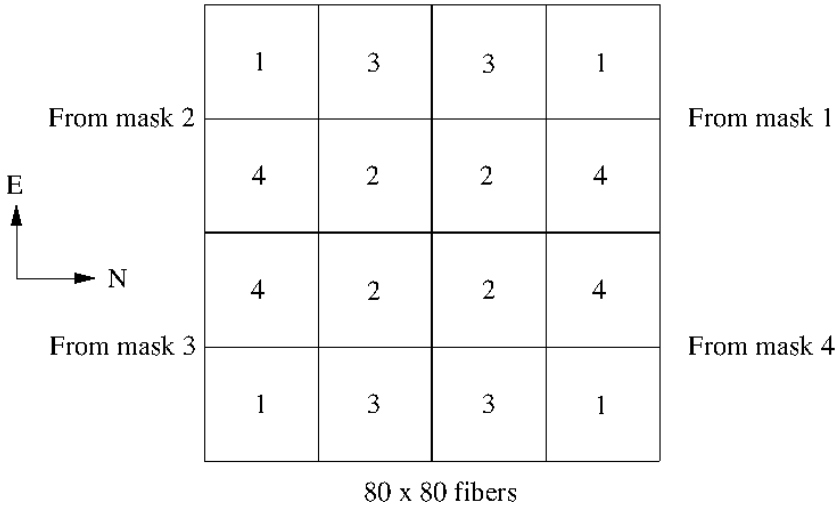

Figure 3.4: *IFU head. The number of the corresponding pseudo-slit is indicated within each 20x20 fiber module IFU pseudo-slits.*

North is to the right, and East is up. The exact spatial position for each individual fiber is listed in the IFU tables (see Table 7.17, page 141).

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 Quick start

In this section the most immediate usage of the VIMOS pipeline recipes is described. In particular, typical data reduction sessions for each instrument operating mode are presented in following sections. Note that for imaging mode it is recommended to consult the Reflex VIMOS imaging tutorial instead.

4.1 VIMOS pipeline recipes

The current VIMOS pipeline is based on a set of 25 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:

vimos_ima_bias: Combine a list of bias frames into a mean bias frame. Optionally compare the output frame to a reference bias frame.

vimos_ima_dark: Combine a list of dark frames into a mean dark frame. Optionally compare the output frame to a reference dark frame.

vimos_ima_det_noise: Compute the readnoise and gain for the vimos detectors using two bias frames and two flat field frames.

vimos_ima_fringe: Take a list of science frames and correct them to remove bias, dark and flat field signatures. Combine the images to form a new master fringe frame.

vimos_ima_twilight_flat: Combine a list of twilight flat frames into a mean frame. Optionally compare the output frame to a reference twilight flat frame

Direct imaging flux calibration:

vimos_ima_standard: Take a list of standard frames and correct them to remove bias, dark and flat field signatures. Optionally defringe the reddest filter images. Astrometrically and photometrically calibrate the individual images, but do no stacking.

Direct imaging data reduction:

vimos_ima_science: Take a list of science frames and correct them to remove bias, dark and flat field signatures. Optionally defringe the reddest filter images. Stack jitter sequences. Astrometrically and photometrically calibrate the stacks.

Creation of MOS calibration data:

vmmoscalib: computing the optical distortion, the spatial curvature, and the inverse dispersion models. Producing a spectral master flat field.

MOS data reduction:

vmmosscience: reducing a scientific exposure, or a set of (dithered) scientific exposures. In case of standard star exposures, also computing the instrument spectral efficiency and response curves.

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, and computing the instrument spectral efficiency and response curves from a standard star exposure.

IFU data reduction:

vmifuscience: reducing a scientific exposure.

vmspphot: applying flux calibration to already reduced spectra.

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.

4.2 An introduction to Reflex, Gasgano and EsoRex

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

An instrument pipeline consists of a set of data processing modules that can be called from different host applications, namely:

- *Reflex* is a graphical tool that helps the user to execute data reduction workflows which contain several recipes. This dramatically decreases the time the user needs to run a whole reduction chain, from calibration and raw data down to the final products. *Reflex* takes care of grouping the different data sets, associating the calibration frames and managing the interdependencies between recipes in the calibration cascade. **Reflex is the recommended software tool for reducing your data.**
- *Gasgano* is an alternative data management tool that simplifies the data organization process. In addition, *Gasgano* allows the user to execute directly the pipeline recipes on a set of selected files.
- *EsoRex* is a command line tool used to run the pipeline recipes. *EsoRex* commands can be easily scripted.
- The Paranal observatory implements automatic data management tools that trigger the execution of pipeline recipes. This aspect is not covered in this manual.

4.2.1 Using Reflex

Reflex is the recommended tool to reduce complete data sets that include all the calibration frames. It is an advanced tool, and yet easy to use, that is geared towards maximum scientific return. It is based on the workflow engine *Kepler* [6].

Currently there is one Reflex workflow which supports VIMOS MOS data and one which supports IFU data. In the future more observing modes will be supported. This manual does not cover the installation of *Reflex* . Please refer to the VIMOS Reflex tutorials and Reflex manual for the installation procedure which also contains a detailed description of the *Reflex* application. The latest documentation is available at *http://www.eso.org/sci/software/pipeline* What follows is a very brief summary of it.

Once installed, *Reflex* can be executed with the command:

```
user@host# reflex &
```

Reflex main concepts are workflows and actors. Workflows are canvasses which show the interdependence of the pipeline recipes, allowing the user to easily obtain an overview of the reduction steps. Workflows have the advantage of requiring a small learning curve in order to get the pipeline running.

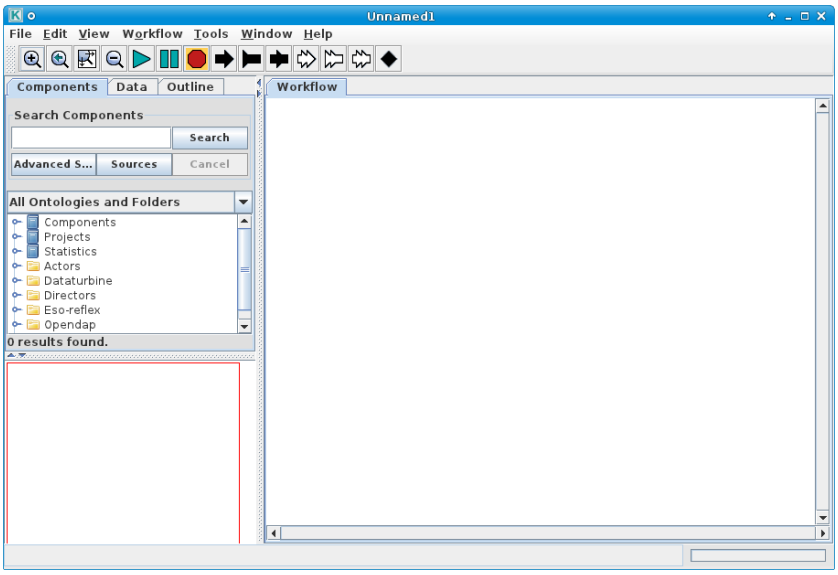

Figure 4.1: *Fresh Reflex canvas.*

Actors are the entities which actually perform some kind of operation. In *Reflex*, to each main actors correspond the pipeline recipes themselves, which perform the data reduction steps, but there are other actors such as the DataOrganizer, or the FitsRouter that are useful to manage the data files. Each actor can be configured by right-clicking on it and selecting *Configure Actor* as shown in Figure 4.2. In the case of the recipe actors, the recipe parameters are part of the actor and make up the second group of parameters.

In addition to those elements, the workflow contains variables that contain the most important settings, such as the directories where data is located and will be saved.

The VIMOS MOS workflow (see figure 4.3) is meant to reduce MOS data up to the science, including spectrophotometric calibrations. It has been thoroughly tested and provides some interactive windows that will help the user to asses the quality of the reduced data and offer the option to change easily the most relevant pipeline parameters. These interactive windows display the most relevant products created by the pipeline.

For more details about using the VIMOS MOS workflow it is strongly recommended that you read the tutorial VIMOS MOS Reflex tutorial available at *http://www.eso.org/sci/software/pipelines/*.

The VIMOS IFU workflow (see figure 4.4) is a basic Reflex workflow that will mainly help in the association of calibration frames to your science. Take into account that it has not undergone a major testing effort but it will help as a useful front end to run the pipeline. It doesn't contain any data visualization.

Please refer to the VIMOS IFU Reflex tutorial available at *http://www.eso.org/sci/software/pipelines/* for further details on the VIMOS IFU workflow.

4.2.2 Using Gasgano

Another convenient tool useful for familiarizing oneself with the VIMOS pipeline recipes and their usage is the graphical user interface *Gasgano*. It provides a complete graphical user interface for data browsing, classifica-

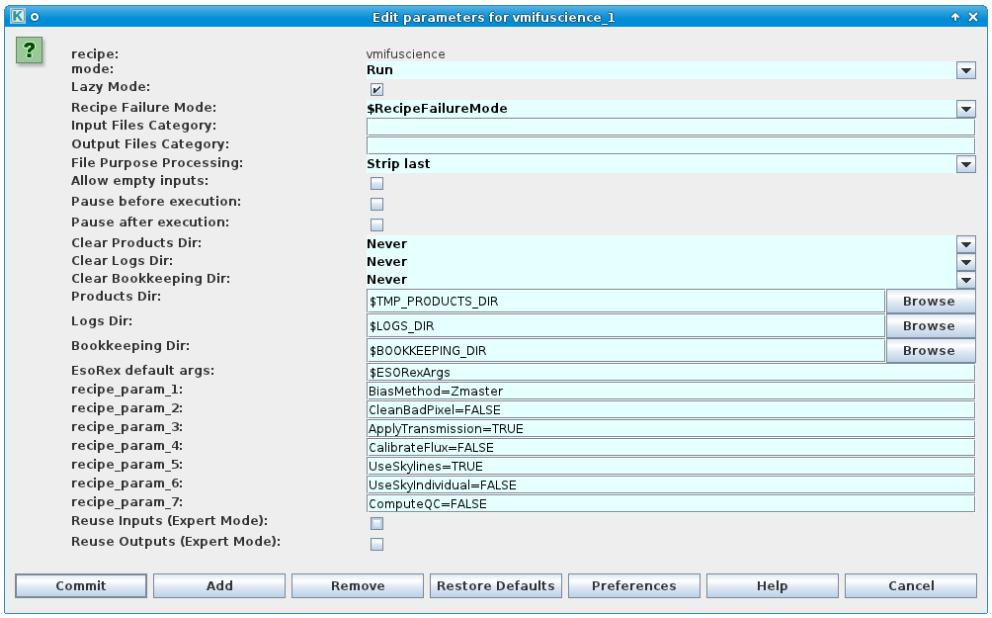

Figure 4.2: *Parameters of a recipe actor. The first group of parameters affect the execution of the pipeline recipe and are common to all recipe actors. The second group of parameters are specific to the pipeline recipe to be called and they are identical to those that can be configured in EsoRex (see 4.2.3).*

tion and association, and offers several other utilities such as easy access to pipeline recipes, documentation and the preferred data display tools.

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

```
gasgano &
```

The *Gasgano* main window will appear. On Figure 4.5 (next page), a view on a complete set of VIMOS IFU data is shown as an example. *Gasgano* can be pointed to the directories where the data to be handled are located using the navigation panels accessible via the *Add/Remove Files* entry of the *File* menu (shown on the upper left of the figure).

The data are hierarchically organised as preferred by the user. In this example the default grouping is shown, and for clarity only the sub-groups belonging to the first VIMOS quadrant are expanded. After each file name are shown the observation date, the target name, the used grism-filter combination and, as an important reference, an identifier of the type of data, listed in the `CLASSIFICATION` field.[2]

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

Frames can be selected from the main window for being processed by the appropriate recipe: on Figure 4.6, the flat field and arc lamp exposures, an already produced master bias frame, and the necessary static calibration

[2]This is known as the data *DO category*, see Section 5, page 46.

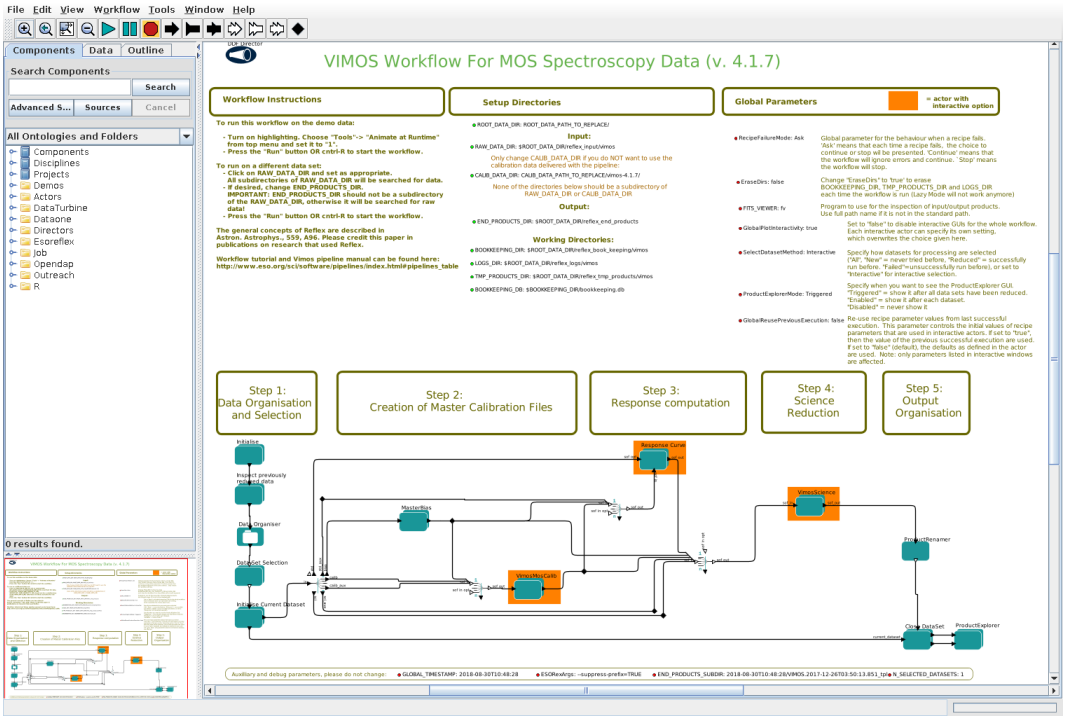

Figure 4.3: *VIMOS MOS Reflex workflow main layout.*

tables, are all selected and sent to the *vmifucalib* recipe. This will open a *Gasgano* recipe execution window (see Figure 4.7), having all the specified files listed in its *Input Frames* panel.

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

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

Please refer to the *Gasgano User's Manual* ([4]) for a more complete description of the *Gasgano* interface.

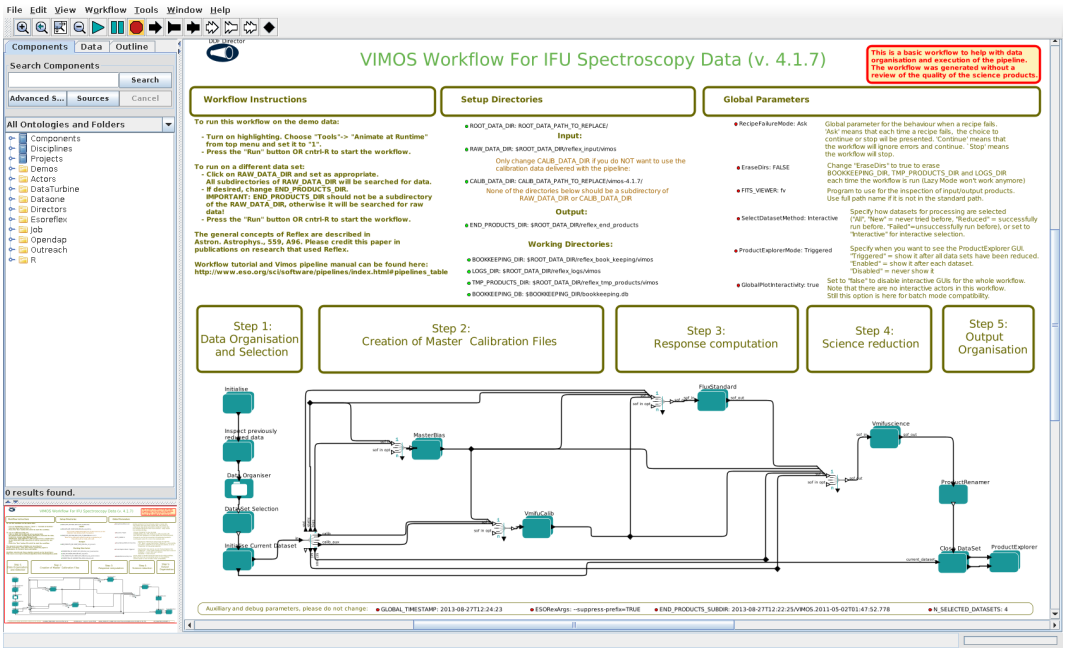

Figure 4.4: *VIMOS IFU Reflex workflow main layout.*

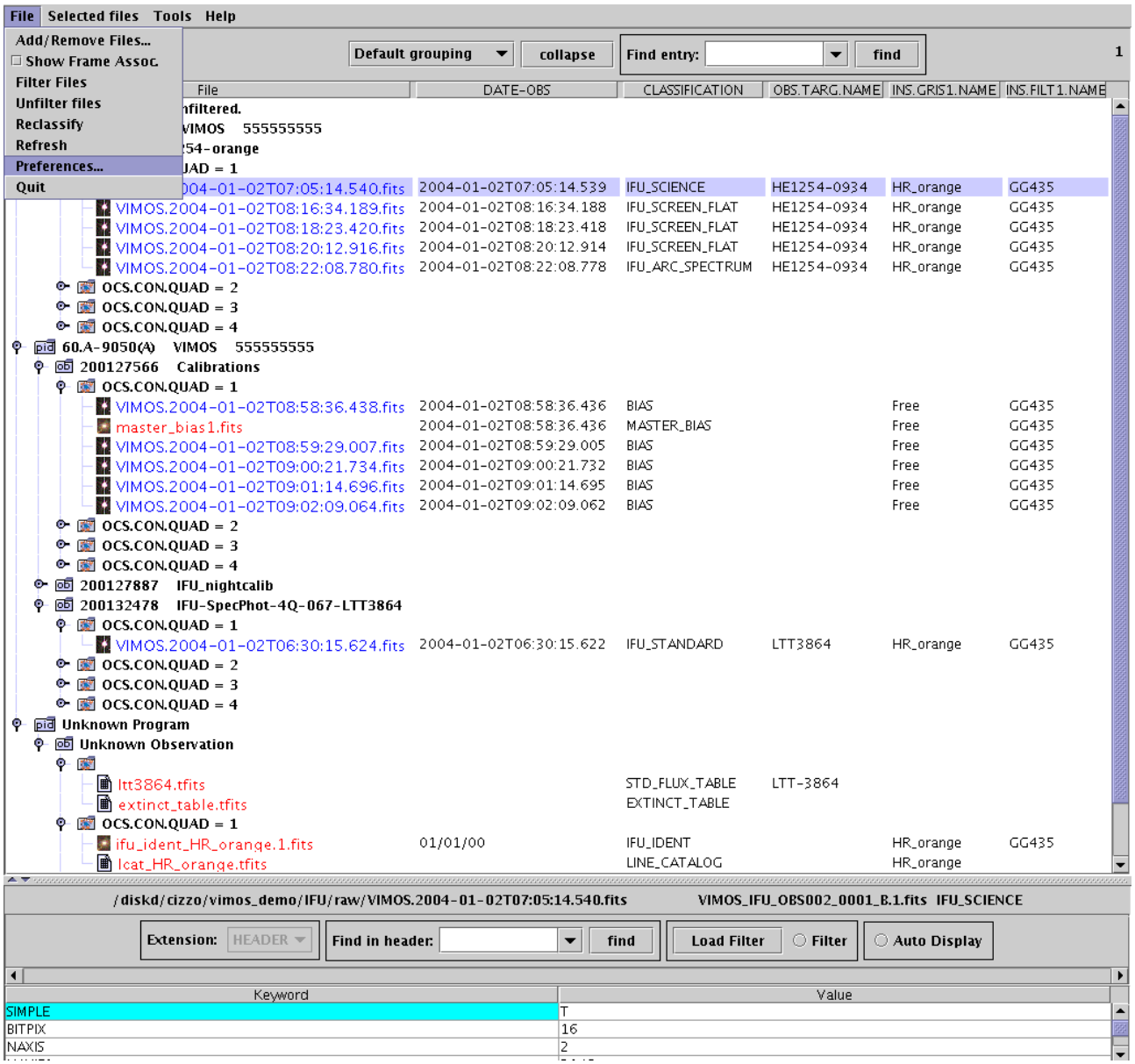

Figure 4.5: *The Gasgano main window.*

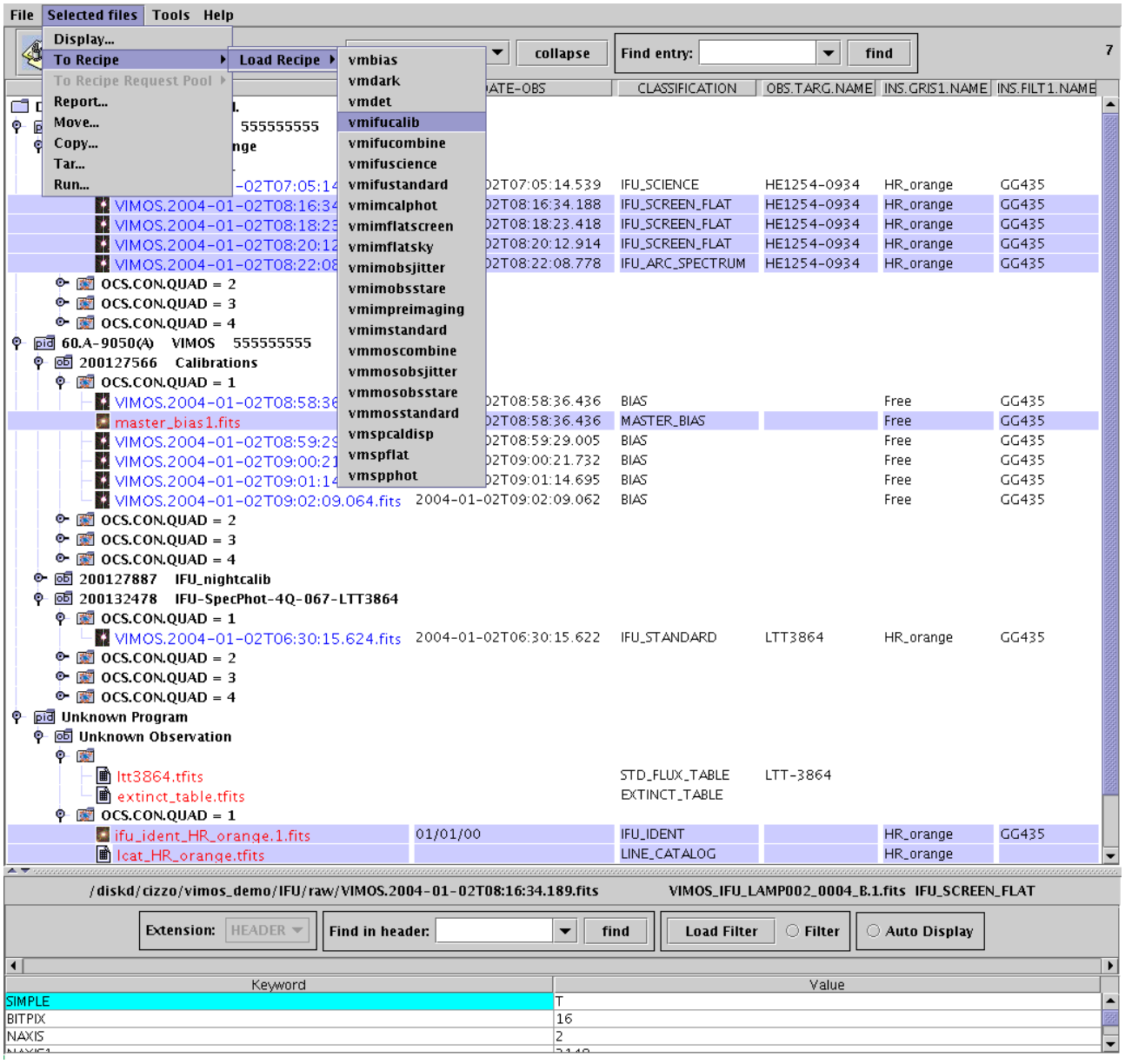

Figure 4.6: *Selecting files to be processed by a VIMOS pipeline recipe.*

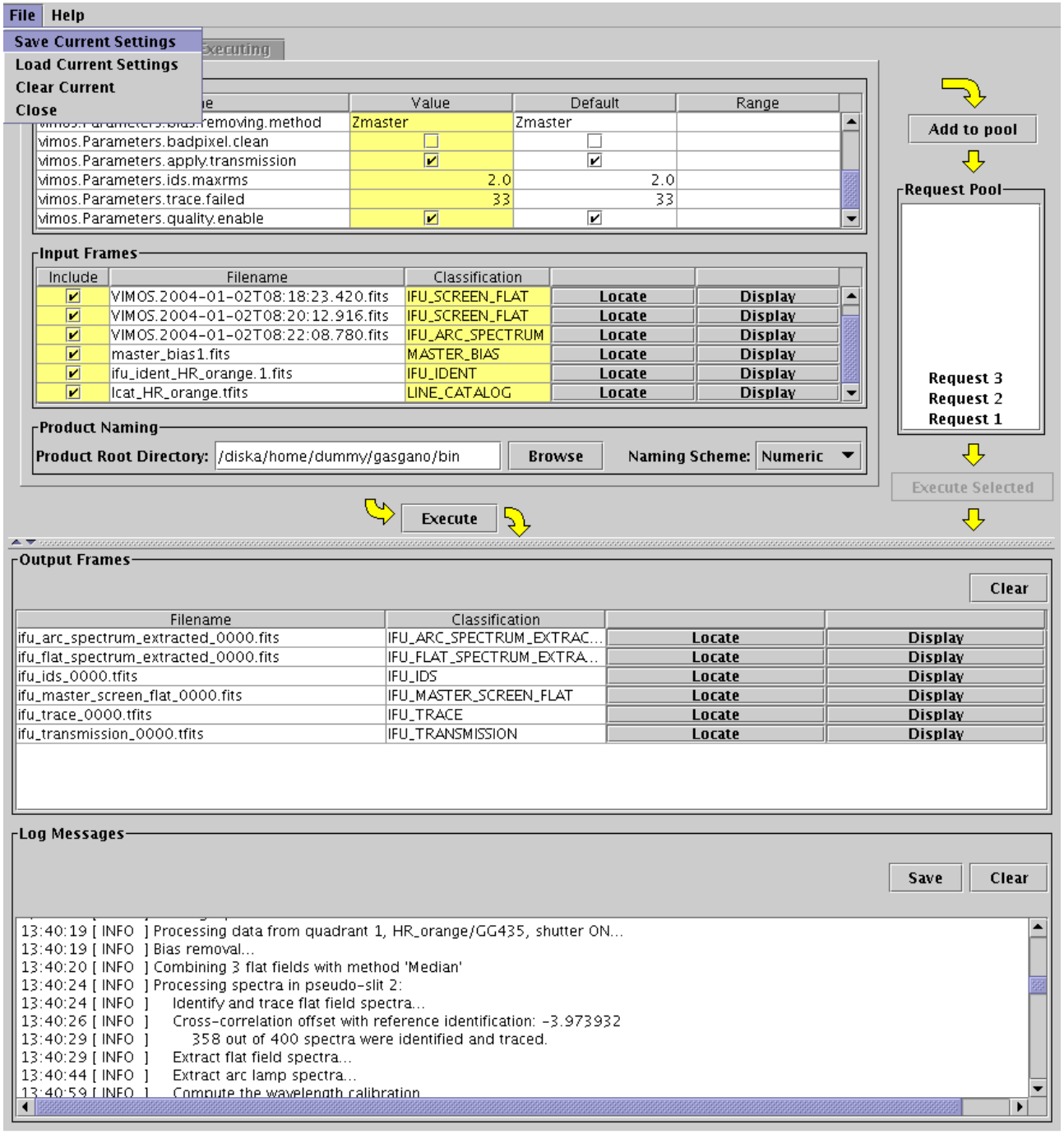

Figure 4.7: *The Gasgano recipe execution window.*

4.2.3 Using EsoRex

EsoRex is a command line utility for running pipeline recipes. It may be embedded by users into data reduction scripts for the automation of processing tasks. On the other hand, *EsoRex* doesn't offer all the facilities available with *Reflex* or *Gasgano*, and the user must classify and associate the data using the information contained in the FITS header keywords (see Section 5, page 46). The user should also take care of defining the input set-of-frames and the appropriate configuration parameters for each recipe run:

The set-of-frames: Each pipeline recipe is run on a set of input FITS data files. When using *EsoRex* the filenames must be listed together with their DO category in an ASCII file, the *set-of-frames* (SOF), that is required when launching a recipe. [3]

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

```
VIMOS.2010-05-11T03:46:22.860.fits        MOS_ARC_SPECTRUM
VIMOS.2010-05-11T03:42:10.264.fits        MOS_SCREEN_FLAT
VIMOS.2010-05-11T03:43:36.323.fits        MOS_SCREEN_FLAT
VIMOS.2010-05-11T03:45:02.342.fits        MOS_SCREEN_FLAT
/home/vimos/cal/mbias.3.fits              MASTER_BIAS
/home/vimos/cal/lcat_MR.1.fits            LINE_CATALOG
/home/vimos/cal/VIMOS_GRS_MR_GG475_1.fits CONFIG_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 by the user in the SOF. In the above example, the recipe *vmmoscalib* will treat the frame `VIMOS.2010-05-11T03:46...` as a MOS_ARC_SPECTRUM, the frame `mbias.3.fits` as a MASTER_BIAS, etc., even when they do not contain this type of data.

Using *Gasgano* as an interface to the pipeline recipes will always ensure a correct classification of all the data frames, assigning the appropriate DO category to each one of them (see Section 4.2.2, page 26).

Recipe configuration: Each pipeline recipe may be assigned an *EsoRex* configuration file, containing the default values of the parameters related to that recipe.[4] The configuration files are normally generated in the directory `$HOME/.esorex` (as will be shown in the subsequent examples), and have the same name as the recipe to which they are 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 ( Average | Median | MinMax | Ksigma | Auto )
vimos.Parameters.stacking.method=Average
```

[3] The set-of-frames corresponds to the *Input Frames* panel of the *Gasgano* recipe execution window (see Figure 4.7, page 32).

[4] The *EsoRex* recipe configuration file corresponds to the *Parameters* panel of the *Gasgano* recipe execution window (see Figure 4.7, page 32).

In this example, the parameter `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 (`StackMethod`) that could be used as a command line option (see ahead).

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.

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, 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. The command:

```
esorex --create-config vmbias
```

will create in the `$HOME/.esorex` directory a configuration file `vmbias.rc` with the default parameters settings for the recipe *vmbias*. This file can then be modified with the preferred text editor.[5]

A recipe can be run by specifying its name to *EsoRex*, together with the name of a set-of-frames. For instance, the following command line would be used to run the recipe *vmbias* for processing the files specified in the set-of-frames `vmbias.sof`:

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

```
esorex 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, and they will supersede the system defaults and/or the configuration file settings. 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 --msg-level=debug vmskyccd --SExtractor.Window=1,1,1900,2300 skyccd
esorex vmmoscalib --dradius=15 --sradius=20 in.sof
esorex --time=true vmmosscience --stack_method=ksigma 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 *vmmoscalib* 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 with a running box of 15 pixels in the dispersion direction, and 20 pixels in the spatial direction. The input SOF is `in.sof`.

Finally, in the *vmmosscience* example the execution of the recipe is timed and the input scientific frames (if more than one is specified) are stacked using a k-sigma clipping method. The input SOF is `input1.sof`.

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

4.3 Example of MOS data reduction using the current recipes

A simple, typical MOS data reduction procedure is described here.[6] It is assumed that the following data are available:

One scientific exposure:

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

One standard star exposure:

```
VIMOS.2004-09-27T03:12:12.006.fits     MOS_STANDARD
```

[6]The procedure using *Gasgano*, instead of EsoRex, is conceptually identical.

Three flat field exposures obtained with the mask used for the scientific exposure:

```
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 obtained with the mask used for the scientific exposure:

```
VIMOS.2004-09-27T19:13:03.631.fits     MOS_ARC_SPECTRUM
```

Three flat field exposures obtained with the mask used for the standard star exposure:

```
VIMOS.2004-09-27T19:22:22.308.fits     MOS_SCREEN_FLAT
VIMOS.2004-09-27T19:23:14.722.fits     MOS_SCREEN_FLAT
VIMOS.2004-09-27T19:24:52.651.fits     MOS_SCREEN_FLAT
```

One arc lamp exposure obtained with the mask used for the standard star exposure:

```
VIMOS.2004-09-27T19:33:44.097.fits     MOS_ARC_SPECTRUM
```

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
```

All the listed data are meant to belong to the same VIMOS quadrant, with the same grism and filter in use.

In the following, it is also assumed for simplicity that the flag `suppress-prefix` is set to `TRUE` in the *EsoRex* configuration file, so that the product file names will just be identical to their product categories, with the extension `.fits`. 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 `/home/vimos/cal`.

The procedure is as follows:

First, a master bias is created with the recipe *vmbias*.

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 and arc lamp exposures, the recipe *vmmoscalib* is used (see Section 7.11, page 106). The input SOF may be defined as follows:

File: *moscalib.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
VIMOS.2004-09-27T19:13:03.631.fits            MOS_ARC_SPECTRUM
master_bias.fits                              MASTER_BIAS
/home/vimos/cal/lcat_LR_red.tfits             LINE_CATALOG
/home/vimos/cal/VIMOS_GRS_LR_red_OS_red.fits  CONFIG_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 vmmoscalib moscalib.sof
```

Several products are created on disk, mainly for check purposes. The products which are necessary for the scientific data reduction are the following:

mos_master_screen_flat.fits: normalised flat field image.

mos_slit_location.fits: slit positions on the CCD.

mos_curv_coeff.fits: coefficients of the spatial curvature fitting polynomials.

mos_disp_coeff.fits: coefficients of the wavelength calibration fitting polynomials.

mos_flat_sed.fits: this contains the spectral energy distribution of the flat. This is needed to properly flux calibrate thet science in the case of holographic prisms.

Products for checking the quality of the result are:

mos_combined_screen_flat.fits: sum of all the input flat field exposures.

mos_curv_traces.fits: table containing the x CCD positions of the detected spectral edges at different y CCD positions, compared with their modeling.

mos_delta_image.fits: deviation from the linear term of the wavelength calibration fitting polynomials.

mos_disp_residuals.fits: residuals for each wavelength calibration fit, produced only if the recipe configuration --check is set.

mos_disp_residuals_table.fits: table containing different kinds of residuals for a sample of wavelength calibration fits.

global_distortion_table.fits: table containing the modeling of the coefficients listed in the `mos_curv_coeff.fits` and `mos_disp_coeff.fits` tables, only produced if more than 6 slits are available.

mos_arc_spectrum_extracted.fits: rectified and wavelength calibrated arc lamp image.

mos_spectra_detection.fits: result of preliminary wavelength calibration applied to the input arc lamp exposure, produced only if the recipe configuration --check is set.

mos_wavelength_map.fits: map of wavelengths on the CCD.

mos_spatial_map.fits: map of spatial positions on the CCD.

mos_slit_map.fits: map of the grism central wavelength, produced only if the recipe configuration --check is set.

mos_spectral_resolution.fits: mean spectral resolution for each reference arc lamp line.

Most of the above FITS files have more than one extension if (like in this case) spectral multiplexing was applied in the observation: each FITS file will include one data section for each rank of spectral multiplexing.

Now the scientific frame can be processed, and for this the recipe *vmmosscience* is used (see Section 7.12, page 123). The following set-of-frames file may be created:

File: *mosscience.sof*

```
VIMOS.2004-09-27T02:39:11.479.fits            MOS_SCIENCE
master_bias.fits                              MASTER_BIAS
mos_master_screen_flat.fits                   MOS_MASTER_SCREEN_FLAT
mos_disp_coeff.fits                           MOS_DISP_COEFF
mos_curv_coeff.fits                           MOS_CURV_COEFF
mos_slit_location.fits                        MOS_SLIT_LOCATION
/home/vimos/cal/VIMOS_GRS_LR_red_OS_red.fits  CONFIG_TABLE
```

Note that the same (optional) CONFIG_TABLE specified in the `moscalib.sof` file is used here. This is advisable, even if not really mandatory.

With the following command:

```
esorex vmmosscience mosscience.sof
```

the following products are created on disk:

mos_science_sky_extracted.fits: image with rectified and wavelength calibrated slit spectra.

mos_science_extracted.fits: image with rectified, wavelength calibrated, and sky subtracted slit spectra.

mos_science_sky.fits: image with rectified and wavelength calibrated slit sky spectra.

mos_unmapped_science.fits: image with the sky subtracted scientific spectra on the CCD.

mos_sci_unmapped_sky.fits: image with the modeled sky spectra on the CCD.

mos_science_reduced.fits: image with extracted objects spectra.

mos_sci_sky_reduced.fits: image with sky corresponding to the extracted objects spectra.

mos_sci_error_reduced.fits: image with the statistical errors corresponding to the extracted objects spectra.

object_sci_table.fits: slit positions on the CCD, on the mapped images, and positions of the detected objects within the slits.

mos_sci_skylines_offsets_slit.fits: table containing the observed sky lines offsets that were used for adjusting the input wavelength solutions.

mos_sci_wavelength_map_sky.fits: map of wavelengths on the CCD.

mos_sci_disp_coeff_sky.fits: wavelength calibration polynomials coefficients after alignment of the solutions to the position of the sky lines.

More products would be created, depending on possible options specified on the command line.

The recipe *vmmosscience* may also be used to handle more than one scientific exposure, provided that all exposures were obtained using the same mask. In that case, the input frames would be reduced one by one, and finally aligned and stacked into products analogous to those described above.

Support for a spectro-photometric calibration is also available, but in this case an atmospheric extinction table (see entry EXTINCT_TABLE, page 124) and a spectral response curve for the present instrument configuration (see MOS_SPECPHOT_TABLE entry, page 131) must also be specified in input.

Spectral response curves can be produced using the same *vmmosscience* recipe. In this example, the following set-of-frames files would be created, respectively for the *vmmoscalib* and *vmmosscience* recipes:

File: *stdcalib.sof*

```
# These are calibrations obtained with the standard star mask:
      VIMOS.2004-09-27T19:22:22.308.fits             MOS_SCREEN_FLAT
      VIMOS.2004-09-27T19:23:14.722.fits             MOS_SCREEN_FLAT
      VIMOS.2004-09-27T19:24:52.651.fits             MOS_SCREEN_FLAT
      VIMOS.2004-09-27T19:33:44.097.fits             MOS_ARC_SPECTRUM
# Static calibrations remain typically the same as for science:
      master_bias.fits                               MASTER_BIAS
      /home/vimos/cal/lcat_LR_red.1.tfits            LINE_CATALOG
      /home/vimos/cal/VIMOS_GRS_LR_red_OS_red.fits   CONFIG_TABLE
```

File: *stdscience.sof*

```
      VIMOS.2004-09-27T03:12:12.006.fits             MOS_STANDARD
      master_bias.fits                               MASTER_BIAS
      mos_master_screen_flat.fits                    MOS_MASTER_SCREEN_FLAT
      mos_disp_coeff.fits                            MOS_DISP_COEFF
      mos_curv_coeff.fits                            MOS_CURV_COEFF
      mos_slit_location.fits                         MOS_SLIT_LOCATION
      /home/vimos/cal/VIMOS_GRS_LR_red_OS_red.fits   CONFIG_TABLE
# Added atmospheric extinction table:
      /home/vimos/cal/extinct_table.tfits            EXTINCT_TABLE
      /home/vimos/cal/gd108.tfits                    STD_FLUX_TABLE
      /home/vimos/cal/vimos_telluric_regions.tfits   TELLURIC_CONTAMINATION
```

The telluric contamination table can be used to avoid telluric regions in the fitting of the response. This depends on parameter *--resp_ignore_mode* (see section 7.12.3). If you want to remove the flat spectral energy distribution before computing the response, you can add the [mos_flat_sed.fits] product created by the *vmmoscalib* recipe to both *stdscience.sof* and *mosscience.sof*. Note that for holographic grisms this is neccessary due to the position dependant response, see 9.16.1 for details.

The produced file `mos_specphot_table.fits` can then be added to the `mosscience.sof` file (previous page). If file `mos_specphot_table.fits` is added to the sof file, the additional production of flux calibrated extracted spectra is performed:

mos_science_flux_extracted.fits: image with rectified, wavelength calibrated, sky subtracted, and flux calibrated slit spectra.

mos_science_flux_reduced.fits: image with extracted and flux calibrated objects spectra.

mos_sci_error_flux_reduced.fits: image with the statistical errors corresponding to the extracted and flux calibrated objects spectra.

4.4 Example of IFU data reduction

A simple, typical IFU data reduction procedure is described here.[7] It is assumed that the following data are available:

One scientific exposure:

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

One standard star exposure:

```
VIMOS.2004-04-27T05:05:11.542.fits     IFU_STANDARD
```

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, with the same grism and filter in use.

In the following, it is also assumed for simplicity that the flag `suppress-prefix` is set to `TRUE` in the *EsoRex* configuration file, so that the product file names will just be identical to their product categories, with the extension `.fits`. 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 `/home/vimos/cal`.

The procedure is as follows:

First, a master bias is created with the recipe *vmbias*.

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:

[7] The procedure using *Gasgano*, instead of EsoRex, is conceptually identical.

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
/home/vimos/cal/lcat_HR_red.tfits           LINE_CATALOG
/home/vimos/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.fits`; the wavelength calibration, `ifu_ids.fits`; and the relative transmission factors, `ifu_transmission.fits`.

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.fits                       IFU_IDS
ifu_trace.fits                     IFU_TRACE
ifu_transmission.fits              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`.

If, as in this case, an IFU exposure of a standard star is available, it can be reduced using the recipe *vmifustandard*:

File: *ifustandard.sof*

```
VIMOS.2004-04-27T05:05:11.542.fits     IFU_STANDARD
master_bias.fits                       MASTER_BIAS
ifu_ids.fits                           IFU_IDS
ifu_trace.fits                         IFU_TRACE
ifu_transmission.fits                  IFU_TRANSMISSION
/home/vimos/cal/extinct_table.tfits    EXTINCT_TABLE
/home/vimos/cal/ltt4816.tfits          STD_FLUX_TABLE
```

This will be processed with:

```
esorex vmifustandard ifustandard.sof
```

This will generate, among other products, the spectro-photometric table, `ifu_specphot_table.fits`. This table, that contains the instrument efficiency and response curves, should be added to the `ifuscience.sof` set-of-frames, together with the atmospheric extinction table, as in the following:

File: *ifuscience.sof*

```
VIMOS.2004-04-27T04:15:24.227.fits     IFU_SCIENCE
master_bias.fits                       MASTER_BIAS
ifu_ids.fits                           IFU_IDS
ifu_trace.fits                         IFU_TRACE
ifu_transmission.fits                  IFU_TRANSMISSION
/home/vimos/cal/extinct_table.tfits    EXTINCT_TABLE
ifu_specphot_table.fits                IFU_SPECPHOT_TABLE
```

and processed by:

```
esorex vmifuscience --CalibrateFlux=true ifuscience.sof
```

This run would then generate, in addition to the usual products of this recipe, also the file containing the extracted and flux calibrated IFU spectra, `ifu_science_flux_reduced.fits`.

4.5 Known problems

The following is a list of currently-known issues with VIMOS recipes, and workarounds, if available:

The spatial curvature model may not be determined accurately from MOS flat field exposures by recipe *vmmoscalib*, if the slit spectra display any spatial overlap, or if spectral multiplexing is present: the confusion introduced at the edges of the slit spectra may strongly bias the tracing task. A safe and complete solution to this problem is not yet available.

The spectral fringing correction performed by recipe *vmmosscience* involves only a sky fringing subtraction, and therefore is of limited help in the case of bright object spectra.

The atmospheric extinction table made available in the static calibration directories (see Table 7.10, page 124), is based on data obtained at La Silla. This table may be replaced with another having the same structure: the wavelengths at which the atmospheric extinction is given can be chosen freely, since all the scientific data reduction recipes (*vmifustandard, vmifuscience, vmmosscience* and *vmsphot*) interpolate the available values as necessary.

The IFU fiber identification performed by recipe *vmifucalib* appears to be negatively affected by changes in temperature. If in the recipe products more than about 50 fibers appear to be "lost" in one pseudo-slit, it may help to rerun the recipe using the "blind" fiber identification method: this method is always triggered if no fiber identification table is specified in the input set-of-frames.

The filter/grism combination OS_red/MR is not supported by the IFU data reduction recipes.

5 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 come from external sources (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 based on FITS keyword values. In the case of raw frames, classification can be defined by looking at least at three keyword values: DPR TYPE, DPR CATG, and DPR TECH. In the case of data products, the classification of the frame is stored into the hierarchical keyword PRO CATG.

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 FITS 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 automatic pipeline environment at Paranl 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 at Paranal 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 4.2.3, page 33).

5.1 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.

5.1.1 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

5.1.2 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

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: `vimos_ima_twilight_flat`

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`

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`

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

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

5.1.3 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: `vmmoscalib`

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
	`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

- **Arc lamp spectra:**

  DO category: `MOS_ARC_SPECTRUM`
  Processed by: `vmmoscalib`

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: vmmosscience

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: vmmosscience

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

5.1.4 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:
  `DPR CATG = CALIB`
  `DPR TYPE = FLAT,LAMP`
  `DPR TECH = IFU`

Association keywords:	Note:
`INS MODE`	Instrument mode
`OCS CON QUAD`	Quadrant used
`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:
  `DPR CATG = CALIB`
  `DPR TYPE = WAVE,LAMP`
  `DPR TECH = IFU`

Association keywords:	Note:
`INS MODE`	Instrument mode
`OCS CON QUAD`	Quadrant used
`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

5.2 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 *Gasgano* ([4]) 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.

Note that, after the definition of the new MOS recipes *vmmoscalib* and *vmmosscience*, several products are no longer in production. Those products are here indicated as deprecated as the recipes that created them.

5.2.1 General frames

- **Master bias:**

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

  This product contains two image extensions: the first one with the master bias data and the second one with the error estimation of the master bias.

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

5.2.2 Direct imaging frames

- **Master sky flat field:**

  DO category: `IMG_MASTER_SKY_FLAT`

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`

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`

  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`

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`

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

- **Sky + sky fringes map:**

  DO category: `IMG_FRINGES`

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`

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`

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

- **Coefficients for photometric calibration:**

  DO category: `PHOT_COEFF_TABLE`

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

- **Initial guess for photometric calibration:**

  DO category: `PHOTOMETRIC_TABLE`

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

- **Photometric catalog:**

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

  No association keyword required.

5.2.3 MOS frames

- **Master screen flat field:**

  DO category: `MOS_MASTER_SCREEN_FLAT`
  Created by: `vmmoscalib`

  This file contains two image extensions: one with the master screen flat and a second one with the error estimation of the master screen flat.

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: `vmmoscalib`

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:
`INS GRIS[1-4] ID`	Grism ID for beam 1 to 4

- **Identification of extracted spectra:**

  DO category: `OBJECT_SCI_TABLE`
  Created by: `vmmosscience`

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

- **Grism dependent configuration parameters:**

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

Association keywords:	Note:
`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: `MOS_CURV_TRACES`
  Created by: `vmmoscalib`

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

  DO category: `MOS_CURV_COEFF`
  Created by: `vmmosscalib`

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
`INS MASK[1-4] ID`	Mask ID for beam 1 to 4

DO category: `MOS_DISP_COEFF`
Created by: `vmmosscalib`

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
`INS MASK[1-4] ID`	Mask ID for beam 1 to 4

DO category: `MOS_SLIT_LOCATION`
Created by: `vmmosscalib`

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
`INS MASK[1-4] ID`	Mask ID for beam 1 to 4

DO category: `MOS_FLAT_SED`
Created by: `vmmosscalib`

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
`INS MASK[1-4] ID`	Mask ID for beam 1 to 4

DO category: `MOS_WAVELENGTH_MAP`
Created by: `vmmoscalib`

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

DO category: `GLOBAL_DISTORTION_TABLE`
Created by: `vmmoscalib`

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

- **Extracted spectra:**

  DO category: `MOS_SCIENCE_REDUCED`
  Created by: `vmmosscience`

  This product contains as many extensions as 2 x multiplexing order. For instance, if multiplexing is 3, then 6 image extensions will be present. Each set of two images corresponds to one multiplexing order. The first extension of the set corresponds to the reduced science and the second to the propagated statistical error.

- **Extracted and flux calibrated spectra:**

  DO category: `MOS_SCIENCE_FLUX_REDUCED`
  Created by: `vmmosscience`

- **Extracted sky spectra:**

  DO category: `MOS_SKY_REDUCED`
  Created by: `vmmosscience`

- **Atmospheric extinction:**

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

- **Extracted standard star spectrum:**

  DO category: `MOS_STANDARD_REDUCED`
  Created by: `vmmosscience`

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

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

Association keywords:	Note:
`OBS TARG NAME`	Standard star name

- **Spectral response and instrument efficiency:**

  DO category: `MOS_SPECPHOT_TABLE`
  Created by: *vmmosscience*

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

5.2.4 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 and instrument efficiency:**

  DO category: `IFU_SPECPHOT_TABLE`
  Created by: *vmifustandard*

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

- **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

6 Static Calibration Data

In the following all the VIMOS static calibration tables related to direct imaging, IFU and spectroscopic modes are listed. The indicated *DO category*, written to the FITS header keyword PRO.CATG, is a label assigned to any data type after it has been classified.

6.1 Detector bad pixel table

DO category: `CCD_TABLE`

This table lists bad pixels for each of the VIMOS detectors. Currently available are:

badpixel.1.tfits
badpixel.2.tfits
badpixel.3.tfits
badpixel.4.tfits
badpixel.5.tfits
badpixel.6.tfits
badpixel.7.tfits
badpixel.8.tfits
VI_GBPM_100801A_Q1.fits
VI_GBPM_100801A_Q2.fits
VI_GBPM_100801A_Q3.fits
VI_GBPM_100801A_Q4.fits

Each table contains the following columns:

X: X coordinate of the bad pixel
Y: Y coordinate of the bad pixel

6.2 Photometric standard stars catalog

DO category: `PHOTOMETRIC_CATALOG`

This table is a list of photometric standard stars parameters. Currently only the Stetson's photometric standard stars catalog is included in the pipeline in file phstd_stetson.tfits.

The phstd_stetson.tfits table includes the following columns:

ID: Name of standard star
RA: Right Ascension (degrees)
DEC: Declination (degrees)

MAG_U: U magnitude (mag)
MAG_B: B magnitude (mag)
MAG_V: V magnitude (mag)
MAG_R: R magnitude (mag)
MAG_I: I magnitude (mag)

Currently the U magnitudes are all set to 99.99, marking them as invalid.

6.3 Grism config table

DO category: `CONFIG_TABLE`

This table defines a subset of recipe configuration parameters controlling the way spectra are extracted for any particular grism. The table consists of a single row of values labeled with the parameters names. A standard grism table is provided for each VIMOS grism: this table is named following the convention:

VIMOS_GRS_<grism resolution>_<grism cwnameID>_<filter name>.fits

where in case no filter is used the *filter name* is set to "free".

If a grism table is used, it will modify the recipe parameters with its new values, with the exception of those which are explicitly given on the command line. Without a grism table, the input recipe parameters values will just be read from the command line, or from an *esorex* configuration file if present, or from their generic default values (that are rarely meaningful). The configuration parameters included in the grism table are the following:

--dispersion	rough expected spectral dispersion
--peakdetection	threshold for preliminary peak detection
--wdegree	polynomial degree for wavelength calibration
--cdegree	polynomial degree for spatial curvature
--startwavelength	start wavelength for spectral extraction
--endwavelength	end wavelength for spectral extraction
--reference	reference wavelength used across the calibration procedure
--RESP_USE_FLAT_SED	whether to use the flat sed normalisation for the response
--resp_fit_degree	maximum degree to use in the polynomial fit of the response
--resp_fit_nknots	maximum number of nknots to use in the spline fit of the response

6.4 Arc lamp lines catalog

DO category: `LINE_CATALOG`

This table contains a set of reference wavelengths (in Ångstrom) for the arc lamp used.

The following columns are present in this table:

WLEN: Wavelength of the arc line
NAME: Name of the arc line, usually with the ion name
FLUX: Flux of the line. Currently ignored by the pipeline
COMMENT: General comment on the line. Ignored by the pipeline
LINE_SET: This can have the value standard or extended. By default the pipeline will only use line

6.5 Master distortion table

DO category: `GLOBAL_DISTORTION_TABLE`

Table containing the modeling of the coefficients of the local distortion models listed in the MOS_DISP_COEFF and the MOS_CURV_COEFF tables. This table is used for enabling the on-line processing of scientific data with the recipe *vmmosscience* when appropriate (day) calibrations are not yet available.

Conventionally this table consists of 6 columns and 10 rows. Each row corresponds to the modeling of one coefficient of the original polynomial coefficients belonging to the local distortion solutions (presumably obtained with a calibration mask), performed by fitting a bivariate polynomial:

$$c_r = \sum_{i=0}^{2} \sum_{j=0}^{2-i} a_{ij} x^i y^j$$

where r is the table row number (counted from 0) and c_r is a polynomial coefficient of a local solution. For $r = 0$ and $r > 6$ (x, y) are positions on the telescope focal plane (e.g., on a mask), otherwise they are positions on the CCD. The first 6 table rows are a global description of the dispersion solution up to the fifth polynomial degree; these rows are followed by a row where just the first element is assigned the value of the central wavelength used for the given dispersion solution. The remaining 3 rows are a global description of the spatial curvature up to the second polynomial degree. The local dispersion solutions could be obtained with:

$$x = \sum_{r=0}^{5} c_r (\lambda - \lambda_o)^r$$

where x is the x CCD pixel position and λ_o is the central wavelength of the grism used. The local spatial curvature solutions could be obtained with:

$$y = \sum_{r=7}^{9} c_r x^{(r-7)}$$

where y is the y CCD pixel position and x is obtained with the previous formula.

The global distortion table columns are labeled a00, a01, a02, a10, a11, a20, indicating the coefficients of the fitting bivariate polynomials.

The global distortion table is produced by the *vmmoscalib* recipe with the tag GLOBAL_DISTORTION_TABLE

6.6 Atmospheric extinction table

Currently the atmospheric extinction table valid for Paranal, as extended by Moehler et al. (2014, A&A 568, A9), is made available in the calibration directory, in a file named `extinct_table.fits`.

This table includes the following columns:

wave: Wavelength (Å)
extinction: Atmospheric extinction (mag/airmass)

6.7 Standard star flux table

A set of standard star flux tables, corresponding to the 30 spectro-photometric standard stars which are included in the VIMOS calibration plan is available in the calibration directory.

A standard star flux table includes the following columns:

WAVE: Wavelength (Å)
FLUX: Flux (10^{-16} erg cm^{-2} s^{-1} Å^{-1})
BIN: Bin width (Å)
STLLR_ABSORP: Flag to specify if a bin is affected by stellar absorption

The names of the available standard star flux tables, and the name of the standard stars as reported in the FITS header keyword `ESO OBS TARG NAME`, are listed in Table 7.14.

In order to use the information in the column STLLR_ABSORP, the parameter *--resp_ignore_mode* must include the string *stellar_absorption*.

6.8 Telluric contamination table

DO category: `TELLURIC_CONTAMINATION`

The file named `vimos_telluric_regions.fits` contains a list of wavelength intervals that might be affected by atmospheric telluric contamination. Since each grism might be affected in a different way, there is a column per grism with the proper flag.

In order to use this information, the *vmmosscience* recipe has to have this table as an input and parameter *--resp_ignore_mode* must include the string *telluric*.

6.9 IFU fiber identification

DO category: `IFU_IDENT`

This calibration contains an image that can be used to identify the fibers, i. e., to give an approximate position for each IFU fiber along an image row. This calibration is produced by an external MIDAS procedure by means of trial and error. If this calibration is wrong, the reconstructed image will show an scrambled pattern.

Each time a physical grism is replaced or simply removed and put back this calibration must be updated. Therefore there are several versions of this file that should be used depending on the observing date. The MJD-OBS keyword of the file can be used to get the starting point from which a certain calibration can be used.

7 Pipeline Recipes Interfaces

In this section the usage of the VIMOS pipeline recipes is described in detail, including input files, parameters, output files, and QC parameters.

7.1 vimos_bias_combine

7.1.1 Description

Combine a list of bias frames into a mean ('master') bias frame. Optionally compare the output frame to a reference bias frame.

7.1.2 Input frames

Frame tag	**Constraint**	**# files**
`BIAS`	Required	$1 \leq n$
`REFERENCE_BIAS`	Optional	1

Note that a complete set of four files (one per chip) does not necessarily have to be provided to the recipe. However, an incomplete set may create pipeline products that are not very useful as input to other recipes.

7.1.3 Recipe parameters

Parameter	Type	Values **default**,other	Description
`combtype`	string	**median**, mean	When combining images, use the mean or median pixel value.
`xrej`	bool	**true**, false	If true, an extra rejection cycle is performed after combination, i.e. an extra pass ensures that what has been clipped deserves to have been clipped.
`thresh`	float	**5.0**, $0 < x$	During combination, reject pixels more than `thresh` sigma above the background
`ncells`	int	**64**, 1,2,4,8,16,32	If a reference bias is provided, divide each chip into `ncells` equal-area cells when calculating the statistical difference between the reference bias and master bias.

continued on next page

Parameter	Type	Values **default**,other	Description
`prettynames`	bool	**false**, true	If false, output files will have standard name. If true, files will have a more descriptive name.

7.1.4 Product frames

PRO.CATG	Filename		Description
	Default	Pretty	
MASTER_BIAS	biascomb.fits	bias_<NIGHT>.fits	Master bias image
DIFFIMG_BIAS	biasdiff.fits	biasdiff_<NIGHT>.fits	If a reference bias is provided in SoF, this image is the difference between the master bias and reference bias, i.e. (MASTER - REFERENCE).
DIFFIMG_STATS_BIAS	biasdifftab.fits	biasdifftab_<NIGHT>.fits	If a reference bias is provided in SoF, this is a table with statistical description of the difference image within small areas of each chip.

Note that <NIGHT> refers to the eight digit representation of the 'civil' night (local time starting at midday) on which the data were taken, e.g. 20140820.

7.1.5 Quality control parameters

The following QC parameters are found in each extension of the `MASTER_BIAS`:

`QC.BIASMED`: Median of mean bias frame

`QC.BIASRMS`: RMS of mean bias frame

`QC.USCAN_MED`: Median of mean bias frame underscan

`QC.USCAN_RMS`: RMS of mean bias frame underscan

`QC.OSCAN_MED`: Median of mean bias frame overscan

`QC.OSCAN_RMS`: RMS of mean bias frame overscan

The following QC parameters are found in each extension of the `DIFFIMG_BIAS`:

`QC.BIAS_DIFFMED`: Median of bias difference image

`QC.BIAS_DIFFRMS`: RMS of bias difference image

7.2 vimos_dark_combine

7.2.1 Description

Combine a list of dark frames into a mean ('master') dark frame. Optionally compare the output frame to a reference dark frame.

7.2.2 Input frames

Frame tag	Constraint	# files
DARK	Required	$1 \leq n$
MASTER_BIAS	Required	1
REFERENCE_DARK	Optional	1
MASTER_BPM	Optional	1
MASTER_CONF	Optional	1

Note that a complete set of four files (one per chip) does not necessarily have to be provided to the recipe. However, an incomplete set may create pipeline products that are not very useful as input to other recipes.

Also note that if both `MASTER_BPM` and `MASTER_CONF` are given, the `MASTER_CONF` will be used. In practice, the number of raw `DARK` files should be at least 5 in order to get useful results.

7.2.3 Recipe parameters

Parameter	Type	Values **default**,other	Description
`combtype`	string	**median**, mean	When combining images, use the mean or median pixel value.
`scaletype`	string	**exptime**, none, additive, multiplicative	If not `none`, scale each image by parameter value to bring background values into agreement before combining
`xrej`	bool	**true**, false	If true, an extra rejection cycle is performed after combination, i.e. an extra pass ensures that what has been clipped deserves to have been clipped.
`thresh`	float	**5.0**, $0 < x$	During combination, reject pixels more than `thresh` sigma above the background
`ncells`	int	**8**, 1,2,4,16,32,64	If a reference dark is provided, divide each chip into `ncells` equal-area cells when calculating the statistical difference between the reference dark and master dark.
`prettynames`	bool	**false**, true	If false, output files will have standard name. If true, files will have a more descriptive name.

7.2.4 Product frames

PRO.CATG	Filename		Description
	Default	Pretty	
MASTER_DARK	darkcomb.fits	dark_<NIGHT>.fits	Master dark image
DIFFIMG_DARK	darkdiff.fits	darkdiff_<NIGHT>.fits	If a reference dark is provided in SoF, this image is the difference between the master dark and reference dark, i.e. (MASTER - REFERENCE).
DIFFIMG_STATS_DARK	darkdifftab.fits	darkdifftab_<NIGHT>.fits	If a reference dark is provided in SoF, this is a table with statistical description of the difference image within small areas of each chip.

Note that <NIGHT> refers to the eight digit representation of the 'civil' night (local time starting at midday) on which the data were taken, e.g. 20140820.

7.2.5 Quality control parameters

The following QC parameters are found in each extension of the `MASTER_DARK`:

`QC.DARKMED`: Median of mean dark frame

`QC.DARKRMS`: RMS of mean dark frame

`QC.PARTICLE_RATE`: [N/(detector*sec)] Particle rate where N is number of objects classified as cosmic rays

`QC.NHOTPIX`: Number of hot pixels, where a hot pixel is one that is consistently too high

`QC.HOTFRAC`: Hot pixel fraction

The following QC parameters are found in each extension of the `DIFFIMG_DARK`:

`QC.DARKDIFF_MED`: Median of dark difference image

`QC.DARKDIFF_RMS`: RMS of dark difference image

7.3 vimos_twilight_flat_combine

7.3.1 Description

Combine a list of twilight flat frames into a mean ('master') frame. Optionally compare the output frame to a reference twilight flat frame.

7.3.2 Input frames

Frame tag	Constraint	# files
FLAT_TWILIGHT	Required	$1 \leq n$
MASTER_BIAS	Required	1
MASTER_DARK	Required	1
REFERENCE_TWILIGHT_FLAT	Optional	1
MASTER_BPM	Optional	1
MASTER_CONF	Optional	1

Note that a complete set of four files (one per chip) does not necessarily have to be provided to the recipe. However, an incomplete set may create pipeline products that are not very useful as input to other recipes.

If both `MASTER_BPM` and `MASTER_CONF` are provided, only the `MASTER_CONF` will be used. It is recommended to use at least 10 raw flats with $\approx 20,000 - 30,000$ ADU/pixel to create a robust, high S/N master flat.

7.3.3 Recipe parameters

Parameter	Type	Values **default**,other	Description
`combtype`	string	**median**, mean	When combining images, use the mean or median pixel value.
`scaletype`	string	**multiplicative**, additive, exp-time, none	If not `none`, scale each image by parameter value to bring background values into agreement before combining
`lthr`	float	**4000**, $0 \leq x \leq 65535$	Discard pixels with values below `lthr` ADUs in the raw flat images.
`hthr`	float	**60000**, $0 \leq x \leq 65535$	Discard pixels with values higher than `hthr` ADUs in the raw flat images.
`thresh`	float	**5.0**, $0 < x$	During combination, reject pixels more than `thresh` sigma above the background
`xrej`	bool	**true**, false	If true, an extra rejection cycle is performed after combination, i.e. an extra pass ensures that what has been clipped deserves to have been clipped.
`ncells`	int	**8**, 1,2,4,16,32,64	If a reference flat is provided, divide each each chip into `ncells` equal-area cells when calculating the statistical difference between the reference flat and master flat.
`prettynames`	bool	**false**, true	If false, output files will have standard name. If true, files will have a more descriptive name.

7.3.4 Product frames

PRO.CATG	Filename		Description
	Default	Pretty	
MASTER_TWILIGHT_FLAT	twilightcomb.fits	<FILTER>_flat_<NIGHT>.fits	Combined twilight flat image
MASTER_CONF	twilightconf.fits	<FILTER>_conf_<NIGHT>.fits	Confidence map derived from raw flat images
MASTER_BPM	bpmmap.fits	<FILTER>_bpm_<NIGHT>.fits	Bad pixel map derived from raw flat images
RATIOIMG_TWILIGHT_FLAT	twilightratio.fits	<FILTER>_ratio_<NIGHT>.fits	If a reference flat is provided in SoF, this is the ratio of the master dark to the reference flat, i.e. (MASTER/REFERENCE).
RATIOIMG_STATS_TWILIGHT_FLAT	twilightratiotab.fits	<FILTER>_ratiotab_<DATE>.fits	If a reference flat is provided in SoF, this is a table with statistical description of the ratio image within small areas of each chip.

Note that <NIGHT> refers to the eight digit representation of the 'civil' night (local time starting at midday) on which the data were taken, e.g. 20140820.

7.3.5 Quality control parameters

The following QC parameters are found in each extension of `MASTER_TWILIGHT_FLAT`:

`QC.FLATRMS`: RMS of output flat

`QC.FLATMIN`: Ensemble minimum

`QC.FLATMAX`: Ensemble maximum

`QC.FLATAVG`: Ensemble average

`QC.FLATRNG`: Ensemble range

`QC.TWIPHOT`: [adu] Estimated photon noise

`QC.TWISNRATIO`: Estimated S/N ratio

The following QC parameters are found in each extension of the `RATIOIMG_TWILIGHT_FLAT`:

`QC.FLATRATIO_MED`: Median of ratio map

`QC.FLATRATIO_RMS`: RMS of ratio map

The following QC parameters are found in each extension of the `MASTER_BPM`:

`QC.NBAD`: Number of bad pixels detected

`QC.BADFRAC`: Fraction of bad pixels detected

7.4 vimos_detector_noise

7.4.1 Description

Compute the readnoise and gain for the VIMOS detectors using two bias frames and two flat field frames.

7.4.2 Input frames

Frame tag	Constraint	# files
`FLAT_TWILIGHT`	Required	$1 \leq n$
`BIAS`	Required	Same as `FLAT_TWILIGHT`

continued on next page

Frame tag	**Constraint**	**# files**
MASTER_BPM	Optional	1
MASTER_CONF	Optional	1

Note that this recipe will process the first two complete sets of chips (8 files) that appear in the SoF. An incomplete set may create pipeline products that are not very useful as input to other recipes.

If both MASTER_BPM and MASTER_CONF are provided, only the MASTER_CONF will be used. Users are advised to select two flat fields of similar illumination that are bright enough to have a very high S/N, but not so bright that they are near saturation. It is also recommended to run this recipe using data close in time to the science frames, because the detector readnoise and gain may vary over time.

7.4.3 Recipe parameters

Parameter	Type	Values **default**,other	Description
thresh	float	**5.0**, $0 < x$	During combination, reject pixels more than thresh sigma above the background

7.4.4 Product frames

PRO.CATG	Filename		Description
	Default	Pretty	
MASTER_READGAIN	readgain.fits	N/A	Table containing readnoise, gain, and covariance for each chip

7.4.5 Quality control parameters

This recipe does not create products with QC parameters.

7.5 vimos_fringe_combine

7.5.1 Description

Take a list of science frames and correct them to remove bias, dark and flat field signatures. Combine the images to form a master fringe frame.

7.5.2 Input frames

Frame tag	**Constraint**	**# files**
OBJECT	Required	$1 \leq n$
MASTER_BIAS	Required	1
MASTER_DARK	Required	1
MASTER_TWILIGHT_FLAT	Required	1
MASTER_CONF	Required	1
MASTER_READGAIN	Required	1

Note that a complete set of four files (one per chip) does not necessarily have to be provided to the recipe. However, an incomplete set may create pipeline products that are not very useful as input to other recipes.

Users are advised to use images with good seeing, photometric conditions, and uncrowded fields to create a useful master fringe frame. Analysis of archival data shows that a master fringe frame in i is stable on time scales of about one year. For z, the fringe frame is much less stable (a few hours).

7.5.3 Recipe parameters

Parameter	Type	Values **default**,other	Description
prettynames	bool	**false**, true	If false, output files will have standard name. If true, files will have a more descriptive name.
psm_ipix	int	**5**, $1 \leq n \leq 10^5$	for sky bkg estimation, min. number of pixels for objects
psm_thresh	float	**3.0**, $0 \leq x \leq 10^{10}$	for sky bkg estimation, object detection threshold above sky in units of σ
psm_nbsize	int	**64**, $1 \leq x \leq 2048$	for sky bkg estimation, size of bkg smoothing box in pixels

7.5.4 Product frames

PRO.CATG	Filename		Description
	Default	Pretty	
MASTER_FRINGE	fringe.fits	<FILTER>_fringe_<DATE-OBS>.fits	Master fringe image
MASTER_FRINGE_VAR	fringevar.fits	<FILTER>_fringevar_<DATE-OBS>.fits	Variance map of MASTER_FRINGE

Note that <DATE-OBS> is the mean observation date among the first complete group of 4 files listed in the SoF (among the collection of files used to create that product), e.g. 2008-09-25T08:46:03.577. (Note that this string may not necessarily appear in any of the raw input filenames.)

7.5.5 Quality control parameters

This recipe does not create any QC parameters for product frames.

7.6 vimos_standard_process

7.6.1 Description

Take a list of standard frames and correct them to remove bias, dark and flat field signatures. Optionally defringe the reddest filter images. Astrometrically and photometrically calibrate the individual images, but do not stack any images together.

7.6.2 Input frames

Frame tag	**Constraint**	**# files**
STD	Required	$1 \leq n$
MASTER_BIAS	Required	1
MASTER_DARK	Required	1
MASTER_TWILIGHT_FLAT	Required	1
MASTER_CONF	Required	1
MASTER_READGAIN	Required	1
PHOTCAL_TAB	Required	1
SCHLEGEL_MAP_NORTH	Required	1
SCHLEGEL_MAP_SOUTH	Required	1
MASTER_FRINGE	Optional	1
MASTER_FRINGE_VAR	Optional	1
MASTER_2MASS_CATALOGUE_ASTROM	Optional	1
MASTER_PPMXL_CATALOGUE_ASTROM	Optional	1
MASTER_APASS_CATALOGUE_ASTROM	Optional	1
MASTER_LOCAL_CATALOGUE_ASTROM	Optional	1
MASTER_APASS_CATALOGUE_PHOTOM	Optional	1
MASTER_LOCAL_CATALOGUE_PHOTOM	Optional	1

Note that a complete set of four files (one per chip) does not necessarily have to be provided to the recipe. However, an incomplete set may create pipeline products that are not very useful.

7.6.3 Recipe parameters

Parameter	Type	Values **default**,other	Description

continued on next page

Parameter	Type	Values **default**,other	Description
`preview_only`	bool	**false**, true	If true, don't run recipe but instead print out a summary of how data will be processed
`minphotom`	int	**1**, $1 \leq n \leq 10^5$	During photometric calibration, at least `minphotom` number of standards must be used.
`prettynames`	bool	**false**, true	If false, output files will have standard name. If true, files will have a more descriptive name.
`savemstd`	bool	**false**, true	If true, create matched standard catalogues
`cdssearch_astrom`	string	**none**, 2mass, usnob, ppmxl, wise	If not `none`, retrieve subset of specified catalogue from CDS for astrometric calibration
`cdssearch_photom`	string	**none**, apass	If not `none`, retrieve subset of specified catalogue from CDS for photometric calibration
`ignore_fringe`	bool	**false**, true	If true, do not correct for fringing even if a `MASTER_FRINGE` is present in the SoF.
`src_cat_ipix`	int	**5**, $1 \leq n \leq 10^5$	min. number of pixels for catalogued objects
`src_cat_thresh`	float	**2.5**, $0 < x$	detection threshold above sky in units of σ above bkg for catalogued objects
`src_cat_icrowd`	bool	**true**, false	if true, use deblending for catalogued objects
`src_cat_rcore`	float	**5.0**, $0 < x \leq 1024$	size of standard aperture in units of pixels for catalogued objects
`src_cat_nbsize`	int	**128**, $1 \leq x \leq 2048$	size of smoothing box in pixels when estimating bkg for catalogued objects
`cacheloc`	string	**.**, any string	directory where cache of calibration catalogue data ('.catcache') is stored relative to current working directory (.)
`magerrcut`	float	**100.0**, any float	objects in the photometric reference catalogue with a magnitude error greater than this value will be excluded from the calibration

continued on next page

Parameter	Type	Values **default**,other	Description

7.6.4 Product frames

PRO.CATG	Filename		Description
	Default	Pretty	
BASIC_CALIBRATED_STD	exp_[N].fits	<FILE>_ex.fits	Calibrated image of each input standard star field
BASIC_VAR_MAP_STD	exp_var_[N].fits	<FILE>_ex_var.fits	Variance map of BASIC_CALIBRATED_STD
OBJECT_CATALOGUE_STD	exp_cat_[N].fits	<FILE>_ex_cat.fits	Catalogue of objects in BASIC_CALIBRATED_STD
MATCHSTD_ASTROM	mstd_a_[N].fits	<FILE>_ex_mstd_a.fits	Matched astrometric standard catalogue for BASIC_CALIBRATED_STD
MATCHSTD_PHOTOM	mstd_p_[N].fits	<FILE>_ex_mstd_p.fits	Matched photometric standard catalogue for BASIC_CALIBRATED_STD

The index number [N] reflects the chronological order of the relevant files in the SoF, e.g. the one in the group with the earliest MJD-OBS has [N] = 1.

<FILE> is the root name of the file in the SoF with chip = 'BRIAN' among the collection of files used to create that product, e.g. VIMOS.2011-06-21T08:40:45.959. If 'BRIAN' is missing, the next chip (alphanumerically) is used.

7.6.5 Quality control parameters

The following QC parameters are found in each extension of the `BASIC_CALIBRATED_STD` and `MATCHSTD_PHOTOM` files:

`QC.WCS_DCRVAL1`: [deg] change in crval1

`QC.WCS_DCRVAL2`: [deg] change in crval2

`QC.WCS_DTHETA`: [deg] change in rotation

`QC.WCS_SCALE`: [arcsec] mean plate scale

`QC.WCS_SHEAR`: [deg] abs(xrot) - abs(yrot)

`QC.WCS_RMS`: [arcsec] Average error in WCS fit

`QC.MAGZPT`: [mag] photometric zeropoint

`QC.MAGZERR`: [mag] photometric zeropoint error (1.48 * median absolute deviation of MAGZPT among all extensions). Set to 1.0 if photometric calibration failed.

`QC.LIMITING_MAG`: [mag] 5 sigma limiting mag.

`QC.SKYBRIGHT`: [mag/arcsec**2] sky brightness

`QC.MAGNZPT`: number of stars available for magzpt calc; that actual number used may be less than this, depending on the value of `magerrcut`

The following QC parameters are found in each extension of the `OBJECT_CATALOGUE_STD` files:

`QC.SATURATION`: [adu] Saturation level

`QC.MEAN_SKY`: [adu] Median sky brightness

`QC.SKY_NOISE`: [adu] Pixel noise at sky level

`QC.IMAGE_SIZE`: [arcsec] Average FWHM of stellar object

`QC.ELLIPTICITY`: Average stellar ellipticity (1-b/a)

`QC.POSANG`: [degrees] Median position angle (from North)

`QC.APERTURE_CORR`: Stellar ap-corr 1x core flux

`QC.NOISE_OBJ`: Number of noise objects

`QC.MAGZPT`: [mag] photometric zeropoint

`QC.MAGZERR`: [mag] photometric zeropoint error (1.48 * median absolute deviation of MAGZPT among all extensions). Set to 1.0 if photometric calibration failed.

`QC.LIMITING_MAG`: [mag] 5 sigma limiting mag.

`QC.SKYBRIGHT`: [mag/arcsec**2] sky brightness

`QC.MAGNZPT`: number of stars available for magzpt calc; that actual number used may be less than this, depending on the value of `magerrcut`

The following QC parameters are found in each extension of the `BASIC_VAR_MAP_STD` files:

`QC.WCS_DCRVAL1`: [deg] change in crval1

`QC.WCS_DCRVAL2`: [deg] change in crval2

The following QC parameters are found in each extension of the `MATCHSTD_ASTROM` files:

`QC.SATURATION`: [adu] Saturation level

`QC.MEAN_SKY`: [adu] Median sky brightness

`QC.SKY_NOISE`: [adu] Pixel noise at sky level

`QC.IMAGE_SIZE`: [arcsec] Average FWHM of stellar object

`QC.ELLIPTICITY`: Average stellar ellipticity (1-b/a)

`QC.POSANG`: [degrees] Median position angle (from North)

`QC.APERTURE_CORR`: Stellar ap-corr 1x core flux

`QC.NOISE_OBJ`: Number of noise objects

7.7 vimos_science_process

7.7.1 Description

Take a list of science frames and correct them to remove bias, dark and flat field signatures. Optionally defringe the reddest filter images. Stack jitter sequences. Astrometrically and photometrically calibrate the stacks.

7.7.2 Input frames

Frame tag	Constraint	# files
`OBJECT`	Required	$1 \leq n$
`MASTER_BIAS`	Required	1
`MASTER_DARK`	Required	1

continued on next page

Frame tag	**Constraint**	**# files**
MASTER_TWILIGHT_FLAT	Required	1
MASTER_CONF	Required	1
MASTER_READGAIN	Required	1
PHOTCAL_TAB	Required	1
SCHLEGEL_MAP_NORTH	Required	1
SCHLEGEL_MAP_SOUTH	Required	1
MASTER_FRINGE	Optional	1
MASTER_FRINGE_VAR	Optional	1
MASTER_2MASS_CATALOGUE_ASTROM	Optional	1
MASTER_PPMXL_CATALOGUE_ASTROM	Optional	1
MASTER_APASS_CATALOGUE_ASTROM	Optional	1
MASTER_LOCAL_CATALOGUE_ASTROM	Optional	1
MASTER_APASS_CATALOGUE_PHOTOM	Optional	1
MASTER_LOCAL_CATALOGUE_PHOTOM	Optional	1
MATCHSTD_PHOTOM	Optional	1

Note that a complete set of four files (one per chip) does not necessarily have to be provided to the recipe. However, an incomplete set may create pipeline products that are not very useful.

7.7.3 Recipe parameters

Parameter	Type	Values **default**,other	Description
savecat	bool	**false**, true	If true, create catalogues of objects detected in each input OBJECT image
savemstd	bool	**false**, true	If true, create matched standard catalogues
preview_only	bool	**false**, true	If true, don't run recipe but instead print out a summary of how data will be processed
minphotom	int	**1**, $1 \leq n \leq 10^5$	During photometric calibration, at least minphotom number of standards must be used.
prettynames	bool	**false**, true	If false, output files will have standard name. If true, files will have a more descriptive name.

continued on next page

Parameter	Type	Values **default**,other	Description
`chop_crud`	string	**hardconf_pix**, none, lowconf_block, hardcoded, lowconf_pix	Crop parts of each image that are vignetted by filter holder. If `lowconf_block`, remove a block of pixels where a large decrease in value of confidence map occurs. If `hardcoded`, remove pre-defined region of pixels. If `lowconf_pix`, ignore individual pixels with confidence < 80. If `hardconf_pix`, apply both `hardcoded` and `lowconf_pix`.
`cdssearch_astrom`	string	**none**, 2mass, usnob, ppmxl, wise	If not `none`, retrieve subset of specified catalogue from CDS for astrometric calibration
`cdssearch_photom`	string	**none**, apass	If not `none`, retrieve subset of specified catalogue from CDS for photometric calibration
`ignore_fringe`	bool	**false**, true	If true, do not correct for fringing even if `MASTER_FRINGE` is present in the SoF.
`stk_cat_ipix`	int	**10**, $1 \leq n \leq 10^5$	min. number of pixels for catalogued objects in stack
`stk_cat_thresh`	float	**1.5**, $0 < x \leq 10^{10}$	detection threshold above sky in units of σ above bkg for catalogued objects in stack
`stk_cat_icrowd`	bool	**true**, false	if true, use deblending for catalogued objects in stack
`stk_cat_rcore`	float	**10.0**, $0 < x \leq 1024$	size of standard aperture in units of pixels for catalogued objects in stack
`stk_cat_nbsize`	int	**128**, $1 \leq x \leq 2048$	size of smoothing box when estimating bkg for catalogued objects in stack
`stk_lthr`	float	**3.0**, $0 < x \leq 10^{10}$	when stacking, reject pixel more than `stk_lthr` units of σ below sky bkg
`stk_hthr`	float	**3.0**, $0 < x \leq 10^{10}$	when stacking, reject pixel more than `stk_hthr` units of σ above sky bkg

continued on next page

Parameter	Type	Values **default**,other	Description
`stk_method`	string	**linear**, nearest	when stacking, an RA/DEC is calculated for each input pixel. If `nearest`, the pixel in the output stack whose centre is nearest to that RA/DEC is given that input pixel. If `linear`, the exact pixel location of the RA/DEC on the output grid is calculated and the contribution from the input pixel is spread to the neighbouring 4 pixels that surround that point and is weighted by the fractional overlap.
`stk_seeing`	bool	**false**, true	if true, weight each image by seeing value when stacking
`stk_fast`	string	**auto**, fast, slow	See §9.1
`stk_nfst`	int	**16**, any int	See §9.1
`cacheloc`	string	**.**, any string	directory where cache of calibration catalogue data ('.catcache') is stored relative to current working directory (.)
`magerrcut`	float	**100.0**, any float	objects in the photometric reference catalogue with a magnitude error greater than this value will be excluded from the calibration

7.7.4 Product frames

PRO.CATG	Filename		Description
	Default	Pretty	
BASIC_CALIBRATED_SCI	exp_[N].fits	<FILE>_ex.fits	Calibrated image of each input OBJECT file
BASIC_VAR_MAP_SCI	exp_var_[N].fits	<FILE>_ex_var.fits	Variance map of each BASIC_CALIBRATED_SCI
OBJECT_CATALOGUE_SCI	exp_cat_[N].fits	<FILE>_ex_cat.fits	Catalogue of objects in each BASIC_CALIBRATED_SCI
MATCHSTD_ASTROM	exp_mstd_a_[N].fits	<FILE>_ex_mstd_a.fits	Matched astrometric standard catalogue for each BASIC_CALIBRATED_SCI
JITTERED_IMAGE_SCI	stack_1.fits	<FIRSTFILE>_st.fits	Calibrated stack of all OBJECT files
CONFIDENCE_MAP_SCI	stack_conf_1.fits	<FIRSTFILE>_st_conf.fits	Confidence map of JITTERED_IMAGE_SCI
JITTERED_VAR_IMAGE	stack_var_1.fits	<FIRSTFILE>_st_var.fits	Variance map of JITTERED_IMAGE_SCI
OBJECT_CATALOGUE_SCI_JITTERED	stack_cat_1.fits	<FIRSTFILE>_st_cat.fits	Catalogue of objects in JITTERED_IMAGE_SCI
MATCHSTD_ASTROM	stack_mstd_a_1.fits	<FIRSTFILE>_st_mstd_a.fits	Matched astrometric standard catalogue for JITTERED_IMAGE_SCI

continued on next page

PRO.CATG	Filename		Description
	Default	Pretty	
MATCHSTD_PHOTOM	stack_mstd_p_1.fits	<FIRSTFILE>_st_mstd_p.fits	Matched photometric standard catalogue for JITTERED_IMAGE_SCI

The index number [N] reflects the chronological order of the relevant files in the SoF, e.g. the one in the group with the earliest MJD-OBS has [N] = 1.

<FILE> or <FIRSTFILE> is the root name of the file in the SoF with chip = 'BRIAN' among the collection of files used to create that product, e.g. VIMOS.2011-06-21T08:40:45.959. If 'BRIAN' is missing, the next chip (alphanumerically) is used.

7.7.5 Quality control parameters

The following QC parameters are found in each extension of the `BASIC_CALIBRATED_SCI`, `CONFIDENCE_MAP_SCI`, and `JITTERED_VAR_IMAGE` files:

`QC.WCS_DCRVAL1`: [deg] change in crval1

`QC.WCS_DCRVAL2`: [deg] change in crval2

`QC.WCS_DTHETA`: [deg] change in rotation

`QC.WCS_SCALE`: [arcsec] mean plate scale

`QC.WCS_SHEAR`: [deg] abs(xrot) - abs(yrot)

`QC.WCS_RMS`: [arcsec] Average error in WCS fit

The following QC parameters are found in each extension of the `BASIC_VAR_MAP_SCI` files:

`QC.WCS_DCRVAL1`: [deg] change in crval1

`QC.WCS_DCRVAL2`: [deg] change in crval2

The following QC parameters are found in each extension of the `JITTERED_IMAGE_SCI` files:

`QC.WCS_DCRVAL1`: [deg] change in crval1

`QC.WCS_DCRVAL2`: [deg] change in crval2

`QC.WCS_DTHETA`: [deg] change in rotation

`QC.WCS_SCALE`: [arcsec] mean plate scale

`QC.WCS_SHEAR`: [deg] abs(xrot) - abs(yrot)

`QC.WCS_RMS`: [arcsec] Average error in WCS fit

`QC.MAGZPT`: [mag] photometric zeropoint

`QC.MAGZERR`: [mag] photometric zeropoint error (1.48 * median absolute deviation of MAGZPT among all extensions). Set to 1.0 if photometric calibration failed.

`QC.LIMITING_MAG`: [mag] 5 sigma limiting mag.

`QC.SKYBRIGHT`: [mag/arcsec**2] sky brightness

`QC.MAGNZPT`: number of stars available for magzpt calc; that actual number used may be less than this, depending on the value of `magerrcut`

The following QC parameters are found in each extension of the `OBJECT_CATALOGUE_SCI` and some `MATCHSTD_ASTROM` (from BASIC) files:

`QC.SATURATION`: [adu] Saturation level

`QC.MEAN_SKY`: [adu] Median sky brightness

`QC.SKY_NOISE`: [adu] Pixel noise at sky level

`QC.IMAGE_SIZE`: [arcsec] Average FWHM of stellar object

`QC.ELLIPTICITY`: Average stellar ellipticity (1-b/a)

`QC.POSANG`: [degrees] Median position angle (from North)

`QC.APERTURE_CORR`: Stellar ap-corr 1x core flux

`QC.NOISE_OBJ`: Number of noise objects

The following QC parameters are found in each extension of the `OBJECT_CATALOGUE_SCI_JITTERED`, `MATCHSTD_PHOTOM` files:

`QC.WCS_DCRVAL1`: [deg] change in crval1

`QC.WCS_DCRVAL2`: [deg] change in crval2

`QC.WCS_DTHETA`: [deg] change in rotation

`QC.WCS_SCALE`: [arcsec] mean plate scale

`QC.WCS_SHEAR`: [deg] abs(xrot) - abs(yrot)

`QC.WCS_RMS`: [arcsec] Average error in WCS fit

`QC.SATURATION`: [adu] Saturation level

`QC.MEAN_SKY`: [adu] Median sky brightness

`QC.SKY_NOISE`: [adu] Pixel noise at sky level

`QC.IMAGE_SIZE`: [arcsec] Average FWHM of stellar object

`QC.ELLIPTICITY`: Average stellar ellipticity (1-b/a)

`QC.POSANG`: [degrees] Median position angle (from North)

`QC.APERTURE_CORR`: Stellar ap-corr 1x core flux

`QC.NOISE_OBJ`: Number of noise objects

`QC.MAGZPT`: [mag] photometric zeropoint

`QC.MAGZERR`: [mag] photometric zeropoint error (1.48 * median absolute deviation of MAGZPT among all extensions). Set to 1.0 if photometric calibration failed.

`QC.LIMITING_MAG`: [mag] 5 sigma limiting mag.

`QC.SKYBRIGHT`: [mag/arcsec**2] sky brightness

`QC.MAGNZPT`: number of stars available for magzpt calc; that actual number used may be less than this, depending on the value of `magerrcut`

The following QC parameters are found in each extension of other `MATCHSTD_ASTROM` (from the stacks) files:

`QC.WCS_DCRVAL1`: [deg] change in crval1

`QC.WCS_DCRVAL2`: [deg] change in crval2

`QC.WCS_DTHETA`: [deg] change in rotation

`QC.WCS_SCALE`: [arcsec] mean plate scale

`QC.WCS_SHEAR`: [deg] abs(xrot) - abs(yrot)

`QC.WCS_RMS`: [arcsec] Average error in WCS fit

`QC.SATURATION`: [adu] Saturation level

`QC.MEAN_SKY`: [adu] Median sky brightness

`QC.SKY_NOISE`: [adu] Pixel noise at sky level

`QC.IMAGE_SIZE`: [arcsec] Average FWHM of stellar object

`QC.ELLIPTICITY`: Average stellar ellipticity (1-b/a)

`QC.POSANG`: [degrees] Median position angle (from North)

`QC.APERTURE_CORR`: Stellar ap-corr 1x core flux

`QC.NOISE_OBJ`: Number of noise objects

7.8 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 7.1.

DO category	Type	Explanation	Required
DETECTOR_PROPERTIES	Raw frame	Flat field exposure	√
MASTER_BIAS	Calibration	Master bias	√

Table 7.1: *Input files for the vmdet recipe.*

The products of the *vmdet* recipe are shown in Table 7.2. 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.fits	CCD_TABLE	FITS	Bad pixel table
bad_pixel_map.fits		FITS	Bad pixel image
error_image.fits		FITS	Error image

Table 7.2: *Products of the vmdet recipe.*

The *vmdet* parameters are listed in Table 7.3.

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 7.3: *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 9.17, page 176.

7.9 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 7.4.

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

Table 7.4: *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 in the case of imaging data, and the quadrant number increased by 4 in the case of spectral data). Care should be taken in selecting the appropriate bad pixel tables for imaging and spectral instrument modes.

The only product of the *vmbias* recipe is the master bias, as shown in Table 7.5. This is a MEF FITS file which contains one extension with the data and a second extension with the propagated estimated error.

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

Table 7.5: *Product of the vmbias recipe.*

The *vmbias* parameters are described in Table 7.6.

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

Parameter	Possible values	Default value	Explanation
AllowSingleFrames	true false	true	A single input bias is also allowed More than one input bias is required
StackMethod	Average Median MinMax Ksigma Auto	Median	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*	1	No. of lowest rejected values for rejection method
MaxRejection	*int*	1	No. of highest rejected values for rejection method
KSigmaLow	*float* (sigma)	5.0	Low threshold for K-sigma clipping method
KSigmaHigh	*float* (sigma)	5.0	High threshold for K-sigma clipping method
RemoveOverscan	true false	true	Remove overscan regions from master bias Keep overscan regions in master bias
CleanBadPixel	true false	false	Interpolate bad pixels on master bias No bad pixel correction
CleanCosmic	true false	false	Remove cosmic ray events from each bias No cosmic ray removal
ComputeQC	true false	true	Compute QC parameters Do not compute QC parameters

Table 7.6: *vmbias parameters.*

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 7.4). The bad pixel correction algorithm is described in Section 9.8, page 167.

CleanCosmic: Cosmic ray events removal from each input bias. The cosmic ray rejection algorithm is described in Section 9.9, page 168.

ComputeQC: If this parameter is set, *Quality Control* (QC) parameters will be computed and written to the header of the output master bias. Currently the QC parameters computed by *vmbias* are:

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

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

`QC BIAS RMS`: The population standard deviation of the 1600x1800 central pixels of the first input bias after overscan subtraction (if parameter *--RemoveOverscan* is set to TRUE.

`QC DET OUT? RON`: Readout noise for each detector port computed from the pre/overscan region as the robust standard deviation. See 9.10 for details.

`QC RON`: Readout noise for the whole detector computed as the average readout noise over all detector ports.

`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 9.12 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 9.18, page 178.

7.10 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 7.7.

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

Table 7.7: *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 in the case of imaging data, and the quadrant number increased by 4 in the case of spectral data). Care should be taken in selecting the appropriate bad pixel tables for imaging and spectral instrument modes.

The only product of the *vmdark* recipe is the master dark, as shown in Table 7.8.

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

Table 7.8: *Product of the vmdark recipe.*

The *vmdark* parameters are listed in Table 7.9.

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.

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 7.9: *vmdark parameters.*

BiasMethod: Method for bias removal from the input dark frames. The bias removal procedure is described in some detail in Section 9.10. 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 7.7). The bad pixel correction algorithm is described in Section 9.8, page 167.

CleanCosmic: Cosmic ray events removal from each input dark. The cosmic ray rejection algorithm is described in Section 9.9, page 168.

ComputeQC: If this parameter is set, *Quality Control* (QC) parameters will be computed and written to the header of the output master dark. 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 9.12 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 9.19, page 178.

7.11 vmmoscalib

This recipe identifies reference lines on MOS arc lamp exposures, and traces the spectral edges on the associated flat field exposures. With this information the spectral extraction mask to be applied in the scientific data reduction is determined. From the input flat field exposures a normalised flat field frame is also derived.

The input arc lamp and flat field exposures are assumed to be obtained quasi-simultaneously, so that they would be described by exactly the same optical and spectral distortions.

7.11.1 Input files

In alphabetical order:

CONFIG_TABLE: *optional* configuration table. This table defines a subset of recipe configuration parameters controlling the way spectra are extracted for any particular grism.

The configuration table consists of a single row of values labeled with the corresponding configuration parameters names. In the calibration directory, which is delivered together with the pipeline data reduction software, a standard configuration table is provided for each VIMOS grism and filter combination.

If a configuration table is used, it will modify the parameters of a recipe with its new values, with the exception of those which are explicitly specified on the command line. Without a configuration table, the input recipe parameters values will just be read from the command line, or from an *esorex* configuration file if present, or from their generic default values (that are rarely meaningful). The configuration parameters included in the configuration table are the following:

--dispersion	rough expected spectral dispersion in Å/pix
--peakdetection	threshold for preliminary peak detection
--wdegree	polynomial degree for wavelength calibration
--cdegree	polynomial degree for spatial curvature
--startwavelength	start wavelength for spectral extraction
--endwavelength	end wavelength for spectral extraction
--reference	reference wavelength
--resp_use_flat_sed	whether to use the flat sed normalisation for the response
--resp_fit_degree	degree to use in the polynomial fit of the response
--resp_fit_nknots	number of nknots to use in the spline fit of the response

A complete description of these parameters is given in Section 7.11.3, page 116.

LINE_CATALOG: *required* line catalog. It must contain the reference wavelengths (in Ångstrom) for the arc lamp used. A standard line catalog is also provided for each VIMOS grism in the calibration directory delivered with the pipeline software.

MASTER_BIAS: *required* master bias frame. Just one should be given.

MOS_ARC_SPECTRUM: *required* raw arc lamp spectrum exposure. Just one frame should be specified or the recipe would fail.

MOS_SCREEN_FLAT: *required* raw spectral screen flat exposure. If more than one is provided, the input frames are stacked into one.

7.11.2 Output files

Not all output frames listed here are always produced. Some of them are created only on request (see Section 7.11.3, page 116), and some other are never created in case of long slit data.[8]

[8] Long slit data are obtained when all the mask slits have the same spatial offset; this kind of mask is used in the acquisition of standard star spectra.

In the presence of spectral multiplexing (typically used with the low resolution grisms LR_red and LR_blue, and sometimes with the MR one), many products will be multi-extension FITS files, where each extension refers to a different group of spectra on the CCD. Each group is defined as the largest possible group including spectra which are not spectrally multiplexed with each other. In this way all groups can be reduced separately, as if no spectral multiplexing is present, applying the standard data reduction algorithm.

This is not just a nice application of software reusability and modularity: it is a necessity. It is clear, for instance, that with spectrally multiplexed data it is not possible to produce a single wavelength or spatial map of the CCD, since spectra will overlap. In other words, the same pixel may have different wavelengths and spatial coordinates, depending on what spectrum one is referring to.

Here is the list of all the possible output frames, in alphabetical order, together with a list of related configuration parameters.[9] Whenever a product, in case of spectrally multiplexed data, would include more than one data section, it will be indicated in the following sections with the acronym *MEF* (Multi Extension FITS). If the file also contains further extensions with the associated error, the acronym *ERR* is indicated.

GLOBAL_DISTORTION_TABLE: table containing the modeling of the coefficients of the local distortion models listed in the MOS_DISP_COEFF and the MOS_CURV_COEFF tables. It is produced only if the configuration parameter *--slit_ident* is set, and at least 6 spectra are found on the CCD. This table is currently used for quality control, and to support the on-line quick-look scientific data reduction. See Section 8.3.1 for more details. Note that for multiplexing data this cannot be switched off.

MOS_ARC_SPECTRUM_EXTRACTED: (MEF) rectified and wavelength calibrated arc lamp image (see Figure 7.1).

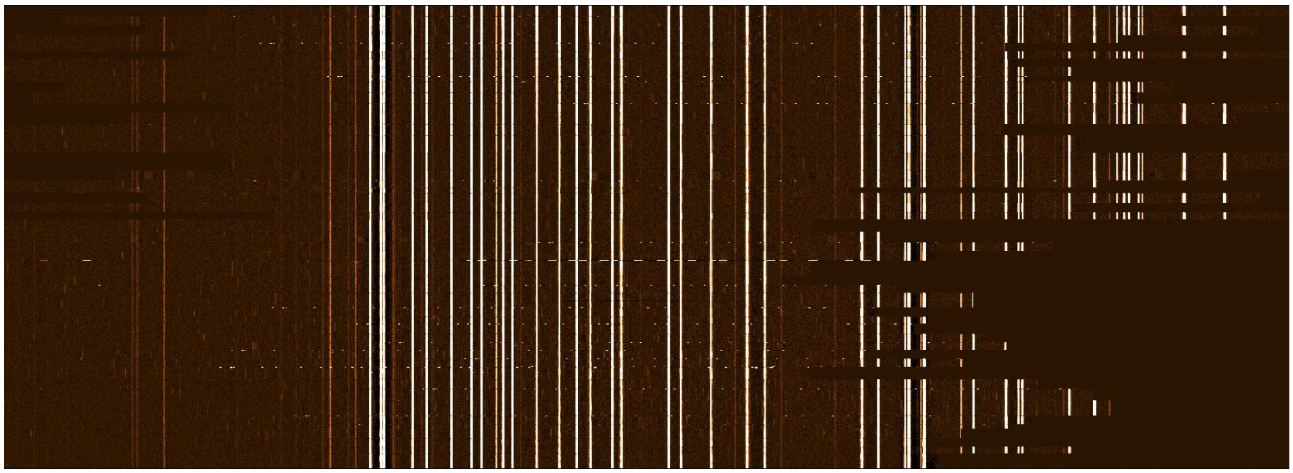

Figure 7.1: *MOS_ARC_SPECTRUM_EXTRACTED from a VIMOS HR_orange arc lamp exposure.*

This is the result of applying the extraction mask derived from the flat field and arc lamp exposures to the input arc lamp exposure itself. This image is just useful to get an immediate feeling of the goodness of the computed extraction mask. Note that this image is also bias and background subtracted. Here the background is estimated by using a minimun filter to each of the rows of the arc and later applying a

[9] See Section 7.11.3, page 116, for a complete description of the recipe configuration parameters.

a smoothing running box. Its x size depends on the spectral extraction range $(\lambda_{min}, \lambda_{max})$ and on the value used for the dispersion in wavelength units per pixel, D, defined by the configuration parameter *--dispersion* (see Section 7.11.3, page 116):

$$N_x = floor\Big(\frac{\lambda_{max} - \lambda_{min}}{D}\Big)$$

The y size of this image is equal to the total number of spatially rectified pixels: each slit spectrum is extracted between the traces of its left and right edges (see products MOS_CURV_TRACES and MOS_CURV_COEFF), and spatially remapped into a constant number of pixels at each y CCD coordinate. The number of rectified pixels for the i-th slit spectrum is computed as

$$N_i = ceil(t_i - b_i) + 1$$

where t_i and b_i are the x CCD coordinates of the i-th slit spectrum edges at the position of the grism central wavelength.[10] N_i is increased by 1 to ensure a slight oversampling of the original signal.[11] The total y size of the image is then given by

$$N_y = \sum_{i=0}^{n} N_i$$

where n is the number of extracted slit spectra. The slit spectra are ordered from top to bottom as they appear on the CCD from left to right. The lower bound of the slit in all the products which have been spatially resampled is record in the column *position* of table MOS_SLIT_LOCATION , while its length is record in column *length* The wavelength of each image pixel can be computed using the CRPIX1, CRVAL1 and CDELT1 FITS keywords:

$$\lambda = CDELT1 \cdot (x - CRPIX1) + CRVAL1$$

where x is the pixel number counted from left starting from 1.

Note that resampling the original spectrum at a constant wavelength step introduces distortions of the signal, that depend on the original signal pixelisation on the CCD. This is a side-effect of interpolation, that tends to systematically bias the interpolated value according to the position of the interpolation point with respect to the original CCD pixels and their values.[12] A detailed analysis of a scientific signal should be based on the unrebinned data matched with the corresponding wavelength map – see entry MOS_WAVELENGTH_MAP.

Configuration parameters directly affecting this product are *--startwavelength* and *--endwavelength*.

Configuration parameters having significant impact are *--dispersion*, *--peakdetection*, *--wradius*, *--wdegree*, *--wmode* and *--wmodemos*.

MOS_COMBINED_SCREEN_FLAT: combined flat field image. It is the bias subtracted sum of all the input screen flat fields.

[10] They correspond to the coefficients $c0$ of the MOS_CURV_COEFF table, or to $xtop$ and $xbottom$ in the MOS_SLIT_LOCATION table.

[11] This introduces a negligible correlation, but it ensures no loss of information in the mapping.

[12] No matter what interpolation method or kernel is chosen, this will always happen, unless the signal to resample is very well known in advance (which makes the interpolation pointless anyway): this would allow a perfect resampling of arc lamp spectra, for instance, but would not be applicable to scientific spectra.

MOS_CURV_COEFF: (MEF) table containing the coefficients of the spatial curvature fitting polynomials. The table columns are the following:

slit_id: Slit identification number (see the MOS_SLIT_LOCATION entry for a definition of the *slit_id*). Each identification appears twice, in consecutive rows: the top row refers to the left flat field spectrum edge, the bottom row to its right edge.

c0, **c1**, **c2**, ...: Curvature coefficients, depending on the degree of the fitting polynomial.

Configuration parameters directly affecting this product are *--cdegree* and *--cmode*.

Configuration parameters having significant impact are *--startwavelength* and *--endwavelength*.

MOS_CURV_TRACES: (MEF) table containing the x CCD positions of the detected spectral edges at different y CCD positions. The table columns are the following:

x: y CCD positions.

t<slit_id>: x CCD positions of the flat spectrum left edge from slit *slit_id* (for the definition of *slit_id* see the MOS_SLIT_LOCATION entry).

b<slit_id>: x CCD positions of the flat spectrum right edge from slit *slit_id*.

t<slit_id>**_mod**: Modeling of the flat spectrum left edge from slit *slit_id*.

b<slit_id>**_mod**: Modeling of the flat spectrum right edge from slit *slit_id*.

t<slit_id>**_res**: Residuals of curvature fit of the flat spectrum left edge from slit *slit_id*.

b<slit_id>**_res**: Residuals of curvature fit of the flat spectrum right edge from slit *slit_id*.

The traces of some edges may be missing because tracing is not always possible between spectra which are very close to each other. This does not prevent the final extraction of all the spectra, if a global spatial curvature model is applied by setting the configuration parameter *--cmode* > 0: but residuals cannot be evaluated in this case.

Note that in case of confusion between nearby spectra, where the exact position of the transition line between one spectrum and the other can be ambiguous, the position of the edge ideally traced by the global curvature model might not exactly correspond to the true (and not observable) spectral edge. It should be understood, however, that the aim of the computed model is primarily the elimination of the spatial curvature, and this can be obtained without knowing the absolute positions of the traces. In summary, observing extracted spectra[13] that include signal from other spectra and/or extending beyond their true spatial extension, does not imply that the spatial curvature was not properly removed. As a matter of fact nearby spectra *do* sometimes contaminate each other physically, by actually mixing their signals (case of crossing edges). In case of doubt, the extracted spectra should be carefully examined and compared with the corresponding original spectra found in the CCD exposure, in order to set the configuration parameter *--cmode* as appropriately as possible.

The only real solution to this problem would be to design masks where spectra are always well separated from each other (a buffer zone of 3 or 4 pixels would be sufficient).[14]

[13] See entry MOS_ARC_SPECTRUM_EXTRACTED in this Section, or entries MOS_SCIENCE_EXTRACTED and MOS_SCIENCE_SKY_EXTRACTED on page 130.

[14] It may be pointed out that this problem would "easily" be solved by applying an accurate physical model of the instrument. This however would be possible only under the assumption of a perfectly stable instrument, a dream that – together with the availability of a very accurate physical model – remains much too often unfulfilled.

Configuration parameters directly affecting this product are *--cdegree* and *--cmode*.

Configuration parameters having significant impact are *--startwavelength* and *--endwavelength*.

MOS_DELTA_IMAGE: (MEF) deviation from the linear term of the wavelength calibration polynomials. This image is used together with the MOS_DISP_RESIDUALS_TABLE to enable quality control of the obtained solutions (see Figure 7.2).

Configuration parameters directly affecting this product are *--startwavelength* and *--endwavelength*.

Configuration parameters having significant impact are *--dispersion*, *--peakdetection*, *--wradius*, and *--wdegree*.

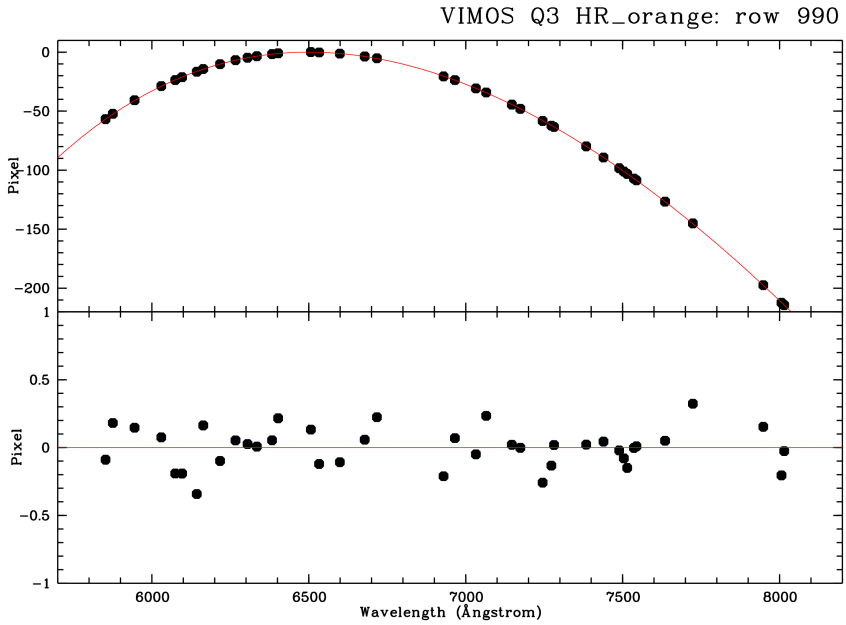

Figure 7.2: *Top panel: deviation of the identified peaks from the linear term of the 990th fitting polynomial (column d990 of the MOS_DISP_RESIDUALS_TABLE). The solid line is the polynomial model with the linear term subtracted, drawn from row 990 of the MOS_DELTA_IMAGE product. Bottom panel: fit residuals of the identified peaks (column r990 of the MOS_DISP_RESIDUALS_TABLE, identical to the residuals recorded at row 990 of the MOS_DISP_RESIDUALS image)..*

MOS_DISP_COEFF: (MEF) table containing the wavelength calibration polynomial coefficients. This table contains as many rows as in the MOS_ARC_SPECTRUM_EXTRACTED image, ordered in the same way. The table columns are the following:

c0, **c1**, **c2**, ...: Model coefficients, depending on the degree of the fitting polynomial.
nlines: Number of identified reference lines used in the fit.
error: Model mean accuracy computed from the observed fit residuals, keeping into account the number of model free parameters and the number of available reference lines:

$$\sigma = \sigma_{res}\sqrt{\frac{(n+1)}{N}}$$

where σ_{res} is the standard deviation of the residuals, n the polynomial degree, and N the total number of reference lines used in the fit. This evaluation of the model accuracy makes sense only in the absence of systematic trends in the residuals shown in the MOS_DISP_RESIDUALS image, and only under the assumption that data are not overfitted (i.e., the degree of the fitting polynomial is not higher than necessary, or practically speaking is the lowest capable of eliminating systematic trends in the residuals). Typical values of the model accuracy range between 0.05 and 0.1 pixels, as directly confirmed by Montecarlo simulations.

Configuration parameters directly affecting this product are *--wdegree* and *--wmodemos*.

Configuration parameters having significant impact are *--dispersion*, *--peakdetection*, *--wradius*, *--wreject*, *--startwavelength* and *--endwavelength*.

MOS_DISP_RESIDUALS: (MEF) residuals of each wavelength calibration fit (in pixels). This image is only created if the *--check* configuration parameter is set. The residuals of the derived wavelength calibration with respect to the measured pixel positions of the reference arc lamp lines are collected in this image, with x pixels corresponding to the original CCD y pixels, and y pixels corresponding to the MOS_ARC_SPECTRUM_EXTRACTED pixels (i.e., to the rectified spatial coordinate, see figure 7.3). Typical observed residuals should be around 0.2 pixels.[15] Note that all residuals are shown, including those from lines that were excluded from the polynomial fit, i.e., residuals larger than the threshold specified with the configuration parameter *--wreject* (see Section 7.11.3, page 116).

Configuration parameters directly affecting this product are *--startwavelength* and *--endwavelength*.

Configuration parameters having significant impact are *--dispersion*, *--peakdetection*, *--wradius*, and *--wdegree*.

MOS_DISP_RESIDUALS_TABLE: (MEF) table containing different kinds of residuals of a sample of wavelength calibration fits. Note that all residuals are shown, including those from lines that were excluded from the polynomial fit, i.e., residuals larger than the threshold specified with the configuration parameter *--wreject* (see Section 7.11.3, page 116). Just one every 10 of the polynomial fits listed in the MOS_DISP_COEFF table are examined. For an overview of all the polynomial fits residuals see the MOS_DISP_RESIDUALS image.

The residuals table columns are the following:

wavelength: Wavelengths of the reference lines (see entry LINE_CATALOG).

r<*row*>: Fit residuals of the identified peaks (in CCD pixel). *row* is the number of the examined row of the MOS_DISP_COEFF table.

d<*row*>: Deviation of the identified peaks from the linear term of the fitting polynomial (in CCD pixel). This can be compared with the corresponding row of the MOS_DELTA_IMAGE product (see Figure 7.2).

p<*row*>: x pixel position of reference lines on CCD.

[15] This is the accuracy of a single peak position measurement, *not* the accuracy of the model.

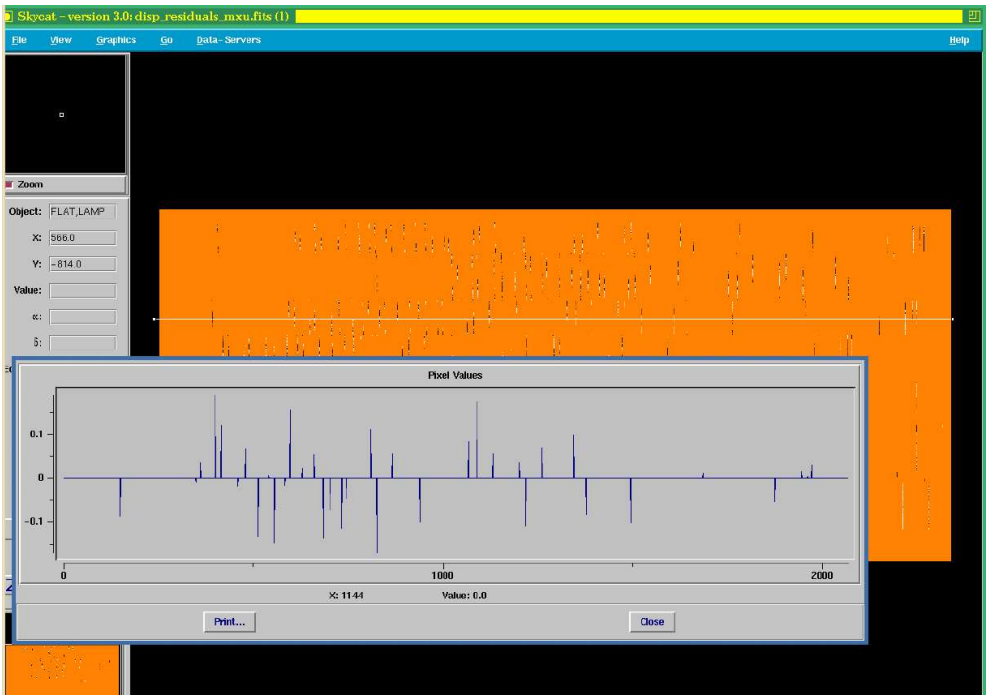

Figure 7.3: *MOS_DISP_RESIDUAL from an arc lamp calibration. In the foreground is a plot of the residuals from one image row.*

Configuration parameters directly affecting this product are *--startwavelength* and *--endwavelength*.

Configuration parameters having significant impact are *--dispersion*, *--peakdetection*, *--wradius*, and *--wdegree*.

MOS_MASTER_SCREEN_FLAT: (MEF, ERR) normalised flat field image, derived dividing the master screen flat by its smoothed version (see the smoothing configuration parameters description in Section 7.11.3, page 116). Comparing this image (or its extensions) with the MOS_COMBINED_SCREEN_FLAT may give an immediate feeling of the goodness of the computed curvature model used for the extraction of the normalised spectra.

Configuration parameters directly affecting this product are *--s_degree*, *--d_nknots*, *--sradius*, *--normalise_spa_local*, *--dradius*, *--startwavelength* and *--endwavelength*.

Configuration parameters having significant impact are *--cdegree* and *--cmode*.

MOS_SLIT_LOCATION: (MEF) slit positions, both on the CCD and on the rectified image of the arc lamp exposure (MOS_ARC_SPECTRUM_EXTRACTED). The slits are listed from left to right, according to their x position on the CCD, and they are identified by a *slit_id* number. The *slit_id* is read from the FITS header of the input data. For example, the identification number of the 46-th slit written to header is given by the keyword ESO.INS.SLIT46.ID. Note that the *slit_id* is unrelated to the top–bottom ordering of the spectra on the CCD (rather referring to the way the mask was manufactured).

The slits location table columns are the following:

slit_id: Slit identification number.
xtop: x CCD position of central wavelength from left end of slit.
ytop: y CCD position of central wavelength from left end of slit.
xbottom: x CCD position of central wavelength from right end of slit.
ybottom: y CCD position of central wavelength from right end of slit.
position: First row of MOS_ARC_SPECTRUM_EXTRACTED image containing the rectified slit spectrum bottom row. Image rows are counted from bottom, starting from 0.
length: Number of rows in MOS_ARC_SPECTRUM_EXTRACTED image including the slit spectrum.

If the slit identification task is not run (see configuration parameter *--slit_ident*, Section 7.11.3, page 116), or if the slit identification task fails (e.g., in the case of just two slits) the *slit_id* is set to the slit sequence number in the left–right ordering of the spectra on the CCD: but in order to avoid confusion with tags assigned to identified slits, a negative integer is used in this case instead of a positive one.

MOS_SLIT_MAP: map of central wavelength on the CCD. This image is only created if the *--check* configuration parameter is set. It has the same size of the MOS_WAVELENGTH_MAP image, from which it is derived. This product can be seen as an image of the mask cast on the CCD (see step 5 in Section 9.20, page 179): the slits images on the CCD are compared with their positions on the mask, to derive the optical distortion model (see steps 6 and 7, always in Section 9.20).

Configuration parameters that may have some impact on this product are *--wdegree*, *--wmodemos*, *--dispersion*, *--peakdetection*, *--wradius*, and *--wreject*.

MOS_SPATIAL_MAP: (MEF) map of spatial positions on the CCD. It has the same size of the CCD, where each pixel has the value of its distance (in CCD pixels) from the left edge of the spectrum it belongs to (see Figure 7.4). In case of confusion between nearby spectra, the spatial coordinate would just reflect the spatial curvature, and not the absolute spatial coordinate along the slit: see the note to the MOS_CURV_TRACES entry in this Section for more details.

Configuration parameters directly affecting this product are *--cdegree* and *--cmode*.

Configuration parameters having significant impact are *--startwavelength* and *--endwavelength*.

MOS_SPECTRA_DETECTION: (MEF) result of the preliminary wavelength calibration applied to the arc lamp exposure. This image is only created if the *--check* configuration parameter is set, and only in case the spectra have not all the same spatial offset. The preliminary wavelength calibration is performed with the purpose of detecting and locating the spectra on the CCD (see step 2 in Section 9.20, page 179). In case of problems found in the recipe products, this image may be examined. All spectra should look aligned in wavelength, in particular around the central wavelength, that is the position used for constructing the slit map (MOS_SLIT_MAP). Gaps in the solution within a spectrum may appear, but if not overwhelming they have generally no consequences for the data reduction, because they are filled up consistently while creating the slit map. The x size of this image equals the x size of the MOS_ARC_SPECTRUM_EXTRACTED image, while its y size matches the x size of the CCD (i.e., no spatial remapping performed).

Configuration parameters directly affecting this product are *--dispersion*, *--peakdetection*, and *--wdegree*.

Configuration parameters having significant impact are *--startwavelength* and *--endwavelength*.

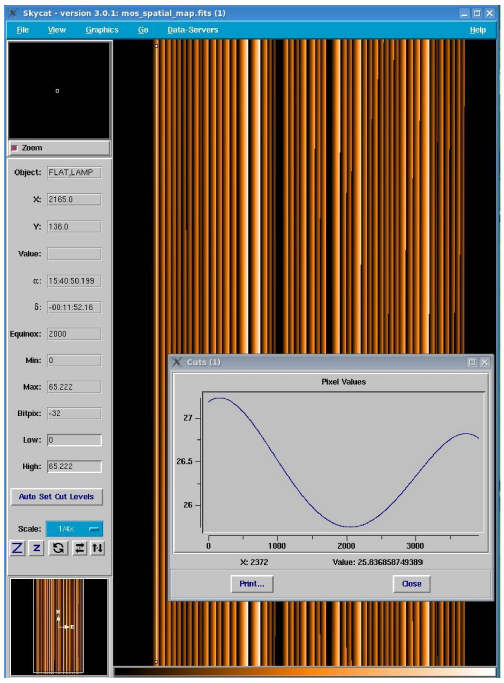

Figure 7.4: *MOS_SPATIAL_MAP from a VIMOS HR_orange flat field tracing, modeled with a 4th degree polynomial. In the foreground is a plot of the distances from the left spectral edge of all pixels from one CCD column (from the first spectrum on the left).*

MOS_SPECTRAL_RESOLUTION: (MEF) Mean spectral resolution for each reference arc lamp line. The table columns are the following:

wavelength: Wavelength of reference line.
fwhm: Mean FWHM of reference line.
fwhm_rms: Standard deviation of all measured FWHM from all the CCD columns including the line.
resolution: Mean spectral resolution, measured as the line *wavelength*, divided by its FWHM.
resolution_rms: Standard deviation of all the measured spectral resolutions from all the CCD columns including the line.

MOS_WAVELENGTH_MAP: (MEF) map of wavelengths on the CCD. This image has the same size of the CCD, where each pixel has the value of the wavelength at its center, if available.

Configuration parameters directly affecting this product are *--startwavelength* and *--endwavelength*.

Configuration parameters having significant impact are *--dispersion*, *--peakdetection*, *--wradius*, and *--wdegree*.

MOS_FLAT_SED: Image containing the spectral energy distribution of the flat. Each image row corresponds to the SED of one slit. The slit order follows that of the MOS_SLIT_LOCATION table.

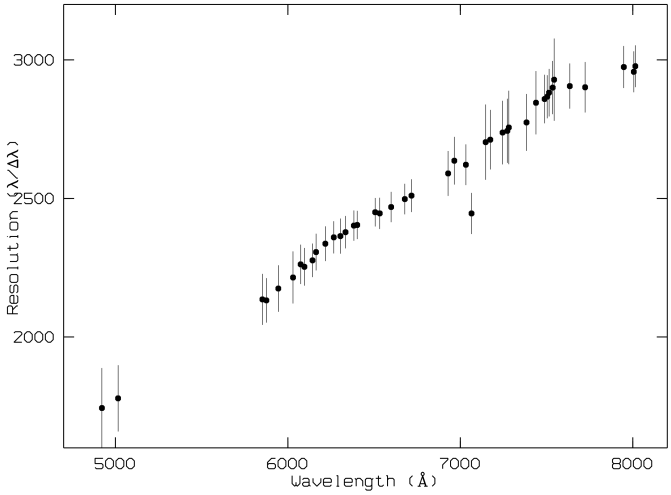

Figure 7.5: *Resolution vs. wavelength in a MOS_SPECTRAL_RESOLUTION table derived from a VIMOS HR_orange arc lamp exposure.*

7.11.3 Configuration parameters

The configuration parameters setting determines the way the *vmmoscalib* recipe will process the input data, and to some extent the product files that will be created. The parameters are conveniently divided into four main sections: wavelength calibration, spatial curvature calibration, flat field normalisation, and quality control.

Wavelength calibration

--dispersion: Expected spectral dispersion. *Default*: 0.0 Å/pixel

This parameter is mandatory (using the default 0.0 would generate an error message). This is a rough value of the expected spectral dispersion, used by the pattern-recognition algorithm described in Section 9.20.2, page 182. The dispersion values listed in table 3.1, page 18, are good, but in exceptional cases they might be tuned for recovering possible failures of the data reduction procedure, or to improve the quality of unsatisfactory results. In general, however, the spectral detection algorithm is very robust to modifications of this parameter: as a typical example, with HR orange grism data, optimal results (at constant quality) are obtained within a 10% variation of the first-guess dispersion[16] (centered around a value which depends on temperature).

Optimal values for this parameter, depending on the applied grism, are included in the CONFIG_TABLEs (see Section 7.11.1, page 107).

--peakdetection: Initial peak detection threshold. *Default*: 0.0 ADU

This parameter is mandatory (using the default 0.0 would generate an error message). This is a threshold value used in the preliminary peak detection task (see Section 9.20.1, page 181): the reference lines

[16]This is not true for all kinds of data: This tolerance mostly depends on the dispersion law of the applied grism, and on the distribution of the available arc lamp lines.

candidates are selected from peaks having a maximum value *above the background* higher than this threshold. Weaker entries of the input line catalog are recovered later on, after the preliminary wavelength calibration is obtained, if the parameter *--wradius* is set to a value greater than zero. It is however crucial that most of the reference lines are already detected at the earliest stage, if the pattern-recognition is meant to give the best possible results. A threshold value of 250 ADU is suitable in most cases, but sometimes the recovery of fainter reference lines may require to lower the threshold almost down to noise level.[17]

Optimal values for this parameter, depending on the applied grism, are included in the CONFIG_TABLEs (see Section 7.11.1, page 107).

--startwavelength: Start wavelength in spectral extraction. *Default*: 0.0 Ångstrom

See the *--endwavelength* parameter.

--endwavelength: End wavelength in spectral extraction. *Default*: 0.0 Ångstrom

This parameter, together with the *--startwavelength* parameter, defines the wavelength interval where calibration is attempted: this interval may not be entirely contained in the CCD for all spectra. Default values of the extraction interval, depending on the applied grism, are included in the CONFIG_TABLEs (see Section 7.11.1, page 107). If both *--startwavelength* and *--endwavelength* are left to 0.0, the extraction interval is computed automatically as the interval between the first and the last identified arc lamp reference lines, extrapolated by 10% at its blue and red ends (see Section 9.20.3, page 185).

--reference: Reference wavelength for calibration. *Default*: 0.0 Ångstrom

This parameter is mandatory (using the default 0 would generate an error message). This is the reference wavelength used in the determination of the inverse dispersion solution. Default values for the reference wavelength, depending on the applied grism, are included in the CONFIG_TABLEs (see Section 7.11.1, page 107).

The reference wavelength should lie within the interval defined by *--startwavelength* and *--endwavelength*. In principle, changing the reference wavelength shouldn't have a big impact in the overall calibration, since it just means that the wavelength calibration is refered to one position or another. However, the slit identification step uses a pattern matching algorithm that compares the nominal positions of the slits as found in the header with the detected slits at the reference wavelength. A change in *--reference* can then lead to slightly different detected positions and therefore the pattern matching algorithm can fail to properly identify the slits. If it is seen that *--reference* affects greatly the calibration, it is recommended to use a value within a region with good wavelength calibration, for instance within a cluster of lines in the line catalogue.

The reference wavelength should lie within the interval defined by *--startwavelength* and *--endwavelength*. In principle, changing the reference wavelength shouldn't have a big impact in the overall calibration, since it just means that the wavelength calibration is refered to one position or another. However, the slit identification step uses a pattern matching algorithm that compares the nominal positions of the slits as found in the header with the detected slits at the reference wavelength. A change in *--reference* can then lead to slightly different detected positions and therefore the pattern matching algorithm can fail to properly identify the slits. If it is seen that *--reference* affects greatly the calibration, it is recommended to use a value within a region with good wavelength calibration, for instance within a cluster of lines in the line catalogue.

[17] Lowering this threshold below a 3-σ noise level would completely destroy the observed pattern. In such extreme cases a preliminary smoothing of the input arc lamp exposure for reducing the random noise may help.

--wdegree: Degree of wavelength calibration polynomial. *Default*: 0

This parameter is mandatory (using the default 0 would generate an error message). The degree used for the wavelength calibration polynomial should be the lowest that would provide non-systematic residuals to the solution (see the MOS_DISP_RESIDUALS entry, page 112).

Note that the *--wdegree* parameter should be more correctly intended as the *maximum* applicable polynomial order: the polynomial is really adapted to the number of identified arc lamp lines used in the fit. This is necessary, because spectra from slits with very high offsets on the telescope focal plane may not be entirely contained in the CCD, and several arc lamp reference lines might be unavailable for calibration. Such spectra would not be properly calibrated if a polynomial with too many free parameters were used. As a rule, a polynomial with the specified *--wdegree* is only used if the number of identified lines is at least twice the number of free parameters: if this were not the case, the applied polynomial order would be

$$n = floor\left(\frac{N}{2}\right) - 1$$

where N is the number of identified reference lines. Accordingly, no solution is computed if less than 4 reference lines are identified.

--wradius: Search radius, if iterating pattern-matching with first-guess method. *Default*: 4 pixel

If this parameter is greater than zero, the peak identification is iterated using the pattern-matching solution as a first-guess model: the wavelengths listed in the input line catalog are transformed to CCD pixel positions using the model, and a peak is searched within the specified search radius.[18] Alternatively, setting *--wradius* = 0 means to accept the pattern-matching solution without further processing. Iterating the solution makes the wavelength calibration more robust, and increasing the search radius may help sometimes to recover from a bad result. It may happen however that the pattern-matching solution is more accurate than the one based on the iteration: this is because in the pattern-matching task peaks are identified by their being part of a pattern, while with a first-guess model each peak is identified by its vicinity to its expected position: the latter approach may lead to occasional misidentifications, and may be more negatively affected by contamination and lines blending (see also Section 9.20.1, page 181).

--wreject: Rejection threshold in dispersion relation fit (pixel). *Default*: 0.7 pixel

The wavelength calibration polynomial fit is iterated excluding any reference line position displaying a residual greater than the specified threshold.

--wmodelss: Interpolation mode of wavelength solution (0 = no interpolation, 1 = fill gaps, 2 = global model). *Default*: 2

This parameter only affects the processing of long slit data.[19] Given the wide availability of similar information on a long slit spectrum, it is conceivable an improvement of the quality of the calibration by modeling the global trend of the local solutions obtained from each CCD column. If *--wmodelss* = 1 the global model is applied just to fill possible gaps in the solution, maintaining the result of the local calibrations where they are available. If *--wmodelss* = 2 the global model solution is used for replacing also the available local solutions. No interpolation is applied to the data if *--wmodelss* = 0.

[18] If a search radius greater than zero is specified, but the reference lines widths are even greater, the search radius is automatically set to the actual lines widths.

[19] Long slit data are obtained when all the mask slits have the same spatial offset; this kind of mask is used in the acquisition of standard star spectra.

--wmodemos: Interpolation mode of wavelength solution (0 = no interpolation, 1 = local (slit) solution, 2 = global model). *Default*: 1

This parameter only affects the processing of randomly distributed spectra (as opposed to long slit spectrum, which includes specifically the standard star acquisition masks, made of slits all at the same offset).

It is conceivable an improvement of the quality of the wavelength calibration by modeling the trend of the solutions within each slit, or even globally (as a function of the position of the slits on the focal plane). If *--wmodemos* = 1 the solutions within each slit are replaced by their best linear fit, while if *--wmodemos* = 2 a bivariate, second order global solution is fitted to the available local solutions and then replaces them. No interpolation is applied to the data if *--wmodemos* = 0.

--ignore_lines: Catalog lines nearest to wavelengths in this list will be ignored for wavelength calibration. *Default*: empty

This parameter contains a string with a comma separated list of lines to be ignore from the reference catalogue line. In fact it is not needed to specify the exact wavelength present in the catalogue, instead, the closest lines in the catalogue to each of the values of this parameter will be ignored for the wavelength calibration. For instance, a value of *4300,5400* will ignore lines 4358.343 and 5460.742 from the standard catalog.

--used_linesets: Linesets in the line catalog to use. *Default*: standard

The [LINE_CATALOG] catalog used to compute the wavelength calibration contains a column [LINE_SET] which defines whether the line belongs to the *standard* set or to the *extended* set. Using the standard set will provide good results in most of the cases, but in some cases the user can also specify to use additionally the extended set, which might contain weaker lines or doublets.

The parameter syntax is a comma separated list of the sets to use, for instance *standard,extended*

Spatial curvature calibration

--cdegree: Degree of spatial curvature polynomial. *Default*: 0

This parameter is mandatory (using the default 0 would generate an error message). Optimal values depending on the applied grism are included in the CONFIG_TABLEs (see Section 7.11.1, page 107).

Systematic residuals, oscillating from positive to negative offsets of about 0.2–0.3 pixels, are frequently observed, and are confirmed also by other data reduction systems (see Figures 7.6, 7.7). The systematic residuals are due to the changing pixelisation of the spectral edges on the CCD, and therefore they should not be considered physical. A low degree polynomial fit appropriately circumvents this effect by cutting through such oscillations. Trying to fit such residuals with higher degree polynomials would lead to unstable and unrealistic solutions.

--cmode: Interpolation mode of curvature solution (0 = no interpolation, 1 = fill gaps, 2 = global model). *Default*: 1

Using a global description of the spatial curvature helps to extract also those spectra whose edges cannot be traced because of confusion with nearby spectra. If *--cmode* = 0 the only recovery strategy consists in supplying a missing trace by replicating the trace of the opposite edge (opportunely shifted). This is however not very accurate, and it is not even applicable if a tracing is missing for both edges of a slit

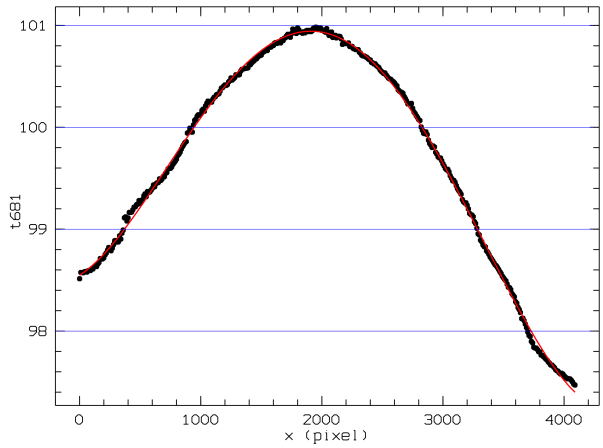

Figure 7.6: *Tracing of the curvature model (from a VIMOS HR_orange flat field exposure).*

spectrum.[20] By setting *--cmode* = 1 a global trend of the curvature coefficients would be determined, allowing to derive a curvature model also for the spectral edges that are lacking a direct tracing. Setting *--cmode* = 2 would recompute the curvature model also for the spectra where a local solution is available: this is generally not advisable, because a local solution is generally more accurate than the one derivable from the global solution.

--slit_ident: Attempt slit identification. *Default*: TRUE

Setting this parameter activates the 2D pattern-recognition task linking the slits positions on the mask with those on the CCD (see Section 9.20.6, page 186). In principle, the only outcome would be the identification of the detected spectra, i.e., their association to the slits on the mask, that is not required for a complete processing of the data: spectra would be extracted anyway, even if lacking a proper identification.[21] However, as shown in Section 9.20.6, the 2D pattern-recognition is also used to define an optical distortion model which helps to improve the accuracy of the preliminary spectra detection, and in some case even to allow the recovery of spectra that were lost to the spectral identification task. This is why the slit identification should always be requested: the only reason why the parameter *--slit_ident* was defined is to offer to possibility to switch the 2D pattern-matching task off in case this affected negatively the data reduction process.

One should keep in mind that the flux normalization of the MOS_FLAT_SED described at the end of Sect. 9.15 can be done properly only if the slits are identified as the width of each slit needs to be taken into account. With *slit_ident=false* this is not possible and a corresponding systematic error will affect the absolute flux scale of the flux-calibrated spectra.

For long slit data, like standard stars, the algorithm is not a 2D pattern matching but a simple linear matching.

[20] In this case the spectrum would not be extracted.

[21] In fact, the 2D pattern-recognition task would fail in case less than three spectra were detected on the CCD, and also in case the spectra were regularly spaced, as it happens with some calibration masks: but in neither situation spectra identification represents a practical issue.

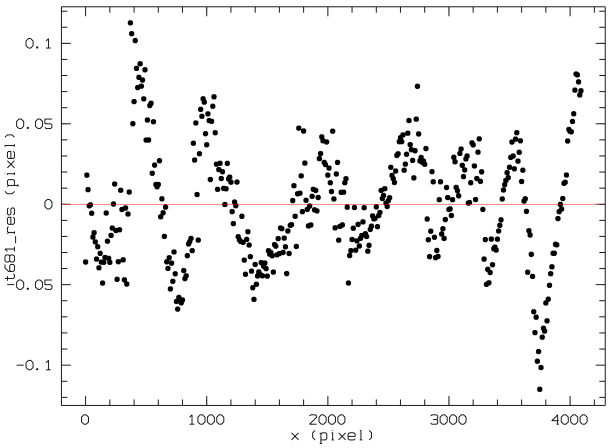

Figure 7.7: *Systematic residuals of curvature model (from a VIMOS HR_orange flat field exposure).*

For multiplexed data this parameter cannot be switched off. Note also that excluding the slit identification would also allow to reduce data from instruments different from VIMOS.[22]

Flat field normalisation

--s_degree: Degree of flat field fitting polynomial along spatial direction. *Default*: -1

If the configuration parameter *--s_degree* is set to a non-negative value, the master flat field normalisation is performed by modeling its large scale trend with a polynomial fitted along the spatial direction for each slit. See 9.11.1 for details of the normalisation algorithm.

--d_nknots: Number of knots used for the spline fitting along dispersion direction. *Default*: -1

If *--d_nknots* is set to a non-negative value, the master flat field normalisation is performed by modeling its large scale trend with a cubic spline polynomial fitted along the dispersion direction. The flat field spectra are spatially rectified applying the curvature model before the fit is performed, and the smoothed result is mapped back to the CCD frame before being used for normalising the master flat field. If *--d_nknots* < 0 the illumination trend is obtained instead by median filtering the spatially rectified spectra with a running box of sizes (see *--dradius*) on averaged flat field spectral profile.

--fit_threshold: Threshold percentage for flat spline fitting and polynomial fitting with respect to the maximum. *Default*: 0.01

The input pixels used for the cubic spline fiting and the polynomial fitting are first filtered to reject values below *--fit_threshold* times the maximum value in the spectrum of that row.

--dradius: Smooth box radius for flat field along dispersion direction. *Default*: 10 pixel

If it is not negative, a median smoothing with half width *--dradius* is applied along the dispersion direction. See also the *--sradius* parameter.

[22]The only reason why the self-calibrating procedure applied here is not readily usable for any MOS instrument is that the way the slit characteristics are listed in the data FITS headers is not standardised.

--sradius: Smooth box radius for flat field along spatial direction. *Default*: 10

If it is not negative, it is the size (in pixel) along the spatial direction of the half-width of the running box applied for smoothing the master flat field along spatial direction.

Setting *--sradius=-1* and *--sdegree=-1* will leave the spatial illumination gradient in the normalised master flat field, which is helpful for LSS data.

--normalise_spa_local: Normalise the spatial profile row-by-row *Default*: TRUE

If TRUE flat field normalization for the spatial profile is done row-by-row. Otherwise the spatial profile is calculated collapsing the slit image along the dispersion profile.

7.11.4 Quality control parameters

Currently the following QC parameters, used by PSO and DPD, are evaluated by the *vmmoscalib* recipe.

QC MOS SLIT WIDTH: Width of slit closest to mask centre. *Units*: mm

Most of the quality control parameters are evaluated on the slit whose location is closest to the mask center. The width of this slit would affect other parameters, such as the mean spectral resolution.

QC MOS FLAT FLUX: Flat field flux at reference wavelength. *Units*: ADU $s^{-1}mm^{-2}$

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.

QC MOS FLAT FLUXERR: Error on flat field flux at reference wavelength. *Units*: ADU $s^{-1}mm^{-2}$

The total counts in the rectangular region where the flux was determined are converted to electrons, they are square-rooted, converted back to ADU, and finally normalised to the unit of area and time.

QC MOS HE LAMBDA: Line for He arc lamp flux determination.

QC MOS NE LAMBDA: Line for Ne arc lamp flux determination.

QC MOS AR LAMBDA: Line for Ar arc lamp flux determination. *Units*: Ångstrom

These are the wavelengths of the Helium/Neon/Argon lines of the arc lamp spectrum that were used for the lamp monitoring.

QC MOS HE FLUX: Flux at chosen He wavelength.

QC MOS NE FLUX: Flux at chosen Ne wavelength.

QC MOS AR FLUX: Flux at chosen Ar wavelength. *Units*: ADU $mm^{-2}s^{-1}$

The CCD region containing the chosen arc lamp line in the slit closest to mask center is determined, and its signal integrated. The total counts are then bias corrected and divided by the area of the slit and by the exposure time.

QC MOS HE FLUXERR: Error on flux at chosen He wavelength.

QC MOS NE FLUXERR: Error on flux at chosen Ne wavelength.

QC MOS AR FLUXERR: Error on flux at chosen Ar wavelength. *Units*: ADU $mm^{-2}s^{-1}$

The total counts in the rectangular region where the flux is determined are converted to electrons, then are square-rooted, converted back to ADU, and finally normalized to the unit of area and time.

QC MOS WAVECAL COEFFi: Median of i-th coefficient of dispersion solution. *Units*: pixel Å^{-i}

This is the median of the i-th coefficient of the polynomial transformations from wavelength to y CCD pixel valid for each x (spatial) pixel position along the slit.

QC MOS RESOLUTION1 LAMBDA: Arc lamp line used in spectral resolution determination at the red end of spectrum.

QC MOS RESOLUTION2 LAMBDA: Arc lamp line used in spectral resolution determination at the center of spectrum.

QC MOS RESOLUTION3 LAMBDA: Arc lamp line used in spectral resolution determination at the blue end of spectrum. *Units*: Ångstrom

QC MOS RESOLUTION1: Mean spectral resolution at red end of spectrum.

QC MOS RESOLUTION2: Mean spectral resolution at center of spectrum.

QC MOS RESOLUTION3: Mean spectral resolution at blue end of spectrum. *Units*: *none*

QC MOS RESOLUTION1 RMS: RMS of spectral resolution at red end of spectrum.

QC MOS RESOLUTION2 RMS: RMS of spectral resolution at center of spectrum.

QC MOS RESOLUTION3 RMS: RMS of spectral resolution at blue end of spectrum. *Units*: *none*

QC MOS IDS RMS: RMS of global dispersion solution. *Units*: pixel

QC FLAT SEDi NORM: The normalisation factor used to get the flat SED for slit i. *Units*: ADUs

QC FLAT SED CORR_SLITWID: It is TRUE if the normalisation factor used to get the flat SED ([QC FLAT SEDi NORM]) was the actual width of the slit (available only when the slits are identified). It is FALSE if the median of the slit widths was used (in case of disabled or failed slit identification). *Units*: *none*

7.12 vmmosscience

This recipe is used for reducing VIMOS / MOS scientific spectra applying the extraction mask and the normalised flat field created by the recipe *vmmoscalib*. In case of a standard star observation, efficiency and response curves are also calculated.

The slit spectra are bias subtracted, flat fielded if requested, and remapped eliminating the optical distortions. The input wavelength calibration can optionally be adjusted to a number of reference sky lines. The sky spectrum can be modeled and subtracted choosing from three available methods. Finally, objects are searched and extracted from the slit spectra.

In case more than one scientific exposure is specified in input, each exposure is processed as above, but all of the cleaned up frames will be aligned to the first frame (in the case of dithered observations) and stacked together before attempting the object detection and extraction steps. After stacking, the airmass of the first scientific exposure is used to correct the extinction of the stacked spectra.

In the presence of spectral multiplexing (typically used with the low resolution grisms LR_red and LR_blue, and sometimes with the MR ones), most input and output data will be multi-extension FITS files, where each extension refers to a different group of spectra on the CCD. Each group is defined by a previous run of *vmmoscalib* (see Section 7.11, page 106) on the calibration exposures associated to the scientific observation. The spectra within each group are not spectrally multiplexed with each other. All groups can therefore be reduced separately, as if no spectral multiplexing is present, applying the standard data reduction algorithm.

This is not just a nice application of software reusability and modularity: it is a necessity. It is clear, for instance, that with spectrally multiplexed data it is not possible to produce a single wavelength or spatial map of the CCD, since spectra will overlap. In other words, the same pixel may have different wavelengths and spatial coordinates, depending on what spectrum one is referring to.

Here are the lists of all the input and output frames, each in alphabetical order. Whenever an input or a product, in case of spectrally multiplexed data, would include more than one FITS extension, it will be indicated in the following sections with the acronym *MEF* (Multi Extension FITS). If the file also contains further extensions with the associated error, the acronym *ERR* is indicated.

In the DO categories listed below the word SCIENCE is replaced by STANDARD, and SCI by STD, in the case of spectroscopic standard stars observations, i.e., whenever MOS_STANDARD frames are specified in input instead of MOS_SCIENCE frames.

7.12.1 Input files

In alphabetical order:

CONFIG_TABLE: *optional* configuration table. See Section 7.11.1, page 107.

EXTINCT_TABLE: *optional* atmospheric extinction table. If a standard star exposure is specified in input, and efficiency and response curves are wanted, then this table is *required*. It is also required in case a spectro-photometric calibration should be applied to the extracted spectra (i.e., when the a flux calibration table is present).

Currently an atmospheric extinction table valid for Paranal is made available in the calibration directories, in a file named `extinct_table.fits`. See Table 7.10 on page 124 for details about the format.

Column name	Explanation
WAVE	Wavelength at which the extinction was evaluated
EXTINCTION	Magnitude loss per one airmass

Table 7.10: *Atmospheric extinction table entries.*

MASTER_BIAS: (ERR) *required* master bias frame. Just one should be given.

MOS_CURV_COEFF: (MEF) *required* table with spatial curvature coefficients. This table is produced by the *vmmoscalib* recipe (see page 110).

MOS_DISP_COEFF: (MEF) *required* table with wavelength solution coefficients. This table is produced by the *vmmoscalib* recipe (see page 111).

MOS_MASTER_SCREEN_FLAT: (ERR) *optional* normalised flat field. This frame is produced by the *vmmoscalib* recipe (see page 113), and it must be provided only if the flat field correction is requested (see configuration parameter *--flatfield*, Section 7.12.3, page 133).

MOS_SCIENCE: *required* scientific exposure. One or more (possibly dithered) frames can be specified.

MOS_SLIT_LOCATION: *required* table of slits positions. This table is produced by the *vmmoscalib* recipe (see page 113).

SKY_LINE_CATALOG: *optional* sky lines catalog. It must contain the reference wavelengths (in Ångstrom) of the sky lines used for adjusting the input wavelength solution to the observed scientific spectra. The only requirement for this table is to contain a column with name "WLEN" listing such wavelengths. If the alignment of the wavelength solution to the sky lines is requested, and a SKY_LINE_CATALOG is not specified in input, an internal sky line catalog is used instead (see Table 7.12).

MOS_SPECPHOT_TABLE: *optional* table with efficiency and response curves. It must be specified in case a spectro-photometric calibration should be applied to the extracted spectra (i.e., when a flux calibration table is present). This table can also be a product of this recipe, and it is described in more detail in the next Section (page 131). See also Section 9.16, page 175, about how the photometric calibration is applied.

STD_FLUX_TABLE: *optional* standard star flux table. If a standard star exposure is specified in input, and efficiency and response curves are wanted, this table must also be specified in input. The table has the structure shown in table 7.11.

Column name	Explanation
WAVE	Wavelength at which the flux was evaluated
FLUX	Flux in erg cm^{-2} s^{-1} $Å^{-1}$
BIN	Bin width in Ångstrom
STLLR_ABSORP	Flag to specify if a bin is affected by stellar absorption

Table 7.11: *Standard star flux table.*

A set of standard star flux tables, corresponding to the 30 spectro-photometric standard stars that are included in the VIMOS calibration plan ([9]) is available in the calibration directories. The names of these tables, and the name of the standard stars as reported in the FITS header keyword ESO OBS TARG NAME, are listed in Table 7.13. The table indicated in the SOF should match the content of the header entry ESO OBS TARG NAME of the input standard star exposure.

MOS_FLAT_SED: *required* master flat spectral profile. If this is input, the extracted spectra will be divided by the master flat spectral profile corresponding to the spectra slit. For standard star observations this will change the final values of the response. It should be used for the science photometric correction if the response was computed with it.

Wavelength	Low resolution	Wavelength	Low resolution
5577.338	√	7329.148	
5889.953		7340.885	
5895.923		7358.659	
5915.301		7571.746	√
5932.862		7750.640	
5953.420		7759.996	
6257.961		7794.112	
6287.434		7808.467	
6300.304	√	7821.503	
6306.869		7841.266	
6363.780		7913.708	
6498.729		7949.204	
6533.044		7964.650	√
6553.617		7993.332	√
6841.945		8014.059	
6863.955	√	8310.719	
6870.994		8344.602	
6889.288		8382.392	
6900.833		8399.170	
6912.623		8415.231	
6923.220		8430.174	
6939.521		8452.250	
6969.930		8493.389	
7003.858		8791.186	
7244.907		8827.096	
7276.405		8885.850	
7284.439		8903.114	
7316.282		8943.395	
		8988.366	

Table 7.12: *Default sky lines wavelengths used by the recipe* vmmosscience. *The marked lines are those used on data from low resolution grisms.*

7.12.2 Output files

Not all output frames listed here are always produced. Some of them are created only on request, and some other are never created in case more than one scientific exposure is specified in input.

Here is the list of all the possible output frames, in alphabetical order, together with a list of related configuration parameters:[23]

MOS_SCI_DISP_COEFF_SKY: (MEF) This adjustment of the input MOS_DISP_COEFF table is only created in case the alignment of the wavelength solution to the sky lines is requested (see the configuration parameter *--skyalign*, Section 7.12.3, page 133). For a description of this product see the MOS_DISP_COEFF entry on page 111. In the MOS_SCI_DISP_COEFF_SKY table the *error* column content is computed by (quadratically) summing the errors of the input wavelength solution with the errors of the sky alignment fit. Similarly, in the *nlines* column the number of sky lines used for the alignment replaces the number of reference arc lamp lines on which the input calibration was based.

Configuration parameters directly affecting this product are *--skyalign*, *--startwavelength* and *--endwavelength*.

MOS_SCI_ERROR_FLUX_REDUCED: (MEF) error on photometrically calibrated scientific spectra. This image matches the MOS_SCIENCE_FLUX_REDUCED image, and it is produced only if the spectrophotometric calibration was requested.

MOS_SCI_ERROR_REDUCED: (MEF) image with errors (one sigma level) corresponding to the extracted objects spectra. This image matches the MOS_SCIENCE_REDUCED image.

Configuration parameters directly affecting this product are *--dispersion*, *--ext_mode*, *--time_normalise*, *--startwavelength* and *--endwavelength*.

Configuration parameters having significant impact are *--slit_margin*, *--ext_radius*, *--cont_radius*.

MOS_SCI_GLOBAL_SKY_SPECTRUM: (MEF) table with supersampled sky spectrum, created only if the global sky subtraction is requested (see configuration parameter *--skyglobal*, Section 7.12.3, page 133). Each wavelength bin is half the resampling step, multiplied by the CCD readout rebin factor (see the configuration parameter *--dispersion*, Section 7.12.3, page 133).

The spectra contained in the input scientific exposure (see the MOS_SCIENCE entry on page 124) are assumed to contain altogether at least 50% of their pixels on the sky. Moreover, all the *scientific* slits are assumed to have the same width.[24] The wavelength map derived from the input MOS_DISP_COEFF table (possibly adjusted by the sky lines alignment task) is used to map all the spectral signal in the CCD into a grid of wavelength bins. The sky spectrum is computed as the median level of all the pixel values of all the CCD spectra in each wavelength bin. The median of the contributing wavelengths (which are not uniformly distributed within the bin) is also assigned to each bin. Empty bins are computed by linear interpolation between the nearest valid bins, and in this case a bin is assigned its central wavelength.

[23] See Section 7.12.3, page 133, for a complete description of the recipe configuration parameters.

[24] If this were not the case, the global sky model quality would be poorer, and only the slits with a median slit width would be properly corrected. This may be fixed by applying a local sky subtraction following the global one, but this would eliminate the advantages of using a global sky model.

File name	Target name	Catalog
bd25d4655.tfits	BD+25d4655	Oke (1990)
bd28d4211.tfits	BD+28d4211	Oke (1990)
bd33d2642.tfits	BD+33d2642	Oke (1990)
cd32d9927.tfits	CD-32-9927	Hamuy et al. (1992, 1994)
eg21.tfits	EG-21	Hamuy et al. (1992, 1994)
eg274.tfits	EG-274	Hamuy et al. (1992, 1994)
feige110.tfits	Feige-110	Hamuy et al. (1992, 1994)
feige56.tfits	Feige-56	Hamuy et al. (1992, 1994)
feige66.tfits	Feige-66	Oke (1990)
feige67.tfits	Feige-67	Oke (1990)
g158_100.tfits	G-158-100	Oke (1990)
g93_48.tfits	G-93-48	Oke (unpublished) data
gd108.tfits	GD-108	Oke (1990)
gd50.tfits	GD-50	Oke (1990)
hilt600.tfits	Hiltner-600	Hamuy et al. (1992, 1994)
hz2.tfits	Hz-2	Oke (unpublished) data
hz44.tfits	Hz-44	Oke (1990)
lds749b.tfits	LDS-749b	Oke (1990)
ltt1020.tfits	LTT-1020	Hamuy et al. (1992, 1994)
ltt1788.tfits	LTT-1788	Hamuy et al. (1992, 1994)
ltt2415.tfits	LTT-2415	Hamuy et al. (1992, 1994)
ltt377.tfits	LTT-377	Hamuy et al. (1992, 1994)
ltt3864.tfits	LTT-3864	Hamuy et al. (1992, 1994)
ltt4816.tfits	LTT-4816	Hamuy et al. (1992, 1994)
ltt6248.tfits	LTT-6248	Hamuy et al. (1992, 1994)
ltt7379.tfits	LTT-7379	Hamuy et al. (1992, 1994)
ltt7987.tfits	LTT-7987	Hamuy et al. (1992, 1994)
ltt9239.tfits	LTT-9239	Hamuy et al. (1992, 1994)
ltt9491.tfits	LTT-9491	Hamuy et al. (1992, 1994)
ngc7293.tfits	NGC-7293	Oke (1990)

Table 7.13: *Spectro-photometric standard stars in the VIMOS Calibration Plan. Full references are: Oke, 1990, AJ 99, 1621; Hamuy et al., 1992, PASP 104, 533; Hamuy et al., 1994, PASP 106, 566. See also* http://www.eso.org/sci/observing/tools/standards/spectra/

The global sky table includes the following columns:

wavelength: Bin wavelength.
sky: Median signal level for each bin.
npoints: Number of points contributing to each bin.

Configuration parameters directly affecting this product are *--skyglobal*, *--time_normalise*, *--startwavelength* and *--endwavelength*.

MOS_SCI_SKY_REDUCED: (MEF) image with sky corresponding to the extracted objects spectra. The sky is extracted in the same way as the objects, e.g., if optimal weights were applied to the object extraction, the same weights are applied to the sky extraction. This image matches the MOS_SCIENCE_REDUCED image.

Configuration parameters directly affecting this product are *--dispersion*, *--ext_mode*, *--time_normalise*, *--startwavelength* and *--endwavelength*.

Configuration parameters having significant impact are *--slit_margin*, *--ext_radius*, *--cont_radius*, *--skyalign*, *--flatfield*, *--skylocal*, *--skyglobal* and *--skymedian*.

MOS_SCI_SKYLINES_OFFSETS_SLIT: (MEF) table containing the observed sky lines offsets that were used for adjusting the input wavelength solution. This table is only produced if the sky lines alignment is requested (see configuration parameter *--skyalign*, Section 7.12.3, page 133). It has one row for each of the sky lines used for the alignment, and one column for each slit where sky lines could be detected.[25]

The included columns are the following:

wave: Sky line wavelength.
offset<slit_id>: Observed offsets for the slit spectrum with identification *slit_id*.

This table may be very useful for judging what would be the most appropriate modeling of the observed offsets, and to what extent the input wavelength calibration really needs to be adjusted.

Configuration parameters directly affecting this product are *--startwavelength* and *--endwavelength*.

MOS_SCI_UNMAPPED_SKY: (MEF) this image has the same size of the CCD, and is created if either the global or the local sky subtraction is requested.

If *--skyglobal* is set (see the configuration parameter *--skyglobal*, Section 7.12.3, page 133), this image contains the global sky model mapped on the CCD frame, derived from the supersampled sky spectrum contained in the MOS_SCI_GLOBAL_SKY_SPECTRUM table. Each one of its pixels is assigned a value obtained by linear interpolation of the two wavelengths of the supersampled spectrum that are closest to its wavelength.

If *--skylocal* is set (see the configuration parameter *--skylocal*, Section 7.12.3, page 133), this image contains the sky model obtained by interpolating the sky signal trend along the spatial direction, directly on the CCD frame.

The global sky subtraction consists of subtracting this image from the original bias subtracted and flat field corrected scientific exposure.

[25] In general the sky lines detection fails for reference slits, that are typically filled up by very bright objects.

Configuration parameters directly affecting this product are *--skyglobal*, *--skyalign*, *--time_normalise*, *--startwavelength* and *--endwavelength*.

MOS_SCI_WAVELENGTH_MAP_SKY: (MEF) This upgraded version of the wavelength map is only produced in case the adjustment of the wavelength solution to the sky lines is requested (see the configuration parameter *--skyalign*, Section 7.12.3, page 133). For a description of this product see the MOS_WAVELENGTH_MAP entry on page 115.

Note that the coordinate system (WCS) of the WAVELENGTH_MAP frames will generally differ, because they are derived from different input data: the coordinate system of MOS_WAVELENGTH_MAP is inherited from the arc lamp frame header, while MOS_SCI_WAVELENGTH_MAP_SKY inherits from the scientific frame header.

Configuration parameters directly affecting this product are *--skyalign*, *--startwavelength* and *--endwavelength*.

MOS_SCIENCE_EXTRACTED: (MEF, ERR) image with rectified, wavelength calibrated and sky subtracted slit spectra. Its x size depends on the spectral extraction range $(\lambda_{min}, \lambda_{max})$ and on the specified resampling step in wavelength units per pixel, D, defined by the configuration parameter *--dispersion* (see Section 7.12.3, page 133):

$$N_x = floor\Big(\frac{\lambda_{max} - \lambda_{min}}{D}\Big)$$

The y size is determined in the same way as for the MOS_ARC_SPECTRUM_EXTRACTED frame (see page 108).

Note that resampling the original spectrum at a constant wavelength step introduces distortions of the signal, that depend on the original signal pixelisation on the CCD, and it introduces noise correlation. See the final note to the MOS_ARC_SPECTRUM_EXTRACTED entry on page 108.

Configuration parameters directly affecting this product are *--dispersion*, *--cosmics*, *--flatfield*, *--time_normalise*, *--skyalign*, *--startwavelength* and *--endwavelength*.

Configuration parameters having significant impact are *--skymedian*, *--skylocal*, and *--skyglobal*.

MOS_SCIENCE_FLUX_EXTRACTED: (MEF) photometrically calibrated scientific slit spectra. This image matches the MOS_SCIENCE_EXTRACTED image, and it is produced only if the spectrophotometric calibration was requested. For those wavelengths in which the response wasn't available, this product contains the value -1.

MOS_SCIENCE_FLUX_REDUCED: (MEF) photometrically calibrated scientific spectra. This image matches the MOS_SCIENCE_REDUCED image, and it is produced only if the spectrophotometric calibration was requested. For those wavelengths in which the response wasn't available, this product contains the value -1.

MOS_SCIENCE_REDUCED: (MEF) image with extracted objects spectra. This image has the same x size of the image with the extracted slit spectra, MOS_SCIENCE_EXTRACTED, and as many rows as the detected and extracted object spectra. Extracted spectra are written to the image rows listed in the OBJECT_SCI_TABLE table (columns row_XXX), which starts numbering from the top to the bottom of the detector.

Configuration parameters directly affecting this product are *--dispersion*, *--ext_mode*, *--time_normalise*, *--startwavelength* and *--endwavelength*.

Configuration parameters having significant impact are *--slit_margin*, *--ext_radius*, *--cont_radius*, *--skyalign*, *--flatfield*, *--skyglobal*, *--skylocal*, *--skymedian*, and *--cosmics*.

MOS_SCIENCE_SKY: (MEF) image with rectified and wavelength calibrated slit sky spectra. This image matches in size the MOS_SCIENCE_EXTRACTED image, and is produced only if any kind of sky subtraction is requested. This image contains the modeled sky which was subtracted from the scientific data, either before or after the scientific spectra rectification (or even both, if the configuration parameters *--skyglobal* and *--skymedian* were both set: the contribution of the global sky model is included in this image even if the global sky subtraction is really applied to the data before their rectification). The sky model component subtracted before the rectification of the scientific spectra can be viewed separately in the MOS_SCI_GLOBAL_SKY_SPECTRUM and the MOS_SCI_UNMAPPED_SKY products.

The MOS_SCIENCE_SKY also includes the identified cosmic ray signal in case the cosmic rays removal was requested (see configuration parameter *--cosmics*, Section 7.12.3, page 133).

Configuration parameters directly affecting this product are *--skymedian*, *--skyglobal*, *--skylocal*, *--cosmics*, *--time_normalise*, *--startwavelength* and *--endwavelength*.

Configuration parameters having significant impact are *--skyalign* and *--flatfield*.

MOS_SCIENCE_SKY_EXTRACTED: (MEF) image with rectified and wavelength calibrated slit spectra (without sky subtraction). This image matches in size the MOS_SCIENCE_EXTRACTED image, and is produced only if any kind of sky subtraction is requested.

Configuration parameters directly affecting this product are *--dispersion*, *--flatfield*, *--time_normalise*, *--skyalign*, *--startwavelength* and *--endwavelength*.

MOS_SPECPHOT_TABLE: tables with efficiency and response curves, produced only when input includes a standard star observation, an EXTINCT_TABLE, and the appropriate STD_FLUX_TABLE matching the observed star.

The MOS_SPECPHOT_TABLE has two table extensions. The first table extension contains wavelength bins that correspond to the input STD_FLUX_TABLE while the second extension contains wavelength bins that correspond to the observed spectrum.

The first table extension include the following columns:

WAVE:	Wavelength (Å)
STD_FLUX:	Standard star flux (10^{-16} erg cm^{-2} s^{-1} Å^{-1})
OBS_FLUX:	Observed flux (e^- s^{-1} Å^{-1})
RAW_EFFICIENCY:	Ratio between input and detected photons
EFFICIENCY:	Fit of **RAW_EFFICIENCY** by using polynomial or spline
RAW_RESPONSE:	Ratio between **std_flux** and **obs_flux**
RESPONSE:	Fit of **RAW_RESPONSE** by using polynomial or spline
USED_FIT:	Flag to signal whether this bin was used in the response fit

If the response has been corrected by the flat SED then there are additional columns added:

OBS_FLUX_FFSED:	Observed flux (OBS_FLUX) divided by the flat SED
RAW_RESPONSE_FFSED:	Ratio between **std_flux** and **obs_flux_ffsed**
RESPONSE_FFSED:	Fit of **RAW_RESPONSE_FFSED** by using polynomial or spline

The second table extension include the following columns:

WAVE: Wavelength (Å)
EFFICIENCY: Fit of **RAW_EFFICIENCY** (from the in first table extension) by using polynomial or spline
RESPONSE: Fit of **RAW_RESPONSE** (from the in first table extension) by using polynomial or spline

The way this table is produced is described in Section 9.15, page 173.

Configuration parameters directly affecting this product are *--resp_fit_degree*, *--resp_fit_nknots*, *--resp_ignore_mode*, *--resp_ignore_points*, *--resp_use_flat_sed*, *--startwavelength*, *--endwavelength*.

MOS_UNMAPPED_SCIENCE: image with the sky subtracted scientific spectra on the CCD frame, created only if the global or the local sky subtraction is requested (see the configuration parameters *--skyglobal* and *--skylocal*, Section 7.12.3, page 133). This image is derived subtracting the MOS_SCI_UNMAPPED_SKY from the bias subtracted and flat fielded scientific frame.

Configuration parameters directly affecting this product are *--skyglobal*, *--skylocal*, *--skyalign*, *--time_normalise*, *--startwavelength* and *--endwavelength*.

OBJECT_SCI_TABLE: (MEF) This table is an expansion of the input MOS_SLIT_LOCATION table (see page 113), where the positions and the extraction spatial intervals of the detected objects are also included. This table is produced only if any kind of sky subtraction is requested, otherwise no object detection or extraction is attempted. The object table columns are the following:

slit_id: Slit identification number.
xtop: x CCD position of central wavelength from left end of slit.
ytop: y CCD position of central wavelength from left end of slit.
xbottom: x CCD position of central wavelength from right end of slit.
ybottom: y CCD position of central wavelength from right end of slit.
position: First row of the rectified images (such as MOS_SCIENCE_EXTRACTED) containing the rectified slit spectrum. Image rows are counted from bottom, starting from 0.
length: Number of rows in rectified images including the slit spectrum.
object_1, object_2, ...: Detected objects positions in the rectified images.
start_1, start_2, ...: Start position of the extraction interval for each object.
end_1, end_2, ...: End position of the extraction interval for each object.
row_1, row_2, ...: Row number of the MOS_SCIENCE_REDUCED image containing the extracted object spectrum. Image rows are counted from bottom, starting from 0. The sources themselves are numbered from the top to the bottom of the detector. If more than one object per slit is detected, the object referenced in row_1 will be in the row above object in row_2, i.e., row_1 = row_2 + 1.

Configuration parameters directly affecting this product are *--slit_margin*, *--ext_radius*, *--cont_radius*.

Configuration parameters that may have significant impact are *--startwavelength* and *--endwavelength*.

MOS_SCIENCE_REDUCED_IDP_i: IDP-compliant extracted spectra, see [2]. i is an index between 1 and the number of extracted spectra. A product is generated for each spectra. The products are generated only if *--generate_idp* is set to TRUE. If the flux calibration is not available or the position angle on the sky is neither 0 nor 90 degrees a warning is produced and the recipe completes successfully skipping IDP generation.

The configuration parameter *--generate_idp* affects directly this product.

7.12.3 Configuration parameters

The configuration parameters setting determines the way the *vmmosscience* recipe will process the input data, and to some extent the product files that will be created. The parameters are conveniently divided into seven main sections: wavelength calibration, spatial curvature calibration, flat field correction, sky subtraction, alignment and stacking of input frames, objects detection and extraction, and flux calibration.

Wavelength calibration

--skyalign: Polynomial order for sky lines alignment. *Default*: -1

The input wavelength calibration can be adjusted to the observed positions of a set of sky lines, whose wavelengths are listed in an input catalog. The observed sky lines offsets from their expected positions (see entry MOS_SCI_SKYLINES_OFFSETS_SLIT, page 129) are fitted by polynomials that are then added to the input wavelength calibration polynomials (see MOS_DISP_COEFF entry on page 111). A *--skyalign* = 0 would just determine a median offset from all the observed sky lines, while *--skyalign* = 1 would try to fit a slope (often useful with VIMOS data, where the mean spectral dispersion depends on the temperature and the sky lines offsets display a significant dependency on the wavelength with respect to the day calibration). Polynomials with order greater than 2 generate a friendly error message. Setting *--skyalign* < 0 disables any sky line alignment, accepting the input wavelength calibration as-is.

Note that the *--skyalign* parameter should be more correctly intended as the *maximum* applicable polynomial order: the polynomial is really adapted to the number of identified sky lines used in the fit. As a rule, a polynomial with the specified order is only used if the number of identified sky lines is at least greater than the number of free parameters: if this were not the case, the applied polynomial order would be

$$n = N - 1$$

where N is the number of identified sky lines. Consistently, for $n = 0$ a median offset would be computed.

Spatial curvature calibration

The input curvature model is not aligned to the observed scientific slit spectra. Offsets up to 1 pixel may be observed in the tracing of scientific spectra.

Flat field correction

--flatfield: Apply flat field correction. *Default*: TRUE

Setting this parameter makes mandatory to specify a normalised flat field frame (see page 113, entry MOS_MASTER_SCREEN_FLAT). The flat field correction consists in dividing the bias subtracted input scientific frame by the normalised flat field frame.

Sky subtraction

--skylocal: Subtract sky spectrum from CCD scientific data. *Default*: TRUE

The local sky subtraction consists on modeling the sky trend along the dispersion axis for each spectrum on the CCD.[26] The advantage of this method is that the signal is not resampled before the sky is subtracted, reducing in this way the problems related to small-scale interpolation.

A MOS_SCIENCE_SKY (page 131) is produced in this case. Note that global and median sky subtractions cannot be used if the local sky subtraction is applied.

Note however that this method may not apply well to curved or tilted slits: in such case, the *--skymedian* option may be preferred.

--skymedian: Subtract sky spectrum from rectified scientific data. *Default*: FALSE

The median sky subtraction consists on subtracting a median value of the sky for each wavelength pixel of each rectified slit spectrum.[27]

In general the subtraction of a rectified sky spectrum from rectified data does not give the best results, and in almost all cases the local sky subtraction (see parameter *--skylocal*) should be preferred.

A MOS_SCIENCE_SKY (page 131) is produced in this case. Note that global and median sky subtractions are not mutually exclusive.

--skyglobal: Subtract global sky spectrum from CCD. *Default*: FALSE

In general the subtraction of a global sky spectrum does not give the best results, because the spectral resolution may vary significantly with the position on the CCD. However this operation may turn out to be useful in case either a local or a median sky subtraction would actually destroy spectra from extended objects that fill all, or almost all, the extension of a slit. See entries MOS_SCI_UNMAPPED_SKY and MOS_SCI_GLOBAL_SKY_SPECTRUM on page 129 for more details.

--cosmics: Eliminate cosmic rays hits. *Default*: FALSE

If this parameter is set, then either the local or the global sky subtraction must be requested (see parameters *--skylocal* and *--skyglobal*). Cosmic rays cleaning is almost always superfluous, and should be viewed as mere cosmetics applied to the extracted slit spectra (see entry MOS_SCIENCE_EXTRACTED page 130). Cosmic ray hits are removed anyway by the optimal extraction procedure of the detected objects. The algorithm used by this parameter is explained in 9.9.

Alignment and stacking of input frames

[26] This is an iterative process: initially the sky trend is estimated with a robust linear fitting, then outliers (e.g., objects) are rejected, and according to the slit length the sky is trended using a 2nd degree polynomial.

[27] This is an iterative process: initially the sky is estimated as the median value of all the pixels at the same wavelength, then this first estimation of the sky is subtracted, and the objects are detected; finally the median level is evaluated only on pixels outside the object detection spatial interval.

The following parameters are active only in case the recipe *vmmosscience* is run on more than one input scientific frame. This may happen with dithered observations, or with scientific frames acquired on different nights.

--dither: Align dithered frames before stacking. *Default*: TRUE

If this parameter is set, the input scientific frames will be spatially aligned to the first input frame before being stacked. The stacking happens also before any rectification.

--alignment: Type of alignment of dithered frames. (**integer** = nearest neighbour pixel alignment, **float** = alignment to fractions of pixel). *Default*: **integer**

The spatial alignment of different scientific frames can be done to the nearest integer pixel, or to a fraction of pixel (implying signal interpolation). Alignment to the nearest pixel has the advantage of preserving the noise characteristics of the signal.

--compute: Compute offsets of dithered images. *Default*: FALSE

If *--compute* is set, the relative offsets for the frames alignment are determined using the common detected objects in each frame, while if *--compute*=false the offsets are derived from the pointing direction of the telescope read from the frames FITS headers (keywords RA and DEC). If the input frames belong to different nights, it may advisable to set *--compute*=true.

--stack_method: Frames combination method (**average** = simple average of all input frames, **median** = median stacking of all input frames, **minmax** = stacking frames with minmax rejection, **ksigma** = average frames with k-sigma clipping). *Default*: **average**

If *--stack_method* is set to *minmax*, the following parameters become relevant:

--minrejection: Number of lowest values to be rejected. *Default*: 1

For each pixel position, the number of lowest pixel values specified here are rejected before computing the mean of the remaining pixel values.

--maxrejection: Number of highest values to be rejected. *Default*: 1

For each pixel position, the number of highest pixel values specified here are rejected before computing the mean of the remaining pixel values. The sum of the number of highest and lowest rejected pixels should be less than the number of input frames.

If *--stack_method* is set to *ksigma*, the following parameters become relevant:

--klow: Number of sigmas for lower values rejection. *Default*: 3.0

For each pixel position, a robust determination of the standard deviation from the *median* pixel value is made. All pixel values with a negative residual greater than the specified number of sigmas are rejected, the other values are averaged.

--khigh: Number of sigmas for upper values rejection. *Default*: 3.0

For each pixel position, a robust determination of the standard deviation from the *median* pixel value is made. All pixel values with a positive residual greater than the specified number of sigmas are rejected, the other values are averaged.

--kiter: Maximum number of iterations. *Default*: 999

Maximum number of iterations of the rejection process. The iteration stops as soon as no outliers are detected, or when reaching the maximum number of iterations. At each iteration the median value and the standard deviation are recomputed, and a new k-sigma rejection is applied.

--fringing: Apply fringing correction. *Default*: TRUE

This parameter is only effective in case of dithered observations with relative offsets greater than the amount specified by the *--offset* parameter. If set, the sky fringing will be estimated and eliminated using the algorith described in 9.14.

--offset: Minimum required offset between input frames for applying the sky fringing correction. *Default*: 3.0 pixel

If the minimum offset between input frames is less than the value specified here, the sky fringing correction is not applied (even if *--fringing*=true).

Objects detection and extraction

--detection: Object detection threshold *Default*: 2.0 ADU

Objects are detected within the slit limts applying algorithm described in 9.21.3. This parameter is used to decide whether the central pixel of an object profile is above the threshold detection level.

--slit_margin: Spectrum edge pixels to exclude from object search. *Default*: 3 pixel

The object detection task will reject objects that are detected too close to the edges of a slit spectrum. There might be different reasons for this, such as objects would be truncated, too close to a confusion region, etc.

--ext_radius: Maximum extraction radius for detected objects. *Default*: 6 pixel

The default value is generally good when dealing with point-like objects, but it should be adapted to the size of more extended objects when necessary. Large values of the extraction radius would not harm the extraction quality if an optimal extraction algorithm is applied, but may have devastating effects on the results of a simple aperture extraction. The applied extraction interval is reduced in case nearby objects are detected: an intermediate position between two objects, computed according to the objects luminosity ratio, is never passed.

--cont_radius: Contamination radius. *Default*: 0 pixel

This parameter may help to prevent the extraction of contaminated objects. The contamination radius is the minimum distance at which two point-like objects of equal luminosity are assumed not to contaminate each other. For two objects having different luminosities the reciprocal contamination distances depend on their luminosity ratio. Indicating with L_o the peak value of one object integrated spatial profile and with L the peak value of a nearby object, the quantity

$$S = C \cdot \left(\frac{L}{L_o}\right)$$

is computed, where C is the specified contamination radius. If the distance between the two objects is less than S, the examined object is flagged as contaminated and is not extracted. This empirical formula has

the effect of assigning a larger contamination radius to relatively brighter objects with respect to dimmer ones.

Note that the luminosities appearing above are averaged over the spectral axis.

--ext_mode: Object extraction method. *Default*: 1

Only two methods are currently available for spectral extraction: *--ext_mode* = 0 corresponds to simple aperture extraction, while *--ext_mode* = 1 applies Horne's optimal extraction [14].

Flux calibration

--resp_fit_nknots: Number of knots of the response spline fitting. If -1, then no spline fitting is performed. If -2, then the value is read from the CONFIG_TABLE (see Section 7.11.1).*Default*: -2

See Section 9.15, page 173, for the meaning of this parameter.

--resp_fit_degree: Degree of polynomial for the response polynomial fitting. If -1, then no polynomial fitting is performed. If -2, then the value is read from the CONFIG_TABLE (see Section 7.11.1)*Default*: -2

Take into account that if both *--resp_fit_nknots* and *--resp_fit_degree* are greater than 0 the settings are incompatible and the pipeline will stop. See Section 9.15, page 173, for the meaning of this parameter.

--resp_ignore_mode: Types of lines/regions to ignore in response. Valid ones are 'stellar_absorption', 'telluric' and 'command_line' (from parameter resp_ignore_points) *Default*: stellar_absorption,telluric,command_line

This parameter contains a comma separated list of types of lines or regions to ignore during response computation. If it includes *stellar_absorption* then the lines marked as stellar absorption in the standard star calibration table (STD_FLUX_TABLE) will be ignored. If it includes *telluric* then the lines and regions specified in the telluric contamination table (TELLURIC_CONTAMINATION) will be ignored. If it includes *command_line*, the lines and regions specified in parameter *resp_ignore_points* will be ignored.

--resp_ignore_points: Extra lines/regions to ignore in response *Default*:

This parameter contains a comma separated list of lines to ignore during response computation (if parameter *response_ignore_model* contains *command_line*). A range can also be specified like 4500.0-4600.0.

--resp_use_flat_sed: Flag to determine whether to apply flat sed correction *Default*: grism_table

Possible values are *true, false, grism_table*. If *true*, then the observed spectra will be divided by the flat sed before applying the photometric calibration. This is needed for the proper calibration of holographic grisms which show a position dependant response. If the observed targe tis a standard star, then the response will contain this correction and the science must also be corrected by the same effect.

If the value is *grism_table* then the option is read from the CONFIG_TABLE table, which contains the column **RESP_USE_FLAT_SED**. This is because for some grism (specially holographic ones) this option is strongly recommended, while for others it is not needed.

--time_normalise: Apply exposure time normalisation to relevant products. *Default*: TRUE

The following products are affected by this parameter:

- MOS_SCI_GLOBAL_SKY_SPECTRUM
- MOS_SCIENCE_SKY_EXTRACTED
- MOS_SCIENCE_EXTRACTED
- MOS_SCIENCE_SKY
- MOS_SCIENCE_REDUCED
- MOS_SCI_SKY_REDUCED
- MOS_SCI_ERROR_REDUCED
- MOS_UNMAPPED_SCIENCE, and,
- MOS_UNMAPPED_SKY.

--anyframe: Attempt to reduce any dataset classified as a standard star exposure. *Default*: FALSE

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 star is exposed once for each VIMOS quadrant, and reducing the frames from the unused quadrants is not really a requirement. Setting this parameter to *false* would prevent the processing of such images.

IDP Generation

--generate_idp: Flag to determine whether to generate IDP-compliant products. *Default*: FALSE

If IDP generation fails, the recipe is still considered successful.

7.12.4 Quality control parameters

QC MOS EFFICIENCYi LAMBDA: Wavelength used for step *i* to report the efficiency. This is only computed if the input includes a standard star. *Units*: Angstrom

QC MOS EFFICIENCYi: The efficiency value at wavelength **QC MOS EFFICIENCYi LAMBDA**. This is only computed if the input includes a standard star. *Units*: e^-/photom

QC RESP FLAT SED_CORR: Whether the flat used to reduce the standard star has been sed normalised. This is only present if the input includes a standard star. *Units*: boolean.

QC RESP FLAT SED_NORM: The sed normalisation factor from the master flat used in the slit where the standard star is placed. This is only present if the input includes a standard star and QC RESP FLAT SED_CORR is TRUE. *Units*: ADUs

QC FLAT SED CORR_SLITWID: Whether the normalisation factor used in the flat contains the slit width. This is only present if the input includes a standard star and QC RESP FLAT SED_CORR is TRUE. *Units*: boolean.

File name	Target name	Catalog
bd25d4655.tfits	BD+25d4655	Oke (1990)
bd28d4211.tfits	BD+28d4211	Oke (1990)
bd33d2642.tfits	BD+33d2642	Oke (1990)
cd32d9927.tfits	CD-32-9927	Hamuy et al. (1992, 1994)
eg21.tfits	EG-21	Hamuy et al. (1992, 1994)
eg274.tfits	EG-274	Hamuy et al. (1992, 1994)
feige110.tfits	Feige-110	Hamuy et al. (1992, 1994)
feige56.tfits	Feige-56	Hamuy et al. (1992, 1994)
feige66.tfits	Feige-66	Oke (1990)
feige67.tfits	Feige-67	Oke (1990)
g158_100.tfits	G-158-100	Oke (1990)
g93_48.tfits	G-93-48	Oke (unpublished) data
gd108.tfits	GD-108	Oke (1990)
gd50.tfits	GD-50	Oke (1990)
hilt600.tfits	Hiltner-600	Hamuy et al. (1992, 1994)
hz2.tfits	Hz-2	Oke (unpublished) data
hz44.tfits	Hz-44	Oke (1990)
lds749b.tfits	LDS-749b	Oke (1990)
ltt1020.tfits	LTT-1020	Hamuy et al. (1992, 1994)
ltt1788.tfits	LTT-1788	Hamuy et al. (1992, 1994)
ltt2415.tfits	LTT-2415	Hamuy et al. (1992, 1994)
ltt377.tfits	LTT-377	Hamuy et al. (1992, 1994)
ltt3864.tfits	LTT-3864	Hamuy et al. (1992, 1994)
ltt4816.tfits	LTT-4816	Hamuy et al. (1992, 1994)
ltt6248.tfits	LTT-6248	Hamuy et al. (1992, 1994)
ltt7379.tfits	LTT-7379	Hamuy et al. (1992, 1994)
ltt7987.tfits	LTT-7987	Hamuy et al. (1992, 1994)
ltt9239.tfits	LTT-9239	Hamuy et al. (1992, 1994)
ltt9491.tfits	LTT-9491	Hamuy et al. (1992, 1994)
ngc7293.tfits	NGC-7293	Oke (1990)

Table 7.14: *Spectro-photometric standard stars in the VIMOS Calibration Plan. Full references are: Oke, 1990, AJ 99, 1621; Hamuy et al., 1992, PASP 104, 533; Hamuy et al., 1994, PASP 106, 566. See also* http://www.eso.org/observing/standards/spectra/

7.13 vmifucalib

The VIMOS pipeline recipe *vmifucalib* is used to determine the spatial 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 7.15.

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

Table 7.15: *Input files for the vmifucalib recipe.*

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 spatial extraction mask can be determined.

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

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 reference 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 ideally one IFU_IDENT file for each quadrant/grism combination, named `ifu_ident_`*grism*`_`*q*`.fits` (where *q* indicates the VIMOS quadrant number, and *grism* the grism name). A new set of fiber identification files was added in 2006, in order to support IFU data obtained around 2006 and after. The earlier IFU_IDENT files had the suffix "_2006" added to their names. A further set was added after the Summer 2010 intervention, and similarly the previous IFU_IDENT files had the suffix "_2010" added to their names.

If a 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 optical-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 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 increased by 4). Care should be taken in selecting the appropriate bad pixel tables for the spectral instrument modes (in the case of imaging data q is the quadrant number).

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 shown in Table 7.16.

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.fits	IFU_TRACE	FITS	Extraction mask
ifu_ids.fits	IFU_IDS	FITS	Wavelength calibration
ifu_transmission.fits	IFU_TRANSMISSION	FITS	Transmission correction

Table 7.16: *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.

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 7.17: *IFU position table entries.*

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 7.17). 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`.

The content of the calibration tables generated by the recipe *vmifucalib* is described in Tables 7.18, 7.19, and 7.20.

Column name	Explanation
Ci	i^{st} coefficient of the spectrum tracing polynomial.
RMS	Standard deviation of polynomial fit.

Table 7.18: *IFU extraction mask.*

In the extraction mask `ifu_trace.fits`, 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 7.19: *Inverse dispersion solution.*

In the inverse dispersion solution `ifu_ids.fits` 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 7.20: *Relative transmission factors for all fibers.*

The `ifu_transmission.fits` 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 7.21.

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 9.10. 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. The bad pixel correction algorithm is described in Section 9.8, page 167.

ComputeQC: If this parameter is set, *Quality Control* (QC) parameters will be computed and written to the header of the output tables. Currently the QC parameters computed by *vmifucalib* are:

`QC IFU LOSTi`: Number of fibers that could not be traced on pseudo-slit i.

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
LineIdent	FirstGuess Blind	Line identification based on first-guess models Line identification based on pattern recognition
MaxTraceRejection	*int*	Maximum percentage of rejected positions in tracing
ComputeQC	true false	Compute QC parameters Do not compute QC parameters

Table 7.21: *vmifucalib parameters.*

`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"*.

LineIdent: Arc lines identification method. The identification procedure is described in some detail in Section 9.22.4. Possible settings are:

Blind: Arc lamp lines are identified without making use of first-guess IDS models. This method, based on pattern recognition, just requires the rough estimate of the expected spectral dispersion read from the input grism table.

FirstGuess: Arc lamp lines are identified on the basis of existing models of the spectral distortions, used as first-guesses.

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 9.12 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 9.24, page 218.

7.14 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, and they may optionally be flux calibrated.

The files to be included in the input SOF are listed in Table 7.22.

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	√
EXTINCT_TABLE	Calibration	Atmospheric extinction table	
IFU_SPECPHOT_TABLE	Calibration	Spectro-photometric table	
CCD_TABLE	Calibration	Bad pixel table	

Table 7.22: *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 7.13, page 140).

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 increased by 4). Care should be taken in selecting the appropriate bad pixel tables for the spectral instrument modes (in the case of imaging data *q* is the quadrant number).

If a flux calibration is requested, a spectro-photometric table produced by the recipe *vmifustandard* must be specified together with an atmospheric extinction table (see Tables 7.10 on page 124, and 7.27 on page 149).

Note that a flux calibration can be applied to the reduced data at a later stage, using the recipe *vmspphot* (see Section 7.18, page 149).

All the products of the *vmifuscience* recipe are shown in Table 7.23.

File name	DO category	Type	Explanation
ifu_science_reduced.fits	IFU_SCIENCE_REDUCED	FITS	Reduced scientific spectra
ifu_science_reduced.fits	IFU_SCIENCE_FLUX_REDUCED	FITS	Flux calibrated spectra
ifu_fov.fits	IFU_FOV	FITS	Reconstructed field-of-view image

Table 7.23: *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 7.17 on page 141, and its description in Section 7.13).

If a flux calibration is requested, then an `IFU_SCIENCE_FLUX_REDUCED` image is also created. This image is identical to the `IFU_SCIENCE_REDUCED` one, but the spectra it contains are flux calibrated, and expressed in units of 10^{-16} erg cm^{-2} s^{-1} Å^{-1}. Note, however, that the obtained fluxes are valid only if the input IFU_SPECPHOT_TABLE is based on a standard star observed under similar atmospheric conditions.

The *vmifuscience* parameters are listed in Table 7.24.

Parameter	Possible values	Explanation
BiasMethod	Master	Bias removal with no overscan correction
	Zmaster	Bias removal with overscan correction
CleanBadPixel	true	Clean bad pixels
	false	Do not clean bad pixels
UseSkylines	true	Use sky lines to align wavelength calibration
	false	Do not align wavelength calibration
UseSkyIndividual	true	Align spectra to sky individually
	false	Same sky alignment for all spectra
CalibrateFlux	true	Apply flux calibration to extracted spectra
	false	Do not apply flux calibration

Table 7.24: *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 9.10. 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.

CalibrateFlux: Flux calibration of the extracted spectra. If this option is turned on, an atmospheric extinction table and a spectro-photometric table (see tables 7.10 and 7.27, pages 124 and 149) should be specified in the input SOF. The flux calibration is applied as described in Section 9.16, page 175.

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. The bad pixel correction algorithm is described in Section 9.8, page 167.

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 7.13, page 140), before using them in the spectral extraction task.

UseSkyIndividual: If this parameter is set, together with *UseSkylines*, the alignment of the wavelength solution to the observed positions of the reference sky lines is made independently for each fiber spectrum.

A description of the algorithms used in this recipe is given in Section 9.25, page 218.

7.15 vmifustandard

The VIMOS pipeline recipe *vmifustandard* is used to extract the IFU spectra of a spectro-photometric standard star 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 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. Finally, the instrument efficiency and the response curves are derived by comparison with the corresponding catalog spectrum.

The files to be included in the input SOF are listed in table 7.25.

The extraction mask, the wavelength calibration, and the relative transmission table, are those generated by the recipe *vmifucalib* (see Section 7.13, page 140).

An atmospheric extinction table and the standard star flux table must be specified (see Tables 7.10 and 7.11, page 124). A set of standard star flux tables, corresponding to the 30 spectro-photometric standard stars that are included in the VIMOS calibration plan ([9]), is available in the calibration directories. The names of these tables, and the name of the standard stars as reported in the FITS header keyword ESO OBS TARG NAME, are listed in Table 7.13, page 128. The table indicated in the SOF should match the content of the header entry ESO OBS TARG NAME of the input standard star exposure.

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	√
EXTINCT_TABLE	Calibration	Atmospheric extinction table	√
STD_FLUX_TABLE	Calibration	Spectro-photometric table	√
CCD_TABLE	Calibration	Bad pixel table	

Table 7.25: *Input files for the vmifustandard recipe.*

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 increased by 4). Care should be taken in selecting the appropriate bad pixel tables for imaging and spectral instrument modes (in the case of imaging data *q* is the quadrant number).

All the products of the *vmifustandard* recipe are shown in Table 7.26.

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
ifu_specphot_table.fits	IFU_SPECPHOT_TABLE	FITS	Response curves

Table 7.26: *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 7.17 on page 141, and its description in Section 7.13).

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 output spectro-photometric table has the format specified in table 7.27.

The *vmifustandard* parameters are the same as for recipe *vmifuscience*, and they are listed in Table 7.24. The only exception is the parameter *CalibrateFlux*, missing in the *vmifustandard* recipe, and the parameter *ComputeQC*, that is typically set for monitoring the instrument efficiency at specific wavelengths.

Column name	Explanation
WAVE	Wavelength in Ångstrom
STD_FLUX	Standard star flux in erg cm^{-2} s^{-1} Å^{-1}
OBS_FLUX	Observed flux in e^- s^{-1} Å^{-1}
RAW_EFFICIENCY	Ratio between input and detected photons
EFFICIENCY	Heavily smoothed version of RAW_EFFICIENCY
RAW_RESPONSE	Ratio between STD_FLUX and OBS_FLUX
RESPONSE	Heavily smoothed version of RAW_RESPONSE

Table 7.27: *IFU Spectro-photometric table.*

A description of the algorithms used in this recipe is given in Section 9.26, page 219.

7.16 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.

The mosaic is composed after a relative flux correction between the different input quadrants is applied.

7.17 vmifucombinecube

This recipe is used to rearrange into a single cube the images of extracted spectra from different VIMOS quadrants. Such images are created by the recipes *vmifucalib*, *vmifuscience*, and *vmifustandard*. The input images must belong to different quadrants, so that they cannot be more than 4. The allocated cubes are the smallest possible, depending on the number of quadrants involved and on whether the IFU shutter was on or off. The smallest cubes (20x20xN) are produced when just one quadrant is input and the shutter was on. The largest cubes (80x80xN) are produced when 3 or more quadrants are given in input (and when either quadrants 1 and 3, or 2 and 4, are input), and the shutter was off. The final cube contains basic WCS information (see section 9.23).

7.18 vmspphot

This recipe is used to apply a flux calibration to any number of 1D-extracted spectral frames generated by the recipe *vmifuscience*. The input set-of-frames will include a list of either `MOS_SCIENCE_REDUCED` or `IFU_SCIENCE_REDUCED` frames, all obtained from the same instrument quadrant and with the same instrument mode. An atmospheric extinction table `EXTINCT_TABLE`, and a spectro-photometric table consistent with the chosen instrument mode (that is either `MOS_SPECPHOT_TABLE` or `IFU_SPECPHOT_TABLE`), shall also be added.

This recipe has just one configuration parameter, *ApplyResponse*, that may be set to `false` to indicate that just the atmospheric extinction correction should be applied to the input data. In that case an input spectro-

photometric table is not required. If *ApplyResponse* is set to `true`, then the instrument response correction is also applied. The flux calibration is applied as described in Section 9.16, page 175.

A number of output calibrated frames, equal to the number of input spectral frames, will be created by this recipe. The products header keyword `ESO PRO AIRMASS` will always be set to zero, to indicate that an atmospheric extinction correction was applied. In case an instrument response curve is also applied, then the header keyword `EXPTIME` will be set to 1.0 seconds.

8 Geometrical distortions models

Optical distortions modeling is performed by the VIMOS pipeline by simple polynomial fitting of known quantities *vs* corresponding instrumental responses, e.g. celestial coordinates of astrometric stars, or pinholes on a calibration mask, or standard stars fluxes, or spectral lines wavelengths from a catalog, all are compared to the positions of detected features and patterns on the detector.

The pipeline recipes related to geometrical calibrations in the imaging instrument mode 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, and applied (typically as a first-guess for reference objects identification) on subsequent recipes runs.

Spectral distortion models are also produced by the pipeline recipes related to the MOS instrument mode, but solely to the purpose of QC and instrument health monitoring. Such models are no longer copied to the IWS, since they are not required anymore. Starting from release 2.5.0, the old MOS pipeline recipes have been decommissioned and replaced by new ones which do not require first-guess instrument modeling in order to work, being based on pattern recognition techniques (see Section 9.20, page 179). This new approach was spurred in the past years by the need to cope with the mechanical instabilities typically affecting any real-world instrument (VIMOS topping them all).[28] Optical distortions are not expected to remain constant in time. Small changes are introduced by a changing 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, imposing a constant maintenance effort on first-guess models.[29]

Only the VIMOS imaging pipeline recipes will keep using the distortion models contained in the datasets headers as “first guesses”, since such models are much simpler conceptually and therefore much easier to maintain.

With regard to the IFU instrument mode, the topic of spectral distortions doesn’t apply, since each fiber spectrum is traced and wavelength calibrated individually, making trivial the problem of calibration.

8.1 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 which may include also the geometrical distortions.

Here is an overview of the polynomials used to model each distortion:

[28] Recipes depending on good first-guess modeling work well only until an instrument is mechanically and optically stable: when the instrument distortions change signicantly, any first-guess driven algorithm fails, reducing to a mere abstraction the concept of "automatic instrument monitoring".

[29] In the case of VIMOS, with its 4 quadrants, 6 grisms, and 3 component spectral distortion models, the recomputation, by hand, of 72 spectral distortion models was required at least at any major instrument intervention. With the new recipes, this is no longer necessary.

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) (*no longer produced*)
  - Two bivariate polynomials (separately for the X and the Y CCD coordinate) of mask coordinates *vs* CCD positions
- Optical Distortion (OPT) (*no longer produced*)
  - Two bivariate polynomials (separately for the X and the Y CCD coordinate) of mask coordinates *vs* CCD positions
- Spatial Curvature (CRV) (*no longer produced*)
  - *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) (*no longer produced*)
  - *Local IDS*: Simple polynomial fits of wavelengths *vs* CCD positions
  - *Global IDS*: Bivariate polynomial fits of the coefficients of local IDS *vs* CCD positions
- Global spectral distortion (GDT)
  - New parametrisation of the spectral distortions as produced by the new *vmmoscalib* recipe.

The so-called "optical distortion model" is really a transformation from Mask to CCD coordinates valid for the spectral instrument setup, which 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.

Note that none of the above mentioned spectral models is supported any more, with the only exception of the GLOBAL_DISTORTION_TABLE (GDT). Their documentation is left in this manual just for reason of completeness, since the old VIMOS / MOS recipes are still offered to the public.

Details on the algorithms applied by the relevant pipeline recipes can be found in Section 9. In the present section just a description of the geometrical distortion models is given.

8.2 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.

8.2.1 CCD to Mask transformation and its inverse

The transformation from CCD to Mask coordinates is described by a two-layer model, consisting of a transformation including rotation, shift, and scaling, to which a bivariate polynomial fit of the residuals is added.

The base 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 base transformation for quadrant q are written to the entries of the `IMG_mask2ccd_`q`.cmf` IWS configuration file indicated in Table 8.1.

The residuals to the base transformation are modeled by a bivariate polynomial, that accounts for the higher order distortions of the instrument:

$$\begin{cases} \Delta x = \sum_{i,j} x_{ij} X^i Y^j & \text{with } 0 \le i \le m, 0 \le j \le m \\ \Delta y = \sum_{i,j} y_{ij} X^i Y^j & \text{with } 0 \le i \le m, 0 \le j \le m \end{cases}$$

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 8.1: *CCD to Mask linear transformation coefficients.*

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 8.2.

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 8.2: *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 base 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.

8.2.2 CCD to Sky distortion and its inverse

For transforming CCD pixel coordinates to celestial 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), which 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 8.1).

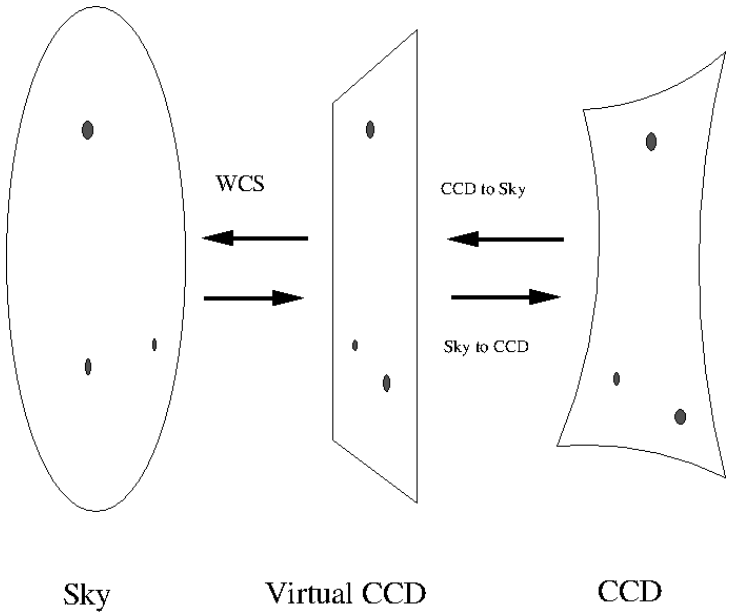

Figure 8.1: *Transformations and distortions between sky and CCD..*

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 8.3.

CCD to Sky distortion model	
`IMG_sky2ccd_q.cmf`	coefficient
`PRO CCD SKY XORD`	m
`PRO CCD SKY YORD`	m
`PRO CCD SKY X_j_i`	α_{ij}
`PRO CCD SKY Y_j_i`	β_{ij}

Table 8.3: *CCD to Sky distortion model coefficients. Please be aware of the indexes order.*

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.

8.3 Spectral distortions

We include in this section any transformation between different coordinate systems while the instrument is configured in spectral (MOS, not 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.

Currently a global description of such distortions is produced by the new recipe *vmmoscalib* and stored in the GLOBAL_DISTORTION_TABLE.

8.3.1 Global distortion table

The global distortion table contains the modeling of the coefficients of the local distortion models listed in any MOS_DISP_COEFF and MOS_CURV_COEFF tables (see page 110).

The global distortion parametrisation is attempted by the recipe *vmmoscalib* whenever at least 6 slit spectra are identified on the CCD (i.e., matched to the corresponding slits on the mask).

This table is used for enabling the on-line processing of scientific data with the recipe *vmmosscience* when appropriate (day) calibrations are not yet available. In fact, it may be input to the recipe *vmmosscience* instead of the MOS_SLIT_LOCATION, MOS_DISP_COEFF and MOS_CURV_COEFF tables.[30]

Conventionally this table consists of 6 columns and 10 rows. Each row corresponds to the modeling of one coefficient of the polynomials solutions obtained for each individual slit spectrum, using a best-fitting a bivariate polynomial:

$$c_r = \sum_{i=0}^{2} \sum_{j=0}^{2-i} a_{ij} x^j y^i$$

where r is the table row number (counted from 0) and c_r is a polynomial coefficient of a local solution. For $r = 0$ and $r > 6$ (x, y) are positions on the mask, otherwise they are positions on the CCD. The first 6 table rows are a global description of the dispersion solution up to the fifth polynomial degree; these rows are followed by a row where just the first element is assigned the value of the reference wavelength used for the given dispersion

[30] The quality of the scientific products, however, will be much less accurate in this case.

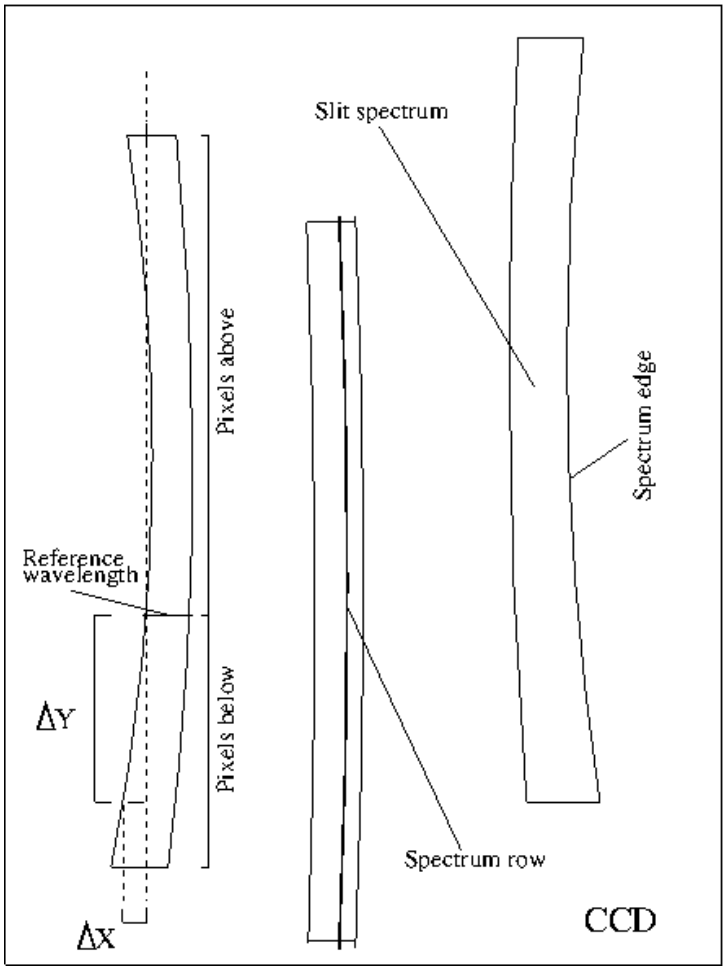

Figure 8.2: *MOS slit spectra on a CCD.*

solution. The remaining 3 rows are a global description of the spatial curvature up to the second polynomial degree. The local dispersion solutions could be obtained with:

$$y = \sum_{r=0}^{5} c_r (\lambda - \lambda_o)^r$$

where y is the y CCD pixel position and λ_o is the chosen reference wavelength. The local spatial curvature solutions could be obtained with:

$$x = \sum_{r=7}^{9} c_r y^{(r-7)}$$

where x is the x CCD pixel position and y is obtained with the previous formula.

The global distortion table columns are labeled a00, a01, a02, a10, a11, a20, indicating the coefficients of the fitting bivariate polynomials.

The global distortion table is produced by the *vmmoscalib* recipe with the tag GLOBAL_DISTORTION_TABLE (see Section 7.11.2, page 107).

A global distortion table doesn't depend on the filter in use: only the grism and the chip matter. In practice, the correct global distortion table can be associated to a given scientific frame using the FITS keywords ESO INS GRIS1 NAME and ESO INS CHIP1 ID, found both in the table and in the raw input frames headers: but this is relevant for online processing (on Paranal) only. In the offline reduction, never input any global distortion table to the *vmmosscience* recipe.

8.3.2 Zero order contamination model (obsolete)

Currently not implemented

8.3.3 Optical distortion model (obsolete)

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 8.4.

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 8.4: *Optical distortion model coefficients.*

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 spatial 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.

8.3.4 Spatial curvature model (obsolete)

The spectrum corresponding to the position (x, y) on the mask traces a curve on the CCD. The spatial 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 8.3.3 and Figure 8.2).

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*. 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 8.5.

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 8.5: *Global curvature model coefficients.*

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.

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.

8.3.5 Inverse dispersion solution (obsolete)

As seen in Section 8.3.3, 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):

$$\Delta Y = \sum_i d_i \Delta\lambda^i$$

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*.

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 8.6.

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.

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 8.6: *Global inverse dispersion solution coefficients.*

8.3.6 Slit spectra extraction (obsolete)

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 distortions.

9 Algorithms

In this section the data reduction procedures applied by the 26 pipeline recipes (21 currently in use, see Section 4.1) are described in some detail. Common algorithms, as cosmic rays removal or bad pixel cleaning, are described separately.

9.1 Stacking

The stacking module used by the VIMOS-IMG pipeline recipes for the science observations has two modes of operation, a `slow` and a `fast` algorithm. The algorithm needs to stack not just the science data, but also the science variance data. It creates an output stack, output stack variance and an output stack confidence map. The volume of data that is read and written can make the memory required quite large and this can be increased further if the jitter offsets are large too.

The 'fast' algorithm can be used when stacking a small number of images. As the name implies it is usually pretty quick, but it is very greedy with memory. For stacking problems with more images it is better to use the slow algorithm. The recipes that use the stacking module also offer an `auto` mode. This allows the recipe to decide which algorithm to use. If the number of input frames is less than or equal to `stk_nfst`), the 'fast' mode is used.

When stacking, an RA/DEC is calculated for each input pixel. If `stk_method = nearest`, the pixel in the output stack whose centre is nearest to that RA/DEC is given that input pixel. If `stk_method = linear`, the exact pixel location of the RA/DEC on the output grid is calculated and the contribution from the input pixel is spread to the neighbouring 4 pixels that surround that point and is weighted by the fractional overlap.

Changes in throughput (i.e. magnitude zero point) and/or seeing between consecutive raw science frames within an OB are not taken into account as part of the stacking procedure. Users are advised to check for changes in seeing or photometric conditions in each image before stacking; no warning are issued by the recipe. Aberrant images may be discarded from the input list to avoid unexpected results.

9.2 Variance maps

All science recipes in the VIMOS-IMG pipeline package propagate the uncertainty of the flux of each pixel in each of the science images. This is done using the standard equations for the propagation of uncertainties. The initial estimate of variance is based on the standard Poisson model using the gain to convert ADUs to electrons. If a pixel is flagged as bad, then its variance estimate is defined as zero.

9.3 Confidence maps

A confidence map is similar to a weight map. These are initially derived from the master twilight flat field for the detector and hence gives an indication of the relative quantum efficiency of each pixel, rather than an indication of the Poisson uncertainty in the pixel flux. The map is stored as an integer array that has been normalised to a mean of 100. Bad pixels (either dead or hot) are given a confidence value of zero. If images are stacked, then the confidence map is propagated and can be used as a weight map or an exposure time map.

9.4 WCS fitting

Cartesian and equatorial coordinates are fitted to standard plate solution models with 6 constants that account for non-zero shear and scale differences. The difference in the predicted x,y coordinates and the true x,y coordinates is used to adjust the tangent point first to correct for telescope pointing error (i.e. `RA` and `DEC` keywords in raw file PHU) . The median difference of the equatorial coordinates between that implied from the two sets of Cartesian coordinates is used to update the tangent point. A full least-squares solution is performed and the results are written back to the given FITS WCS header structure (using a TAN projection).

For a 6 constant model, fits are done with the input standards for the equations:

$$\xi = ax + by + c \tag{1}$$

$$\eta = dx + ey + f \tag{2}$$

where ξ, η are standard coordinates with respect to the tangent point (transformed onto the detector system), to find values of a, b, c, e, d and f.

9.5 Detector readnoise and gain

The read noise and gain is measured using two twilight flat frames of similar illumination and two bias frames. First, a robust estimate of the variance in an image of the difference between the two flat frames is calculated (σ_f^2). The same is done for the bias frames (σ_b^2). A correction for any inter-pixel correlation is made by evaluating the scalar auto-correlation among pixels in the flat frames (ξ). If the mean background of the flat and bias frames are m_{f1}, m_{f2}, m_{b1}, and m_{b2}, the gain (in electrons per ADU) is:

$$g = \frac{1}{\xi} \frac{(m_{f1} + m_{f2}) - (m_{b1} + m_{b2})}{\sigma_f^2 - \sigma_b^2} \tag{3}$$

and the readout noise (in electrons) is:

$$\sigma_{ro} = \frac{g\sigma_b}{\sqrt{2}} \tag{4}$$

9.6 Object detection/catalogue generation

In order to provide quality control, and astrometric and photometric calibration information, the recipes generate detected object (i.e. stars, galaxies) catalogues for each target frame.

Objects are detected and parameterised using the processed images and confidence maps. A high-level summary of this process is:

- estimate the local sky background over the field and track any variations at adequate resolution to eventually remove them

- detect objects/blends of objects and keep a list of pixels belonging to each blend for further analysis (see [5] for details)
- parameterise the detected objects, i.e. perform astrometry, photometry and some sort of shape analysis.

9.6.1 Background analysis and object detection

The possibly-varying sky background is estimated automatically, prior to object detection, using a combination of robust iteratively-clipped estimators.

Any variation in sky level over the frame is dealt with by forming a coarsely sampled background map grid. Within each background grid pixel (specified by recipe parameter `stk_cat_nbsize`), an iteratively k-sigma clipped median value of 'sky' is computed based on the histogram of flux values within the grid pixel zone. A robust estimate of sigma can be computed using the Median of the Absolute Deviation (MAD) from the median. This will then be further processed to form the frame background map.

After removing the (possibly) varying background component, a similar robust estimate of the average sky level and sky noise per pixel can be made. This forms part of the quality control measures and also helps to robustly determine the detection threshold for object analysis.

Individual objects are detected using a standard matched filter approach. Since the only images difficult to locate are those marginally above the sky noise, assuming constant noise is a good approximation (after factoring in the confidence map information) and the majority of these objects will have a shape dominated by the point spread function (PSF), which thereby defines the filter to use.

9.6.2 Object parameterisation

The following image parameters are computed efficiently for each detected object:

Isophotal intensity: integrated flux within the boundary defined by the threshold level, i.e. the 0th object moment

$$I_{iso} = \sum_i I(x_i, y_i) \tag{5}$$

For objects where $I(x_i, y_i)$ is well described by a Gaussian, the isophotal intensity is related to the total intensity by the factor $(1 - I_t/I_p)^{-1}$, where I_p is the peak flux and I_t is the threshold level (all relative to sky).

Position: intensity-weighted location of object on the image, i.e. 1st moment:

$$x_0 = \sum_i xI(x_i, y_i)/I_{iso} \tag{6}$$

$$y_0 = \sum_i yI(x_i, y_i)/I_{iso} \tag{7}$$

Covariance matrix: triad of intensity-weighted 2nd moments used to estimate the eccentricity/ellipticity, position angle, and intensity-weighted size of an object:

$$\sigma_{xx} = \sum_i (x_i - x_0)^2 I(x_i, y_i)/I_{iso} \tag{8}$$

$$\sigma_{xy} = \sum_i (x_i - x_0)(y_i - y_0) I(x_i, y_i)/I_{iso} \tag{9}$$

$$\sigma_{yy} = \sum_i (y_i - y_0)^2 I(x_i, y_i)/I_{iso} \tag{10}$$

The shape parameters are derived by considering these quantities in relation to an elliptical Gaussian function with the same 2nd moments. The scale size ($\sqrt{\sigma_{rr}}$) is computed with $\sigma_{rr} = \sigma_{xx} + \sigma yy$. The eccentricity is $f = \sqrt{(\sigma_{xx} - \sigma_{yy})^2 + 4\sigma_{xy}}/\sigma_{rr}$. The position angle θ is computed with $tan(2\theta) = 2\sigma_{xy}/(\sigma yy - \sigma xx)$. The ellipticity ($e$) is simpler to interpret for estimation potential image distortions (e.g. trailing) and is related to the eccentricity by $e = 1 - \sqrt{(1-f)/(1+f)}$.

Areal profile: a variant on the radial profile, which measures the area of an object at various intensity levels. Unlike a radial profile, which needs a prior estimate of the object centre, the areal profile provides a single-pass estimate of the object area at m discrete intensity levels $T + p_j$ where $p_j; j = (1...m)$ are intensity levels relative to threshold T (usually spaced logarithmically to give even sampling). The threshold T can be specified using the recipe parameter `stk_cat_thresh`.

Peak height: a useful related addition to the areal profile information and is defined as $I_p = max(I(x_i, y_i))$ or alternatively measured by extrapolation from the areal profile if the image is saturated.

Seeing: The areal profile provides a direct method to estimate the seeing of objects in an image by enabling the average area of stellar images (point sources) at half the peak height ($< A >$) to be estimated. The seeing, or FWHM, is then given by $FWHM = 2\sqrt{< A > /\pi}$.

Aperture flux: a series of aperture fluxes are required for object morphological classification. Aperture flux is defined as the integrated flux with some radius r of the object centre:

$$I_{ap} = \sum_{i \in r}^{N} I_i - N \times sky \tag{11}$$

where boundary pixels are weighted pro-rate (soft-edged aperture photometry). A series of these is used to define the curve-of-growth (I_{ap} vs. r) for each object.

The error in the aperture flux is calculated as:

$$\sigma_{ap} = \sqrt{I_{ap}/gain + \pi r^2 (\sigma^2_{globalsky} + \sigma^2_{localsky})} \tag{12}$$

where $\sigma_{localsky}$ is a robust estimate of the sky variance among adjacent cells (defined by `stk_cat_nbsize`), $\sigma_{globalsky}$ is a robust estimate of the sky variance over the whole image, and the $gain$ is the value calculated by the `detector_noise` recipe.

In order to estimate the total flux of an object, an aperture correction must be applied. This correction is calculated from the curve-of-growth analysis and appears in the FITS header of the catalogue (`APCOR_i` is the correction to be applied to the ith aperture fluxaperture correction for each pa

To convert aperture fluxes to magnitudes, an aperture correction must be applied. This correction is calculated from the curve-of-growth and accounts for missing flux in a particular aperture. For example, the magnitude of an object using the flux in the third bin ($i = 3$) may be calculated as

$$m = \texttt{MAGZPT} - 2.5\log_{10}(\texttt{APER_FLUX_3}/\texttt{EXPTIME}) - \texttt{APCOR3}) \tag{13}$$

where `MAGZPT, EXPTIME,` and `APCOR3` appear in the FITS header of the catalogue file; `APER_FLUX_3` is a column in the FITS binary table of the catalogue file.

For overlapping objects that are deblended, the boundaries are in practice simultaneously fitted top-hat functions (to minimise the effects of crowding). Object external to the blend are also flagged and not included in the large radius summations.

9.6.3 Morphological classification

The recipes produce a series of background-corrected flux measures for each object in a set of 'soft-edged' apertures of radius $r/2, r/\sqrt{2}, r, r\sqrt{2}, 2r, ...12r$, where r is specified by the recipe parameter `stk_cat_rcore`. Generally it is recommended that `stk_cat_rcore` be fixed as the median seeing for the site + telescope + camera. For VIMOS-IMG, the default value of `stk_cat_rcore` is 10.0 pixels (2.05″). The average curve-of-growth for stellar objects is used to define automatically an aperture correction for each aperture used and also forms the basis for object morphological classification (required for isolating stellar images for seeing and trailing quality control).

The curve-of-growth of the flux for each object is compared with that derived from the (self-defining) locus of stellar objects, and combined with information on the ellipticity of each object, to generate the classification statistic. This statistic is designed to preserve information on the 'sharpness' of the object profile and is re-normalised, as a function of magnitude, to produce the equivalent of an $N(0, 1)$ measure, i.e. a normalised Gaussian of zero-mean and unit variance. Objects lying within 2-3σ are generally flagged as stellar images, those below $\approx 2\sigma$ (i.e. sharper) as noise-like, and those above 2-3σ (i.e. more diffuse) as non-stellar. A by-product of the curve-of-growth analysis is the estimate of the average PSF aperture correction for each detector.

9.7 Defringing

Atmospheric emission lines may cause interference fringes to be present in the sky background at the level of a few percent of sky. Since the fringes can have complex spatial structures on a range of physical scales on the detector, removing them successfully is a multi-stage process.

First we note that fringing is an additive effect, so if removed as part of a procedure that used night sky data as a flat field source, this would introduce a systematic error in the photometry. To perform sky fringe removal effectively requires the flat fielding to be decoupled from the defringing by, for example, using twilight sky exposures to construct the flat-field frames, where the contribution from sky emission lines is negligible.

Consequently, the first stage of the process is to flat-field the dark sky science data correctly and to use a sequence of offset sky exposures to construct a fringe frame. These input frames are combined after suitable scaling to match the background levels and sigma-clipping to remove astronomical objects.

The defringing process then requires solving for the fringe scale factor k in the following equation:

$$D(x,y) = S(x,y) + kF(x,y) + O(x,y) + T(x,y) \tag{14}$$

where S is the sky contribution, O is the astronomical object contribution and T is the contribution from the detector background.

Since the fringe pattern is characterised by more rapidly varying spatial structure than the sky and thermal contributions, the overall background variation on the target and fringe frame is temporarily removed by use of a robust low-pass filter such that:

$$D'(x,y) \approx kF(x,y) + O(x,y) \tag{15}$$

The objects are localised, therefore a simple robust background noise estimator based on the Median of the Absolute Deviation (MAD) from the median can be used iteratively to find the scale factor k that minimises the background noise in $D'(x,y)$. Allowing the scale factor to vary ensures that the relative contribution of the sky emission lines, which may vary in strength, is correctly dealt with.

More complex options involving decomposing the seasonal fringe patterns into eigen-fringe maps may be required at later stages in the processing but this is outside of the scope of the standard calibration pipeline.

9.8 Bad pixel cleaning for spectroscopy

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. All the VIMOS pipeline recipes allow bad pixel cleaning, with the exception of the new MOS recipes *vmmoscalib* and *vmmosscience*.

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.

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 9.1). 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 value to assign to the 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.

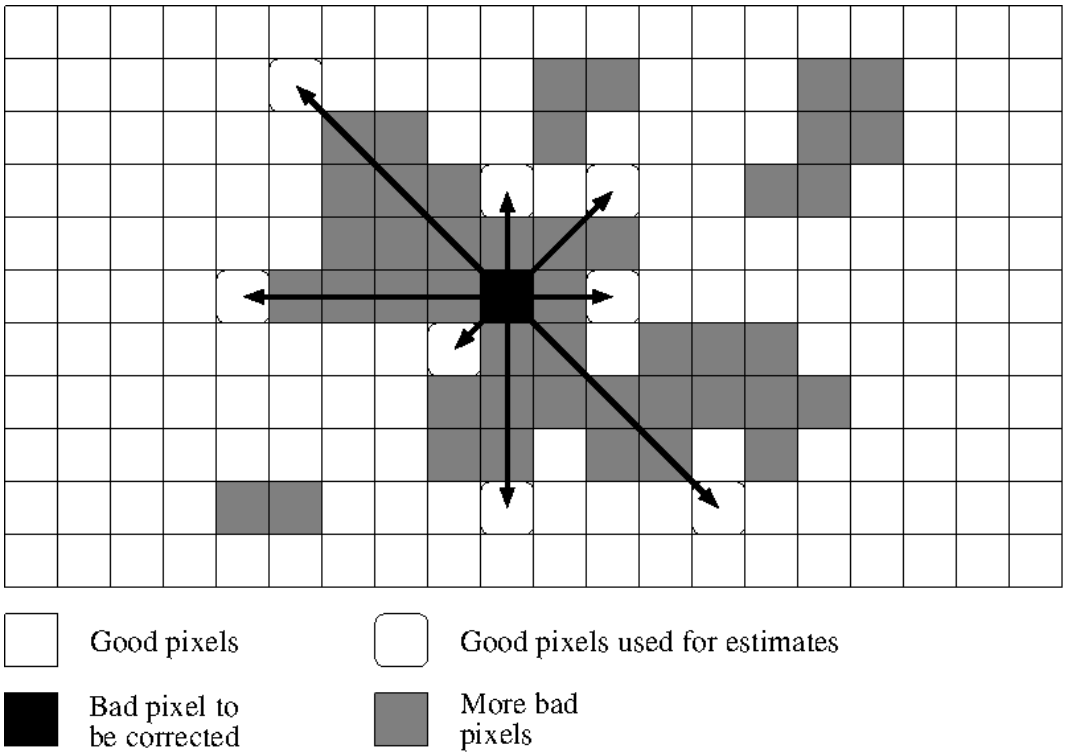

Figure 9.1: *Good pixels to be used in the estimate of a given bad pixel are searched along the indicated directions..*

9.9 Cosmic rays removal for spectroscopy

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 (which 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 and currently a value of 4.0 is used since it typically gives 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 currently set to 2.0, which is known to give 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 9.8. If a bad pixel table is also given to a recipe, then the bad pixels are avoided in the interpolation procedure.

9.10 Bias subtraction for spectroscopy

Removing the bias from any raw frame is a relatively simple process, but not simple enough to avoid a description on its own.

For the MOS recipes, the procedure is as follows:

- the overscan is subtracted from the individual bias frames.
- the readout noise for each individual frame is computed from the robust standard deviation in each of the detector ports. The robust standard deviation is obtained from the interquartile range as follows:

$$RON = \frac{third_quartile - first_quartile}{1.349} \tag{16}$$

  Since the raw images have only integer values, the robust standard deviation will be strongly quantized and the RON can take only a few discrete values.
- the master bias is combined using any of the available methods (mean, wmean, median, minmax, ksigma) and its error is propagated.
- the readout noise of each of the detector ports is computed as the mean of the individual frames readout noise and stored in the master bias under keywords ESO QC DET OUTi RON.

For the imaging and IFU recipes the master bias frame (`MASTER_BIAS`) is used iin a similar way, but the overscan subtraction is sligthly different: if 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. It is strongly recommended that *"Zmaster"* is always used.

9.11 Dark subtraction for spectroscopy

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 $e^- \cdot$ h$^{-1} \cdot$ pixel^{-1}), so this operation would be in most cases more harmful than helpful.

9.11.1 Flat normalisation for spectroscopy

The flat normalisation is applied on a slit per slit basis after the master flat is rectified from spatial distortion. The algorithm calculates, for each slit, one dispersion profile and n_s spatial profiles where n_s is either the number of rows of the rectified slit (n_r) or 1. Then, the algorithm combines (multiples) the two profiles to generate a profile image. The profile image is then de-rectified to go back to CCD pixels space. Finally, the original master flat is divided by the de-rectified profile image.

The generation of the dispersion profile follows these steps if smoothing or fitting are enabled:

- The slit image is collapsed along the spatial axis to generate a dispersion profile;
- If smoothing is enabled, the profile is smoothed using a 1-D median filter with the specified radius;
- If cubic spline fitting is enabled a cubic bspline with the specified number of knots is fitted to the profile. Only values higher than the specified threshold are used for the fit (note that the threshold is specified relative to the highest value in the profile).

If neither smoothing or fitting are enabled, the dispersion profile is a constant.

The generation of the spatial profile can be done row-by-row, hence $n_s = n_r$ or can be done collapsing the slit image along the dispersion axis, hence $n_s = 1$. The generation of the spatial profile for row i follows these steps if smoothing or fitting are enabled:

- The i-th row is extracted from the slit image to generate the i-th spatial profile if row-by-row normalisation is enabled. Otherwise the slit image is collapsed along the dispersion axis to generate the spatial profile;
- If smoothing is enabled, the profile is smoothed using a 1-D median filter with the specified radius;
- If polynomial fitting is enabled a polynomial function is fitted with the specified degree to the profile. Only values higher than the specified threshold are used for the fit (note that the threshold is specified relative to the highest value in the profile).

If neither smoothing or fitting are enabled, the spatial profile is a constant.

Note that the radius used in the 1-D median smoothing cannot be greater than half the size of the profile. In case of violation of this constraint, the radius is reduced to half the size of the profile. The radius is also progressively reduced near the edges.

It has to be noted, also, that the two profiles are rescaled for dimensionality reasons before being combined. This implies, for example, that the user cannot rely on the average level of the per-row normalization to remove effects along the dispersion direction.

Note also that smoothing and fitting can be enabled at the same time: in such a case smoothing is always performed first.

9.12 Frame combination for spectroscopy

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 the number of input frames should be greater than the number of rejected values.

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. The procedure can be iterated. In this case, at least four input frames are required.

9.13 Blind arc lamp lines identification

Starting with the VIMOS pipeline release 2.0, a new method for arc lamp line identification is applicable. This method is the same which has been integrated starting from version 2.5 in the new recipe *vmmoscalib* (see Section 9.20, page 179).

This method may turn useful in the reduction of data for which the available first-guess distortion models turn out to be too inaccurate (perhaps due to mechanical instabilities of the instrument), or even missing (as it would be the case, for instance, with data obtained with a new grism).

In order to work, this method just requires a rough expectation value of the spectral dispersion (in Å / pixel), and a line catalog. The line catalog should include *just* lines that are expected somewhere in the CCD exposure of the calibration lamp.[31]

From the arc lamp spectra extracted following the available spatial curvature model, arc lamp lines candidates are selected and their positions on the CCD are determined. Typically, the arc lamp lines candidates will include light contaminations, cosmic rays hits, and other unwanted signal, but only in very extreme cases this will prevent the pattern recognition algorithm to identify all the spectral lines.

Currently any portion of the arc lamp spectrum peaking above a given threshold (measured relatively to the local background level) is selected as an arc lamp line candidate. In general the default threshold applied by the pipeline works well, but in some cases it may be helpful to lower it, in order to catch more (faint) lines, or to raise it, in order to avoid a too noisy background or a large number of fainter contaminations (coming perhaps from the second order dispersion of a multiplexed spectrum). Care should be taken in avoiding too low

[31] The line catalog represents the pattern that should be searched on the CCD, and adding extra lines would destroy this pattern. Note, however, that a catalog including extra lines at its blue and/or red ends would still be acceptable.

thresholds, that would pick up too much noise from the background, or too high thresholds, that would miss one or more valid arc lamp lines, leaving the pattern recognition algorithm without a pattern to match.

As a general rule, it is important to ensure that (almost) all the line catalog entries have their counterpart on the CCD.[32] In practice, it is safe to have more candidates on the CCD than lines in the input line catalog: the pattern recognition would succeed even in the case the spectra on the CCD contained more arc lamp lines than actually listed in the input line catalog. In this case, however, the tracing of the dispersion relation may not be sufficiently fine.

9.14 Sky fringing correction for spectroscopy

From a jittered observation, both in imaging and in MOS mode, a map of the sky can be obtained by median-stacking the input exposures (see Section 9.12, page 170). This sky map can then be subtracted from each input exposure before any further processing. Naturally, if a fringing pattern is present it will be eliminated from the data with the sky.

This method is not always applicable: if the observed field is too crowded and/or the jittering step is smaller than the extension of any of the observed objects, unwanted "ghosts" would appear on the sky map. It is important to carefully examine the sky map produced by the data reduction recipe, before trusting the results of the fringing correction.

It should also be noted that subtracting a sky map created by median-stacking the input exposures will increase the random noise on the reduced data. In order to minimise this effect, it is advisable to apply the fringing correction only if a large number (say, at least 5) of jittered exposures obtained at *different* offsets is available. That would also help to reduce the probability of "ghosts" on the generated sky map.

Another problem with this method is coming from the sky brightness changing during the observation. This is particularly important in the case of MOS observations, where the exposure times are typically much longer than in imaging, and the different components of the sky spectrum (continuum and emission lines systems) may display different variability patterns.

Nothing is done yet to keep into account a possible sky variability in the case of imaging observations. In the case of MOS observations, instead, systematic residuals are eliminated following different strategies, depending on the fringing correction method applied:

Raw method: A sky + fringes map is created from the median stacking of all the bias subtracted input frames. If the sky changed between exposures, the subtraction of the median sky will leave in general some sistematic residuals on each data frame. Such residuals are removed at a later step by the sky modeling task, that blindly will treat them as "sky".

Resampled method: This method differs from the *Raw* method only with respect to the point at which the median stacking of frames is applied. In this case the median map is created after the slit spectra are resampled at a constant wavelength step and the sky removed by the standard sky modeling task. This map is then subtracted from each processed frame.

Ideal method: This method differs from the *Resampled* method by applying the sky subtraction and by determining and subtracting the median residual map from each frame *before* the resampling.

[32] Losses on the red and/or blue ends are unimportant, however, because even parts of the pattern can be safely identified.

The recipe *vmmosobsjitter* (deprecated) offers the first and the second methods, while the recipe *vmmosscience* only offers the third one.

The median map obtained with the *Resampled* and the *Ideal* methods should generally be regarded as a residual map, rather than a fringing map. In fact, the sky modeling task is applied in this case to a sky that still includes its own fringes: if a median sky level were computed at each wavelength, its value would depend on the changing position of the object along its slit, while if the sky level is computed by polynomial interpolation, the polynomial would tend to fit both sky and fringes.

The choice of what method to apply is not too simple. Clearly, if the sky displays no variability during the observation the *Raw* method is the theoretically correct one, and should therefore be preferred: not only because the object detection task is more efficient if applied to spectra where sky fringes have already been removed, but also because with the *Resampled* method an apparent sky variability may be introduced, even when the sky is perfectly constant.[33]

On the other hand, if the sky spectrum is strongly changing between exposures, the *Resampled* method tends to give the best results, as the above mentioned drawbacks are outweighted by the advantage of evaluating and removing the changing sky on a frame-by-frame basis.

Note however that the *Ideal* method performs nearly equally well in both situations.

Probably the only safe way to proceed is to reduce the data applying all methods, and judge *a posteriori* the quality of the sky fringing removal on the extracted object spectra.

9.15 Computation of the spectral efficiency and response

The efficiency of an instrument is defined as the ratio between detected photons and incoming photons, and can be derived by comparing the observed fluxes with the tabulated fluxes of known objects.

The spectral efficiency of the VIMOS instrument is obtained in the following way:

The extracted standard star spectrum, $X(\lambda)$ is converted into e^- s^{-1} Å^{-1}:

$$S(\lambda) = \frac{g\,X(\lambda)}{t\,\Delta\lambda}$$

where g is the gain factor in e$^-$ / ADU, t the exposure time in seconds, and $\Delta\lambda$ the constant wavelength step at which the spectrum was resampled after its calibration in wavelength.

The magnitude losses $\Delta m(\lambda)$ listed in the column `EXTINCTION` of the atmospheric extinction table (see Table 7.10, page 124) are turned into flux losses, and applied to the observed spectrum:

$$S_o(\lambda) = S(\lambda)\,10^{0.4\,a\,\Delta m(\lambda)}$$

where $S_o(\lambda)$ is the spectrum at airmass zero and a is the airmass of the standard star observation. The values of the atmospheric extinction are linearly interpolated from the tabulated values for all the wavelength of the

[33] Since in different exposures the object has different positions along the slit, then the sky evaluation will be biased by different parts of the fringing profile.

observed spectrum. At those wavelengths where no atmospheric extinction data are available, $S_o(\lambda)$ is set to zero.

The standard star catalog fluxes $C(\lambda)$, given in erg cm^{-2} s^{-1} Å^{-1}, are converted into photons collected by the telescope using

$$F(\lambda) = C(\lambda)\,\frac{A_t}{h\nu}$$

Considering $A_t = 5.18\ 10^5$ cm^2 the VLT efficient area, and $hc = 1.98\ 10^{-8}$ erg Å, one derives

$$F(\lambda) = 2.6\ 10^{-3}\ C(\lambda)\ \lambda$$

(expressed in photons s^{-1} Å^{-1}, as the 10^{-16} factor is included in the numeric part).

The efficiency is finally computed as

$$E(\lambda) = \frac{S_o(\lambda)}{F(\lambda)}$$

(electrons per photon).

The efficiency $E(\lambda)$ is set to zero at those wavelengths where no standard star catalog fluxes are available, and is written to the column `RAW_EFFICIENCY` of the output spectro-photometric table (see Table 7.27, page 149).

The response curve used in the flux calibration of observed scientific spectra (see next Section) is obtained by:

$$R(\lambda) = \frac{C(\lambda)}{S_o(\lambda)}$$

$R(\lambda)$ is set to zero where no standard star catalog fluxes are available, and is written to the column `RAW_RESPONSE` of the output spectro-photometric table.

Since pipeline version 3.x, the intervals $\Delta\lambda$ used to compute the response are those of the tabulated data. This means that the observed spectra are binned to the bin intervals defined in the [STD_FLUX_TABLE] tables.

As a final step either a cubic spline fit or a polynomial fit performed, in order to derive a smoothed version of the curves. If parameter `resp_fit_nknots` is greater than 0, then a spline fitting is performed with the parameter specifiyng the number of knots. If parameter `resp_fit_degree` is greater than 0, a polynomial fit is performed with the parameter specifiyng the polynomial degree. The two options are mutually exclusive. The fit is performed on the response points that have not been excluded from parameters `resp_ignore_points`. Additionally, all points below 0.1% of the mean value of the response are ignored. If the number of remaining points is less than `nknots+2` in the case of spline fitting or less than `degree+1` in the case of polynomial fitting, the fitting algorithm parameters will be adjusted.

The fitted curves are written to the output spectro-photometric table, in columns `EFFICIENCY` and `RESPONSE`.

Some grisms, notably the holographic grisms, have different responses depending on the incident angle. That means that slits placed at different locations will show different behaviour. It has been demonstrated that this

can be corrected by first dividing each spectrum by the average spectral energy distribution of the corresponding slit flat. The *vmmosscience* recipe supports this mode if the input frameset contains the MOS_FLAT_SED file. Both during response computation, and spectrophotometric calibration, the extracted spectrum will be divided by the corresponding flat field spectral energy distribution. Obviously, this will work only if this is done in both steps. The recipe will issue a warning if one tries to calibrate spectrophotometrically a spectrum corrected with a flat field sed with a response that does not include the correction or vice versa. Note that the MOS_SPECPHOT_TABLE product contains the keyword **QC RESP FLAT SED_NORM** that is set to true if the response contains the flat sed correction.

Note that the flat field sed has been normalised by the value of the sed at the reference wavelength divided by the slit length, slit width (only known if *slit_ident=true*, see section 7.11.3) and flat field exposure time. This normalisation factor is stored in the keywords **QC FLAT SEDi NORM**, where i corresponds to the slit number.

In the case of flat field sed correction the formulas above remain the same, but $X(\lambda)$ is now:

$$\frac{X(\lambda) = X_{obs}(\lambda) sed_{std_norm}}{sed(\lambda)} \tag{17}$$

9.16 Spectrophotometry flux calibration

Scientific spectra extracted by the recipes *vmmosscience* and *vmifuscience*, can be calibrated in flux by specifying an atmospheric extinction table (see entry EXTINCT_TABLE, page 124), and an appropriate spectrophotometric table (see entry MOS_SPECPHOT_TABLE, page 131 and entry IFU_SPECPHOT_TABLE, page 7.27). The table must be derived from a standard star observation made with the same grism, the same filter, the same VIMOS quadrant, and the same instrument mode (IFU or MOS). Note that for MOS the spectrophotometric table is also a product of the same *vmmosscience* recipe, when a standard star exposure is specified in input. For IFU, the recipe is *vmifustandard*

An extracted scientific spectrum, $X(\lambda)$, is flux calibrated in the following way:

The spectrum is first converted into e^- s^{-1} Å^{-1}:

$$S(\lambda) = \frac{g\, X(\lambda)}{t\, \Delta\lambda}$$

where g is the gain factor in e$^-$ / ADU, t the exposure time in seconds, and $\Delta\lambda$ the constant wavelength step at which the spectrum was resampled after its calibration in wavelength.

The magnitude losses $\Delta m(\lambda)$ listed in the column `EXTINCTION` of the atmospheric extinction table are turned into flux losses, and applied to the observed spectrum:

$$S_o(\lambda) = S(\lambda) 10^{0.4\, a\, \Delta m(\lambda)}$$

where $S_o(\lambda)$ is the scientific spectrum at airmass zero and a is the airmass of the scientific observation. The values of the atmospheric extinction are linearly interpolated from the tabulated values for all the wavelength of the observed spectrum. At those wavelengths where no atmospheric extinction data are available, $S_o(\lambda)$ is set to zero.

Finally, the flux calibrated spectrum is derived as

$$C(\lambda) = S_o(\lambda)\, R(\lambda)$$

where $R(\lambda)$ is the content of the `RESPONSE` column in the specified spectro-photometric table (see section 9.15, page 173).

The accuracy of the flux calibration mainly depends on the instrument efficiency and other external factors (as the timing of the shutters, the slit width, the seeing conditions and the airmass during both the scientific and the standard star observations).

9.16.1 Flux calibration for holographic grisms

The holographic grisms used in VIMOS show different response behaviour depending on the incident angle. This behaviour unfortunately means that one should place the standard star slit in the same Y coordinate as the science target. Fortunately, it is possible to flux calibrate a slit using a standard star observed in a different slit by taking the ratio of the relative response between the two positions. The flat fields of both the standard star and the science target can be used to determine this relative response.

The *vmmosscience* recipe allow the flat field sed correction when doing flux calibration given that a MOS_FLAT_SED input is given. This must correspond to the flat field associated to the science. The formulae described in section 9.16 are still valid provided that $X(\lambda)$ is subtituted by:

$$\frac{X(\lambda) = X_{obs}(\lambda) sed_{std_norm}}{sed(\lambda) sed_{sci_norm}} \tag{18}$$

Note that in this case the response has to be also corrected from flat field sed as explained in section 9.15. In this case the column used for the response is called REPONSE_FFSED. The recipe will issue a message if one attempts to use the flat field sed correction with a response that has not been corrected by the same effect.

9.17 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.

9.17.1 Read-out-noise determination for spectroscopy

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`.

9.17.2 Bias subtraction for spectroscopy

The master bias is subtracted from each of the flat field using the *"Zmaster"* method, described in detail in Section 9.10.

9.17.3 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 9.17.1.

9.17.4 Bad pixels identification for spectroscopy

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.

9.18 vmbias

This recipe carries out the following fundamental steps:

1. Optional cosmic rays removal (see Section 9.9).
2. Combination of input bias frames (see Section 9.12).
3. Optional bad pixel cleaning from output master bias frame (see Section 9.8).

The details of each step are explained in the specified sections.

9.19 vmdark

This recipe carries out the following fundamental steps:

1. Bias subtraction (see Section 9.10).
2. Optional cosmic rays removal (see Section 9.9).
3. Combination of input dark frames (see Section 9.12).
4. Optional bad pixel cleaning from output master dark frame (see Section 9.8).

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).

9.20 vmmoscalib

A more detailed description of the instrument-independent tasks involved in the complete self-calibration procedure applied by the recipe *vmmoscalib* is given in the next sections. Here an overview is just provided, which is useful for setting the individual tasks in their appropriate context.

1. Retrieve from the reference arc lamp line catalog the line pattern to be searched on arc lamp exposures.
2. After bias and background subtraction, examine the arc lamp exposure one column at a time. For each CCD column:
   (a) Run the 1D peak-detection task (see Section 9.20.1, page 181), to produce a list of reference arc lamp lines candidates.
   (b) Run the 1D pattern-recognition task (see Section 9.20.2, page 182), to select from the list of candidates a list of identified peaks. Not all the arc lamp lines are expected to be always identified, because the spectra are presumably distorted, and some CCD rows may cross a spectrum just partially, or even miss it entirely (see Figure 9.2).
3. Apply a preliminary wavelength calibration to each CCD column, within the specified wavelength range.
4. Choose a reference wavelength (see Section 9.20.4, page 185).
5. Find the CCD position of each connected region of CCD pixels containing the reference wavelength.
6. Run the 2D pattern-recognition task (see Section 9.20.6, page 186), to match the physical positions of the slits on the focal plane with the positions found on the CCD for the reference wavelength.
7. If requested, and if there are enough slits, fit a transformation between slits positions and CCD positions, and upgrade the list of reference positions on the CCD (see Section 9.20.7, page 188).
8. Trace the edges of each flat field spectrum, starting from the found positions of the reference wavelength (see Section 9.20.8, page 188).
9. Fit the traces with a low-degree polynomial (see Section 9.20.9, page 188). If requested, and if there are enough slits, fit also a global model of the obtained coefficients.

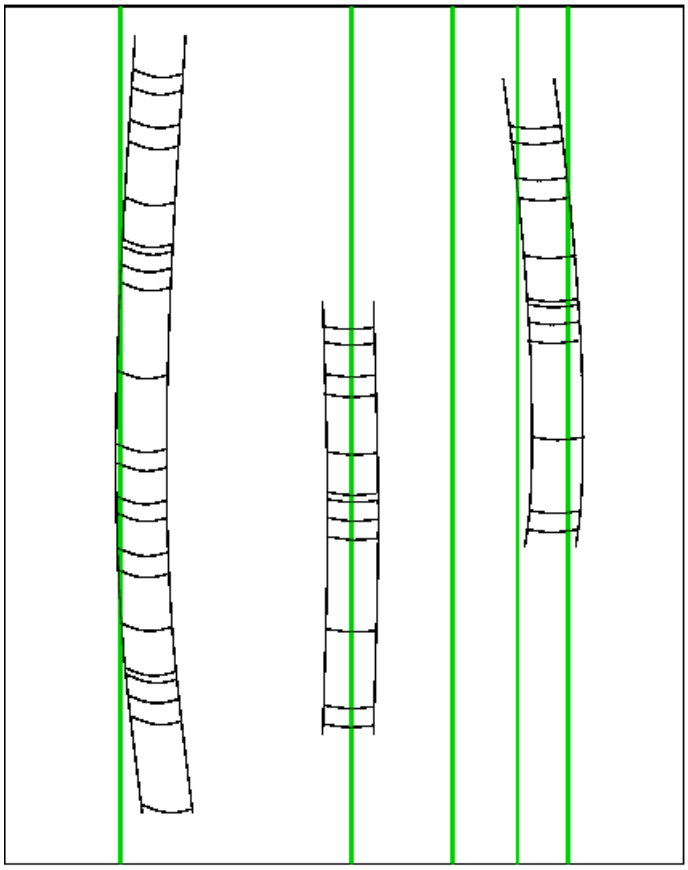

Figure 9.2: *CCD columns may not cut the whole range of the raw arc lamp spectra, because the spectra are not read along their curvature. However, even incomplete portions of the searched pattern can be identified by the pattern-matching algorithm..*

10. Extract the arc lamp spectra following the determined spatial curvature (interpolating fluxes along the spatial direction), and store it (row wise) in the rectified image. For each row of each rectified arc lamp spectrum:
    (a) Run the 1D peak-detection task on the extracted spectra, to produce a list of reference arc lamp lines candidates from the *whole* spectral range.
    (b) Run the 1D pattern-recognition task, using the pattern from the line catalog, to select from the list of candidates a list of identified peaks.
    (c) Fit a relation between the positions of the identified peaks *vs* the corresponding wavelengths.[34]
11. If requested, and if there are enough slits, fit also a global model of the obtained coefficients, in order to improve the local solutions.

At this point the spectral extraction mask is completely determined, and for each spectrum a specific coordinate system is defined, where to a CCD pixel correspond a wavelength and a position on the telescope focal plane. If

[34]This is the local wavelength calibration.

the instrument were stable, it would be possible to extract the scientific spectra applying directly this extraction mask. In general, however, the extraction mask obtained from the day calibration exposures should be aligned to the scientific spectra before being applied.

The extraction mask is also used to apply the flat field normalisation procedure described in Section 9.11.1, page 170.

9.20.1 1D peak-detection

Many sophisticated methods are available for detecting peaks and determining their positions along a one-dimensional signal. Any one of them is in principle suitable for the 1D peak-detection task of an automatic MOS data reduction pipeline.

The most important thing to note, however, is that on a calibration approach based on pattern-recognition the strongest requirement is that *the searched pattern must be present in the data*.[35]

In a traditional approach, peaks are initially rejected by the peak-detection task (depending on their statistical significance), and finally by the model fitting task (if they are found to be outliers). But in case a pattern-recognition algorithm is applied, the significance of a peak is primarily established by its *being part of the expected pattern*.

For this reason virtually any flux excess – no matter how significant – should be flagged as a peak candidate.

In the specific case of arc lamp spectra, the emission lines are very well exposed, and the S/N ratio of the lines to detect is almost always very high. This makes possible to apply a very simple 1D peak-detection method, based on the following two statements:

Any local maximum identifies a peak: in other words, a peak is identified by any pixel that is preceded and is followed by one pixel with a lower value (see Figure 9.3).

A peak position is determined by parabolic interpolation of the three found pixel values: if a local maximum is found, the central pixel and its two neighbours are interpolated by a parabola. The position of the parabola's vertex is taken as the position of the peak (see Figure 9.4). A peak position is then improved by applying more accurate methods: but if such methods fail, for instance finding positions that are significantly different from the parabolic ones, the original peak position is kept.

Even if obvious background noise fluctuations are excluded from the list of found peaks (e.g., by requiring that the values of the local maxima are greater than a given threshold), it is clear that with this method any contamination, hot pixel, cosmic ray, etc., would be reported as a "peak". This fulfills the critical requirement for the 1D pattern-recognition task reported above (see also Section 9.20.2, page 182).

The position x of a peak is given by

$$x = x_o + R$$

where x_o is the (integer) position of the pixel corresponding to a local maximum, and R the offset corresponding to the position of the maximum obtained by parabolic interpolation:

$$R = \frac{1}{2}\left(\frac{v_1 - v_{-1}}{2v_o - v_1 - v_{-1}}\right)$$

[35] Or at least long uninterruped portions of it.

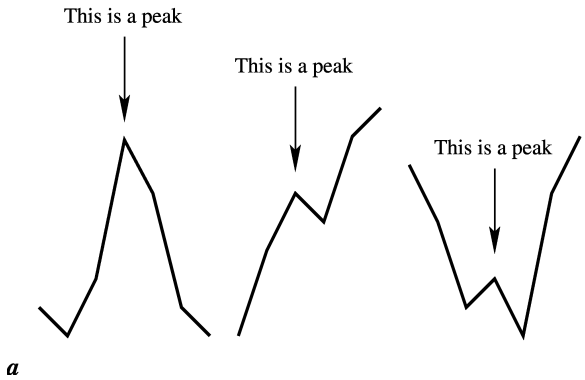

Figure 9.3: *Any local maximum identifies a peak.*

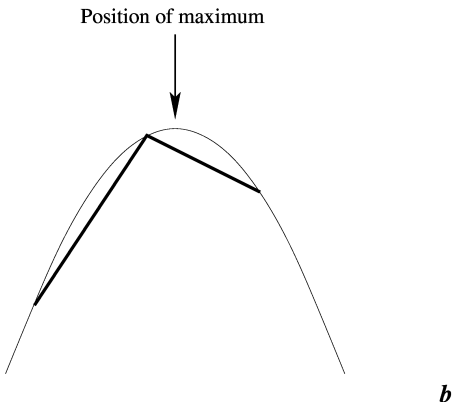

Figure 9.4: *A peak position is determined by parabolic interpolation of three pixel values about the local maximum.*

where v_{-1}, v_o, and v_1 are the values of the pixels $x_o - 1$, x_o, and $x_o + 1$, always fulfilling $v_{-1} \leq v_o$ and $v_1 < v_o$, or $v_{-1} < v_o$ and $v_1 \leq v_o$ (see Figure 9.5).

The quantity R never diverges, and does not depend on the background level (assuming that the background level is the same for the three pixels).[36]

9.20.2 1D pattern-recognition

A simple method for 1D pattern-recognition has been developed in the attempt to increase the robustness of the wavelength calibration, despite possible mechanical instabilities of the instrument.

In order to work, this method just requires a rough expectation value of the spectral dispersion (in Å/pixel), and a line catalog. The line catalog should just include lines that are expected somewhere in the CCD exposure of the calibration lamp.[37]

[36] In the case of very wide slits, the emission lines profiles display a flat top that would prevent the direct application of this method. This is somehow minimised by the preliminary application of a box filter as wide as the lines widths.

[37] The line catalog represents the pattern that should be searched on the CCD, and adding extra lines would destroy this pattern. Note,

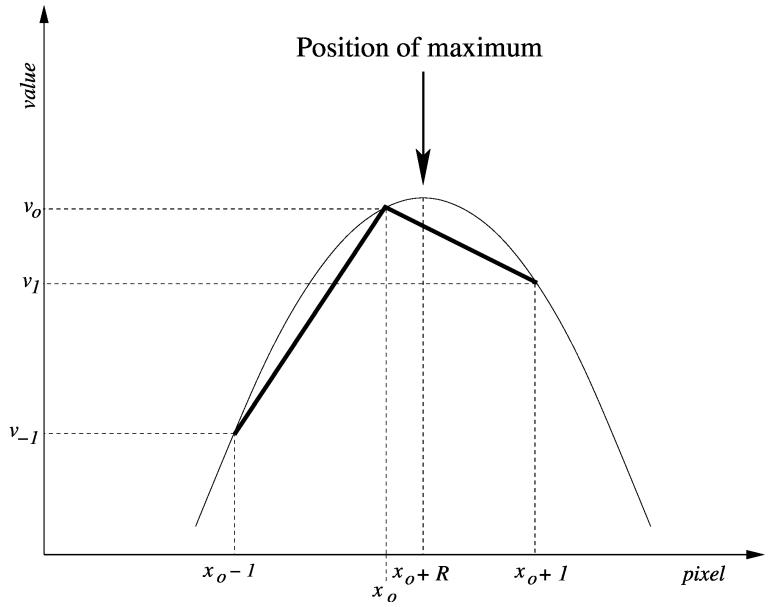

Figure 9.5: *Peak position estimate.*

The line-pattern would be searched in the list of CCD positions of arc lamp lines candidates produced by the 1D peak-detection task (see Section 9.20.1, page 181). Typically, the arc lamp lines candidates will include light contaminations, hot pixels, and other unwanted signal, but only in extreme cases this prevents the pattern-recognition algorithm from identifying all the reference lines. The pattern is detected even in the case the spectra contained more arc lamp lines than actually listed in the input line catalog. In particular, this method is not deceived by spectral multiplexing, even in case of significant spectral overlap: all spectra are identified as separate instances of the same pattern.

This method is based on the assumption that the relation between wavelengths and CCD positions is with good approximation *locally* linear.[38]

The ratio between consecutive intervals in wavelength and in pixel is invariant to linear transformations, and therefore this quantity can be used in the recognition of *local* portions of the searched pattern. All the examined sub-patterns will overlap, leading to the final identification of the whole pattern.

Let be:

d: a rough value of the expected spectral dispersion (Å/pixel).

Δd: a tolerance value on the expected dispersion, large enough to ensure that, at all wavelengths, the real spectral dispersion will be included in the interval from $d - \Delta d$ to $d + \Delta d$.

W: the number of wavelengths in the input line catalog.

however, that a catalog including extra lines at its blue and/or red ends is still allowed.

[38]This is generally true for modern spectrographs, but if this were not the case the detected peaks positions may be preliminary transformed to roughly approach linearity, before being processed and identified by the pattern-matching task described here.

N: the number of detected peaks.[39]

λ_i: the i-th wavelength of the input line catalog, with $1 \leq i \leq W$.

p_j: the position of the j-th peak, with $1 \leq j \leq N$.

All the arc lamp wavelengths λ_i are taken one by one, excluding the first and the last wavelengths ($i = 1$ and $i = W$). The ratio R_i of the wavelength difference with the preceding and the following wavelength is computed:

$$R_i = \frac{\lambda_{i+1} - \lambda_i}{\lambda_i - \lambda_{i-1}}$$

The same ratio is now searched in the list of peak positions: for each i, all the peak positions p_j are checked, excluding the first and the last one, taking care however to exclude from the computation any interval that would be incompatible with the expected spectral dispersion. This is done in the following way: for each considered p_j, the following *forward* search interval p_{min} to p_{max} is defined (see Figure 9.6):

$$p_{min} = p_j + \frac{\lambda_{i+1} - \lambda_i}{d + \Delta d}$$

$$p_{max} = p_j + \frac{\lambda_{i+1} - \lambda_i}{d - \Delta d}$$

A *backward* search interval is similarly defined:

$$p_{min} = p_j - \frac{\lambda_i - \lambda_{i-1}}{d - \Delta d}$$

$$p_{max} = p_j - \frac{\lambda_i - \lambda_{i-1}}{d + \Delta d}$$

Any peak position included either in the forward or in the backward search intervals is used for computing a ratio of distances from the position p_j (analogous to R_i). Each time a computed ratio equals R_i,[40] the three peak positions used for the computation (one is p_j, another is one from the backward search interval, and another is one from the forward search interval) are assigned respectively the wavelengths λ_{i-1}, λ_i, and λ_{i+1}. This assignment is not final: the same wavelength may even be assigned to different peaks, and the same peak may be assigned to different wavelengths. Each time a wavelength is assigned to a peak, a counter is increased, to keep a complete record of the assignments of wavelengths to peaks. Some wavelength assignments might be mistaken, and therefore not confirmed by successive comparisons.[41] The peaks that at the end of the analysis display a high score with respect to a given λ are considered identified, while ambiguous scores are rejected. The identified peaks are submitted to specialised sorting tasks that order them into separate self-consistent sequences (to take care of possible spectral multiplexing). This completes the peak identification process. This procedure is surprisingly fast, and has been tested successfully with VIMOS spectroscopic data obtained with all the available grisms on all the instrument quadrants, both in MOS and IFU modes (i.e., using 48 independent instrument configurations), as well as all the FORS1 and FORS2 grisms in the LSS, MOS, and MXU intrument modes. All the arc lamp lines listed in the line catalog are correctly identified without relying on a pre-existing instrument distortion modeling.

[39] Note that, as said above, it is typically $N > W$, or even $N >> W$.

[40] Within a given tolerance: this tolerance should be large enough to account for any deviation of the real wavelength calibration from the local linear approximation. A preposterously large value of 5% is used successfully with all the VIMOS and FORS instrument modes.

[41] Note that each peak is examined more than once, as the loop on wavelengths proceeds, since it may be included in forward and backward search intervals of other peaks.

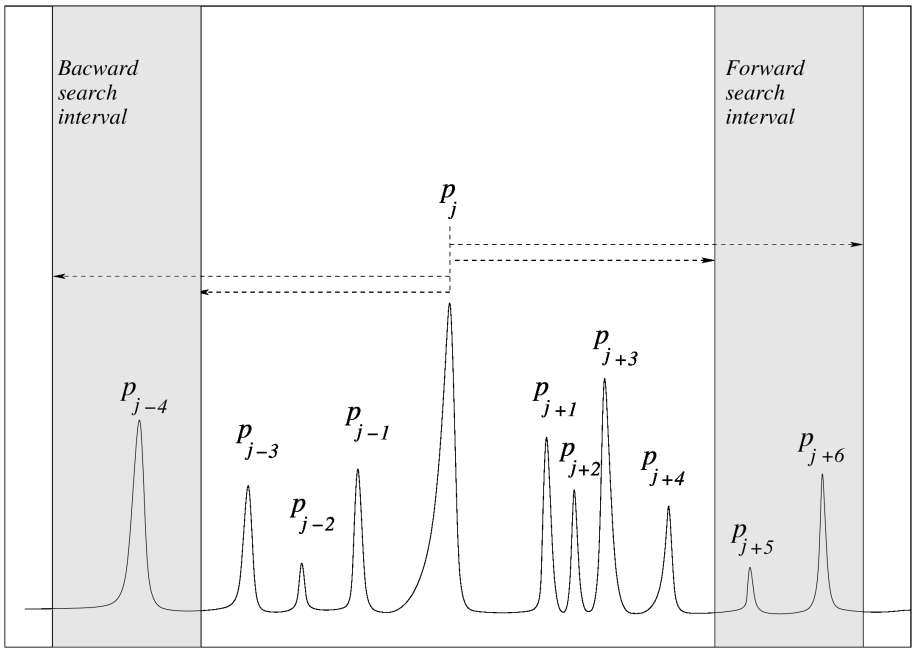

Figure 9.6: *Given a peak p_j, a forward and a backward search intervals compatible with an expected value of the spectral dispersion are defined. In this picture, peak positions $p_\{j-4\}$, $p_\{j+5\}$ and $p_\{j+6\}$ are used in the computation of the distance ratios to be compared with the wavelength interval ratio R_i. This process is repeated for each catalog wavelength and for each peak position, accumulating scores that will allow the final peaks identification.*

9.20.3 Determination of the spectral range

The spectral extraction range is specified by the user. A default range for each instrument configuration is given in the system configuration files (CONFIG_TABLE, see page 107).

9.20.4 Choice of a reference wavelength

The reference wavelength is just an arbitrarily chosen origin for spectral coordinates (both in wavelength and in CCD pixels), used in the definition of the wavelength calibration and of the spatial curvature models. Typically, a reference wavelength may be chosen at the center of the extracted spectral range.

However, if different spectral ranges are specified for the same grism, or if the spectral range is computed automatically (see previous Section), a different reference wavelength might be computed for different data reduction sessions.

9.20.5 Position of the reference wavelength on the CCD

One of the products of the 1D pattern-recognition task run on the extracted CCD rows (see this Section, page 179) is a boolean image, where all the pixels including the reference wavelength are flagged. After applying morphological operators for reducing the impact of occasional gaps in the data, the baricenter of all the con-

nected regions of flagged pixels is computed. The computed coordinates on the CCD should correspond to the positions of the slit centers on the mask plane. The match between the two sets is determined by the 2D pattern-recognition task (see next Section).

9.20.6 2D pattern-recognition

The 2D pattern-recognition method applied here is based on a point-matching algorithm, and it is used for matching positions on the telescope focal plane (mask) with positions on the instrument focal plane (CCD). It will then be possible to determine the transformation between the two coordinate systems, and to match each spectrum with its slit.[42]

Straightforward invariants to translation, rotation, rescaling, and reflection, are distance ratios and angles. In the method described here, distance ratios are preferred, and the reflection-invariance is dropped for reducing the risk of false matches.

For each of the two sets of points – the *pattern* set P, and the *observed* set D – all the possible triangles are constructed. The sides of each triangle are read clockwise,[43] and their lengths L_1, L_2, and L_3 are conventionally listed starting from the longest side (if two sides are equal, the first of the consecutive equal sides is taken). An ordered pair, (α, β), can be associated to each triangle, with

$$\alpha = \frac{L_2}{L_1} \qquad \beta = \frac{L_3}{L_1}$$

Such quantities are used to match similar triangles from both sets. The matches are made by associating nearby points on the $\alpha - \beta$ plane (see Figure 9.7). To each triangle are also assigned the coordinates of their vertices, and the applied normalisation factor L_1.

Initially, only safe matches are selected, corresponding to (α, β) bins containing just *one* triangle from each of the two input sets.[44] Such matches are used to get a first estimate of the scale factor, that is taken as the median of all the scale factors derived from the pairs of matching triangles,

$$S = \frac{(L_1)_P}{(L_1)_D}$$

At this point the complete list of triangles is revisited, eliminating all the matches that are incompatible with the found scale factor.[45] Finally, a rotation angle is computed for each matching pair, and incompatibilities with the median rotation angle are eliminated as well.

From the surviving triangles a list of matching points can be drawn and the geometrical transformation between the two sets can be determined. With the fitted transformation, points that were possibily lost to the matching procedure may be recovered, and a better transformation obtained from the extended sample.

It should be noted that this procedure, like the human brain, fails for regular grids of points: in fact in this case there would be no bin in the $\alpha - \beta$ plane containing just one triangle pair. Regular grids of points are typical of

[42] The 2D pattern-recognition is not applied if less than three spectra are detected on the CCD: in such cases, just local solutions would be used. Incidentally, a mask containing just one or two slits can hardly be considered a MOS mask.

[43] Imposing a reading order to the triangle sides eliminates the reflection invariance of the computed quantities.

[44] A preliminary test on set P would ensure that the pattern is not ambiguous, i.e., that isolated points on the $\alpha - \beta$ plane exists.

[45] In practice, a third dimension is added to the $\alpha - \beta$ plane, corresponding to the absolute size of the triangles in one of the two input sets.

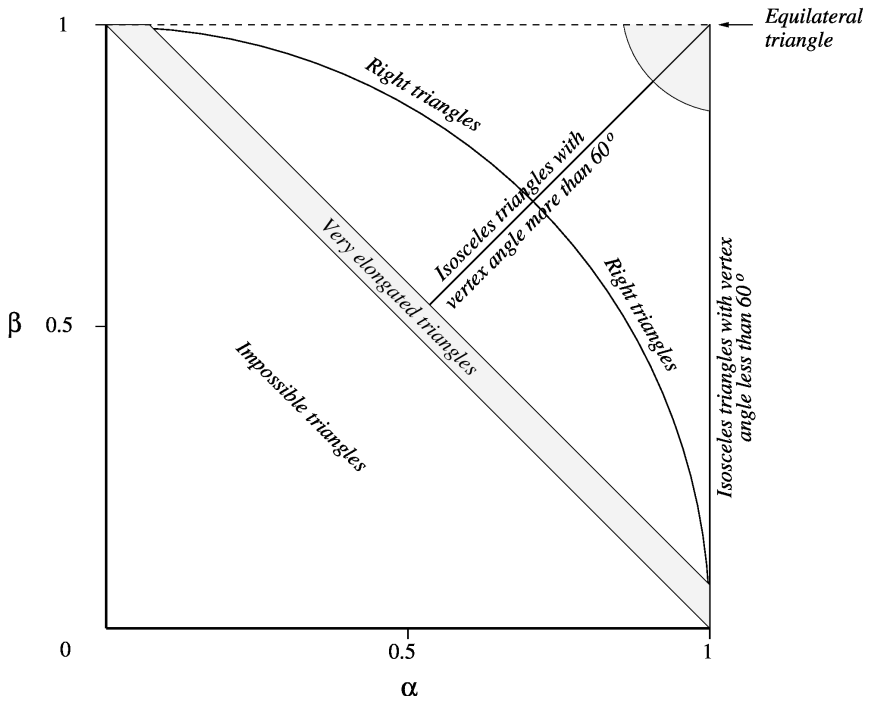

Figure 9.7: *The $\alpha - \beta$ plane. The long shaded region indicates very elongated triangles (including the cases of aligned points), while the region about the equilateral triangle includes ambiguous cases that would not lead to a safe identification of points: the triangles contained in those regions are therefore excluded from the analysis (unless they turn out to be the only triangles available). Note that the coordinates are cyclical: the line $\beta = 1$ (dashed) would include the same triangles described by the line $\alpha = 1$..*

MOS calibration masks, but such masks always contain at least one asymmetric point, misaligned with the rest of the grid. This single point is sufficient to create a great number of unique triangles, making this procedure work.

A possible drawback of this method lies on the exploding number of triangles at the increase of the points in the pattern. The number of possible triangles that can be drawn from a distribution of n points is given by

$$N = \binom{n}{3} = \frac{n(n-1)(n-2)}{6}$$

A VIMOS mask may contain up to 200 slits, meaning more than three million triangles to be handled. Even if elongated and ambiguous triangles are excluded from the analysis, they still need to be computed, and the time complexity of this algorithm remains $O(n^3)$. For this reason a simplified version of this algorithm has been used in the VIMOS pipeline recipes implementation, where not all possible triangles are considered, but just those triangles defined by nearby slits.

This 2D pattern-recognition method is also successfully applied in the correction of the WCS in the FORS1/2 and WFI pipeline, and as a possible recovery method for echelle instrument instabilities in the X-Shooter pipeline.

9.20.7 Optical distortion model determination

The optical distortion model may be (optionally) obtained by fitting a polynomial transformation to the matching points on the mask and on the CCD planes, as found by the 2D pattern-recognition task (see previous Section). The used polyomial model is described in Section 8.3.1, page 156.

Once the optical distortion model is determined, it is applied to the positions of the slits on the mask plane, improving the accuracy of their computed positions on the CCD.

No optical distortion model can really be defined if there are too few spectra on the CCD: in that case, just a local position of the reference wavelength is used for each individual spectrum, and the slits are left unidentified.

Note that slit identification is not essential to the data reduction, and it is hardly a requirement when very few slits are in use.

It should be noted, however, than for multiplexing data it is not possible to switch off slit identification.

9.20.8 Tracing slit spectra edges

The spatial curvature is determined by tracing the slit spectra – typically from flat field and scientific exposures. Flat field spectra are ideal for this operation, because the signal is continuous and with high S/N ratio; on the other hand, it is generally necessary to trace also the scientific spectra, to compensate for possible instrument instabilities. Scientific spectra are generally traceable, because the exposure times are typically long enough to produce a very bright sky spectrum, although problems can occur in the blue part of the spectrum. In case the sky emission is not traceable, then the curvature model derived from the flat field exposures must be used.[46] Currently the pipeline does not support tracing of the scientific spectra.

Tracing spectral edges is not a simple task, because the slit spectra are not always so well detached and isolated from each other, and edges from different spectra may overlap. The only possibility is to try to determine a global trend of the spatial curvature based on the well traceable edges, in order to obtain the curvature also where it cannot be directly measured (see next Section).

9.20.9 Spatial curvature model determination

A local spatial curvature model is derived by fitting a low degree polynomial to the traces of one spectral edge. If enough spectra are available, the local curvature model may be superseded by a global description obtained by modeling the coefficients of the local models of all spectra. The used polyomial model is described in Section 8.3.1, page 156.

9.20.10 Extraction of slit spectra

The extraction of slit spectra consists in reading the spectra following their curvature. The extracted spectra are not wavelength calibrated. This extraction method is only applied to arc lamp or sky spectra before using

[46] Tracing bright point-like object spectra is not a solution, as they are not distorted just by optics, but by atmospheric refraction too.

them for determining the (local) wavelength calibration applying the 1D peak detection and pattern-recognition methods described in Sections 9.20.1 and 9.20.2.

All the spectra are read along the spatial direction (i.e., along the CCD columns), and each column is remapped to a new image where the spatial curvature is eliminated. In other words, the x coordinate of the rectified image is still the x coordinate of the CCD.

9.20.11 Line catalogs and reference spectra

Plots of arc lamp spectra from different grisms and lamps combinations are given in Figures 9.8–9.16, 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 9.1 and 9.2.

Wavelength (Å)	Element	Note
3888.643	He	
3964.729	He	
4026.198	He	
4120.844	He	
4387/929	He	
4471.501	He	
4713.143	He	
4921.931	He	
5015.678	He	
5330.777	He	
5400.562	Ne	
5764.410	Ne	
5852.488	He	
5875.671	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	
6402.248	Ne	
6506.528	Ne	
6532.882	Ne	
6598.953	Ne	
6678.151	He	
6717.043	Ne	
6929.467	Ne	
6965.431	Ar	
7032.413	Ne	
7065.249	He	
7147.042	Ar	
7173.938	Ne	
7245.167	Ne	
7272.936	Ar	
7281.350	He	

Table 9.1: *Arc lamp calibration lines available in the VIMOS spectral range (continued in Table 9.2).*

Wavelength (Å)	Element	Note
7383.980	Ar	
7438.900	Ne	
7488.871	Ne	
7503.869	Ar	
7514.652	Ar	
7535.774	Ne	
7544.044	Ne	
7635.106	Ar	
7723.952	Ar	
7948.176	Ar	
8006.157	Ar	
8014.786	Ar	
8103.693	Ar	
8115.311	Ar	
8264.522	Ar	
8300.326	Ne	
8377.608	Ne	
8408.210	Ar	
8424.648	Ar	
8495.360	Ne	
8521.442	Ar	
8591.259	Ne	
8634.647	Ne	
8654.384	Ne	
8667.944	Ar	
8780.623	Ne	
8853.867	Ne	
9122.967	Ar	

Table 9.2: *(Continued from Table 9.2) Arc lamp calibration lines available in the VIMOS spectral range.*

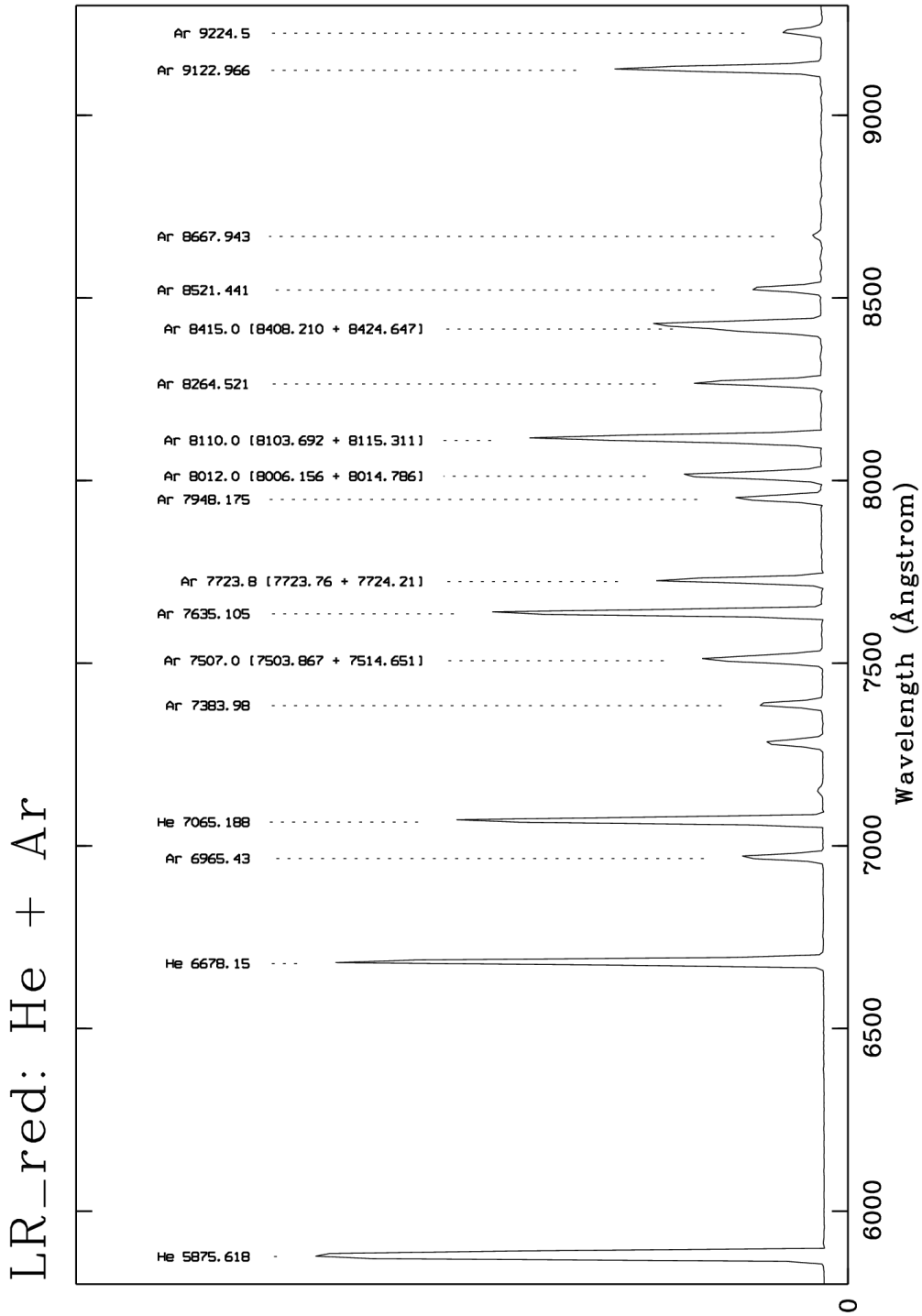

Figure 9.8: *LR red arc line spectrum from 5800 to 9300 Ångstrom. The lines used for calibration are indicated.*

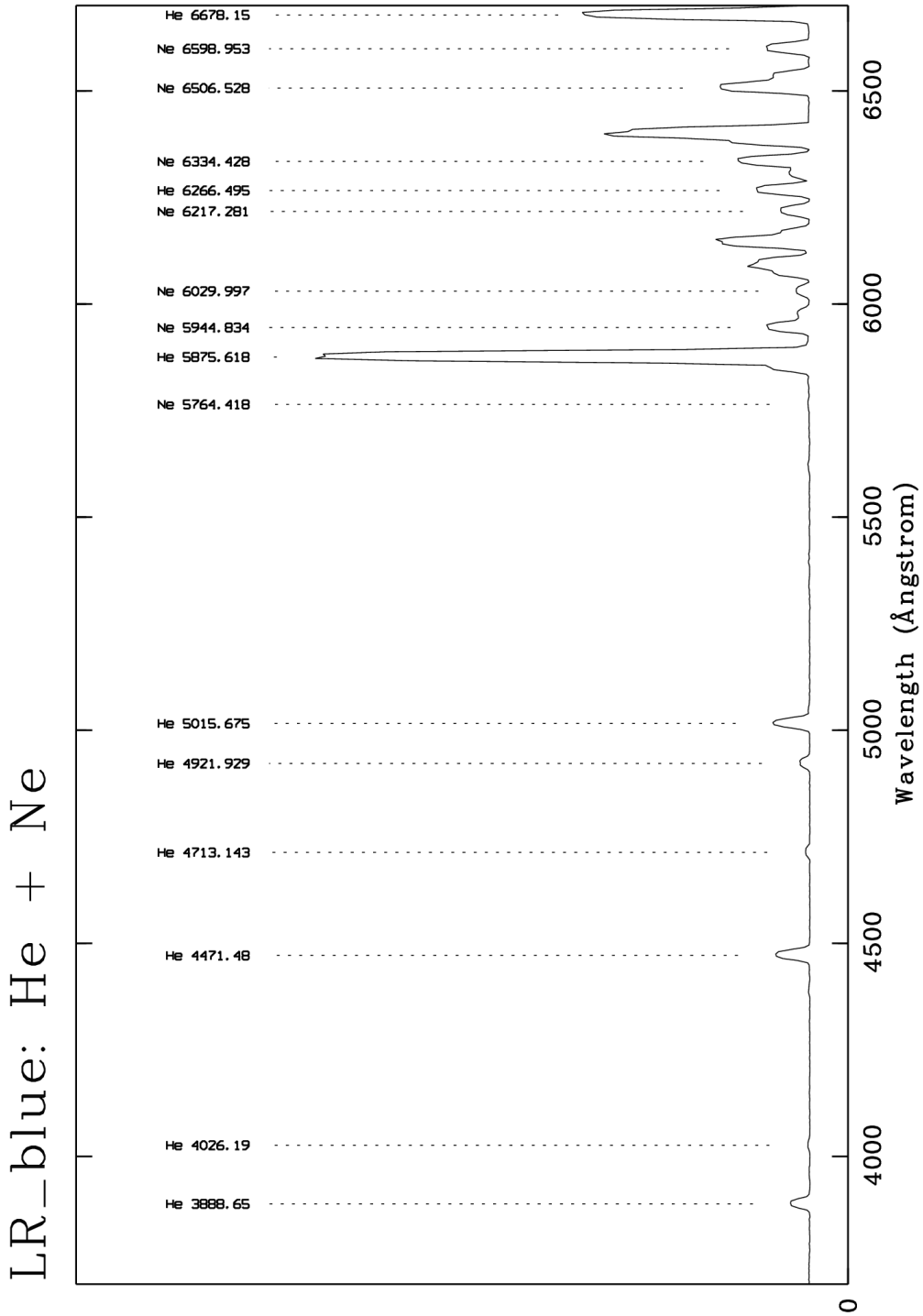

Figure 9.9: *LR blue arc line spectrum from 3800 to 6750 Ångstrom. The lines used for calibration are indicated.*

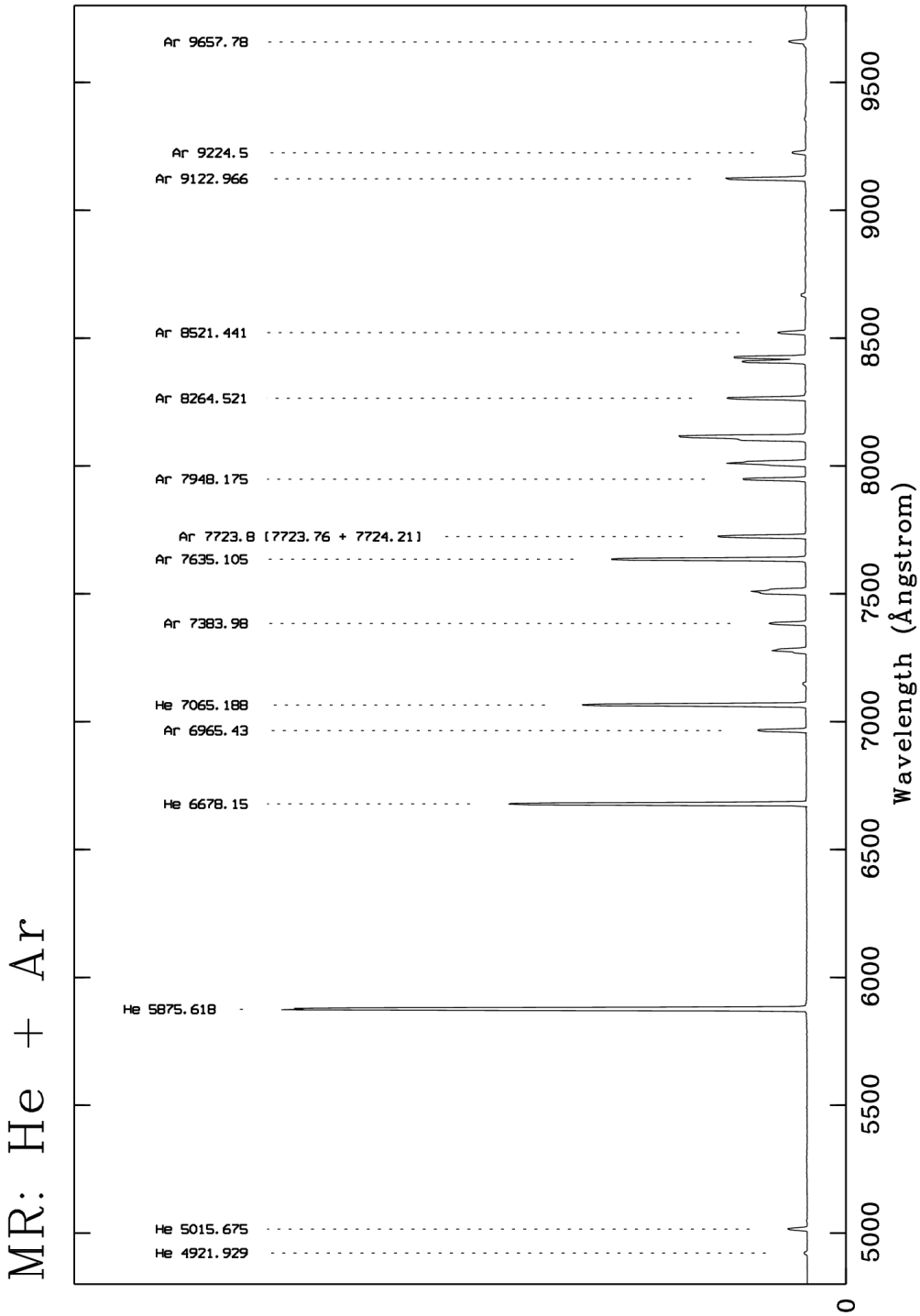

Figure 9.10: *MR arc line spectrum from 4800 to 9800 Ångstrom. The lines used for calibration are indicated.*

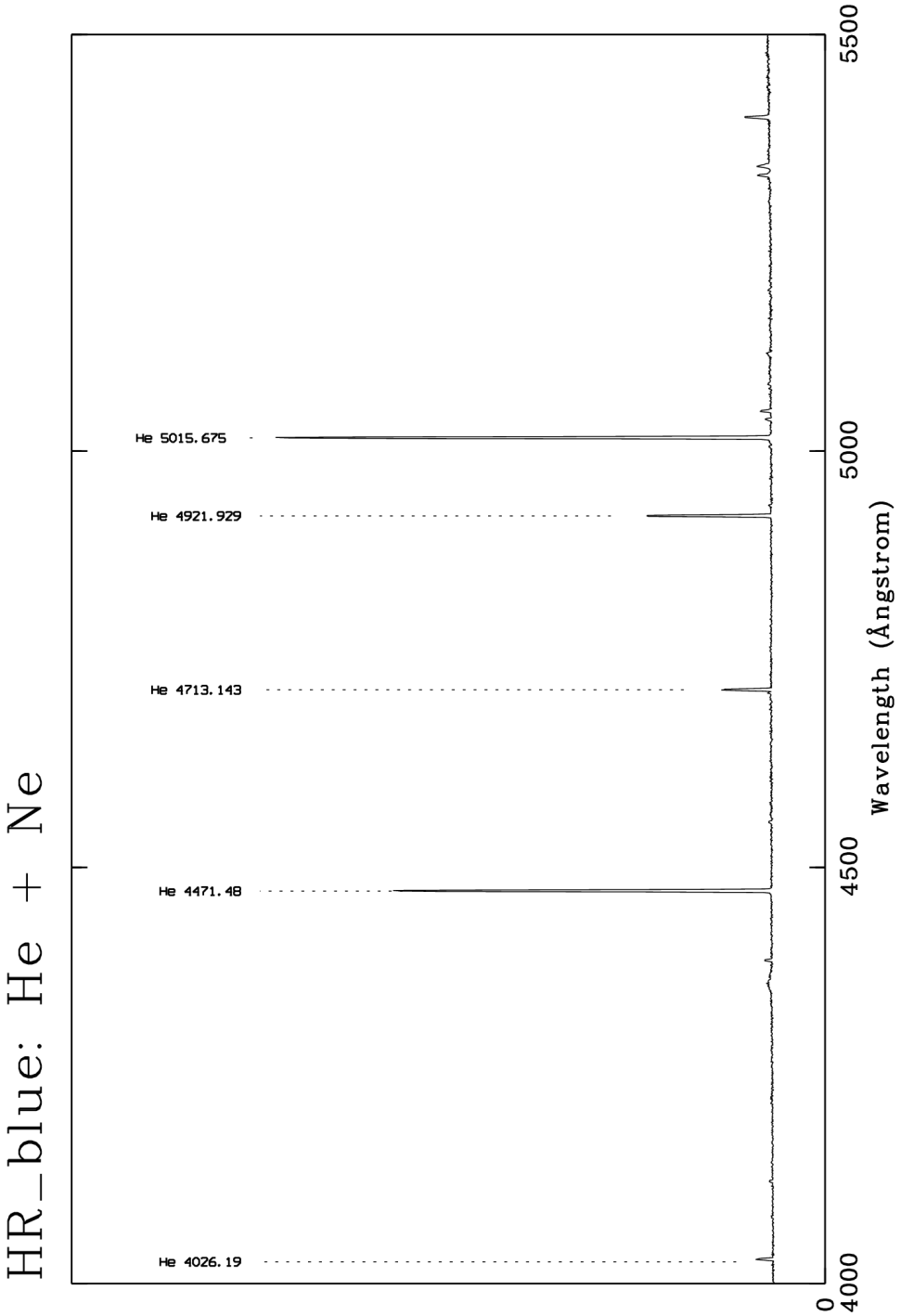

Figure 9.11: *HR blue arc line spectrum from 4000 to 5500 Ångstrom. The lines used for calibration are indicated.*

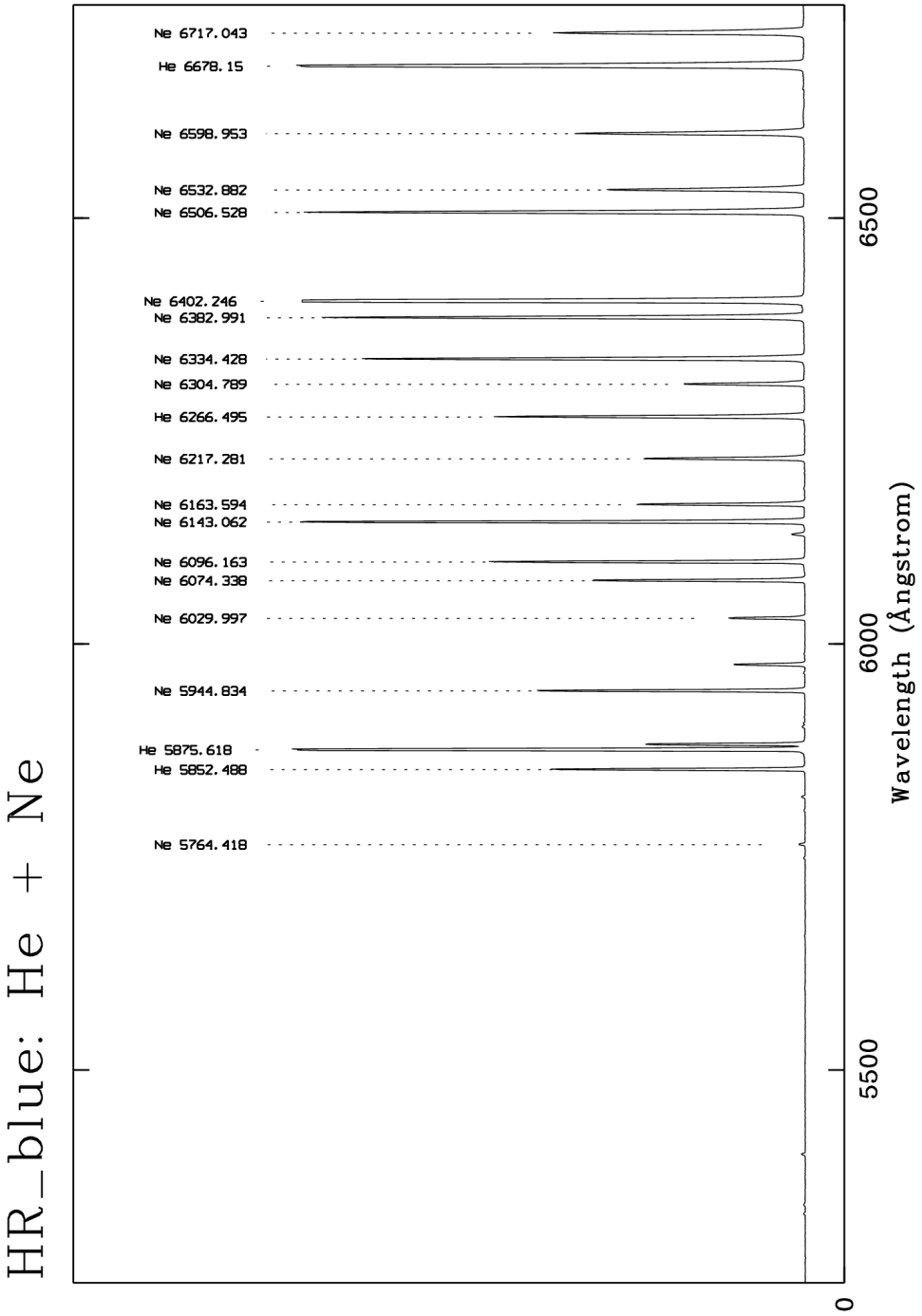

Figure 9.12: *HR blue arc line spectrum from 5250 to 6750 Ångstrom. The lines used for calibration are indicated.*

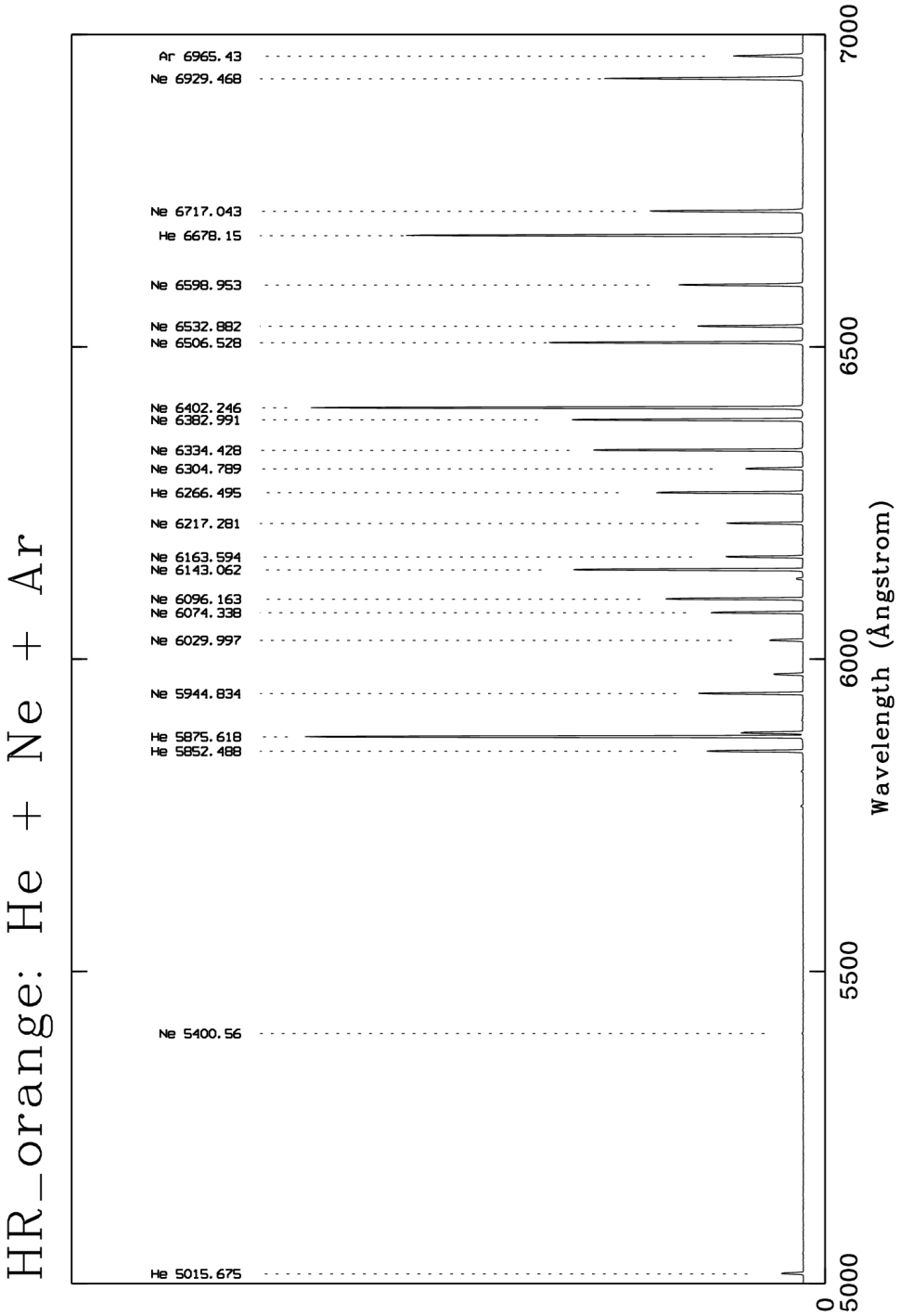

Figure 9.13: *HR orange arc line spectrum from 5000 to 7000 Ångstrom. The lines used for calibration are indicated.*

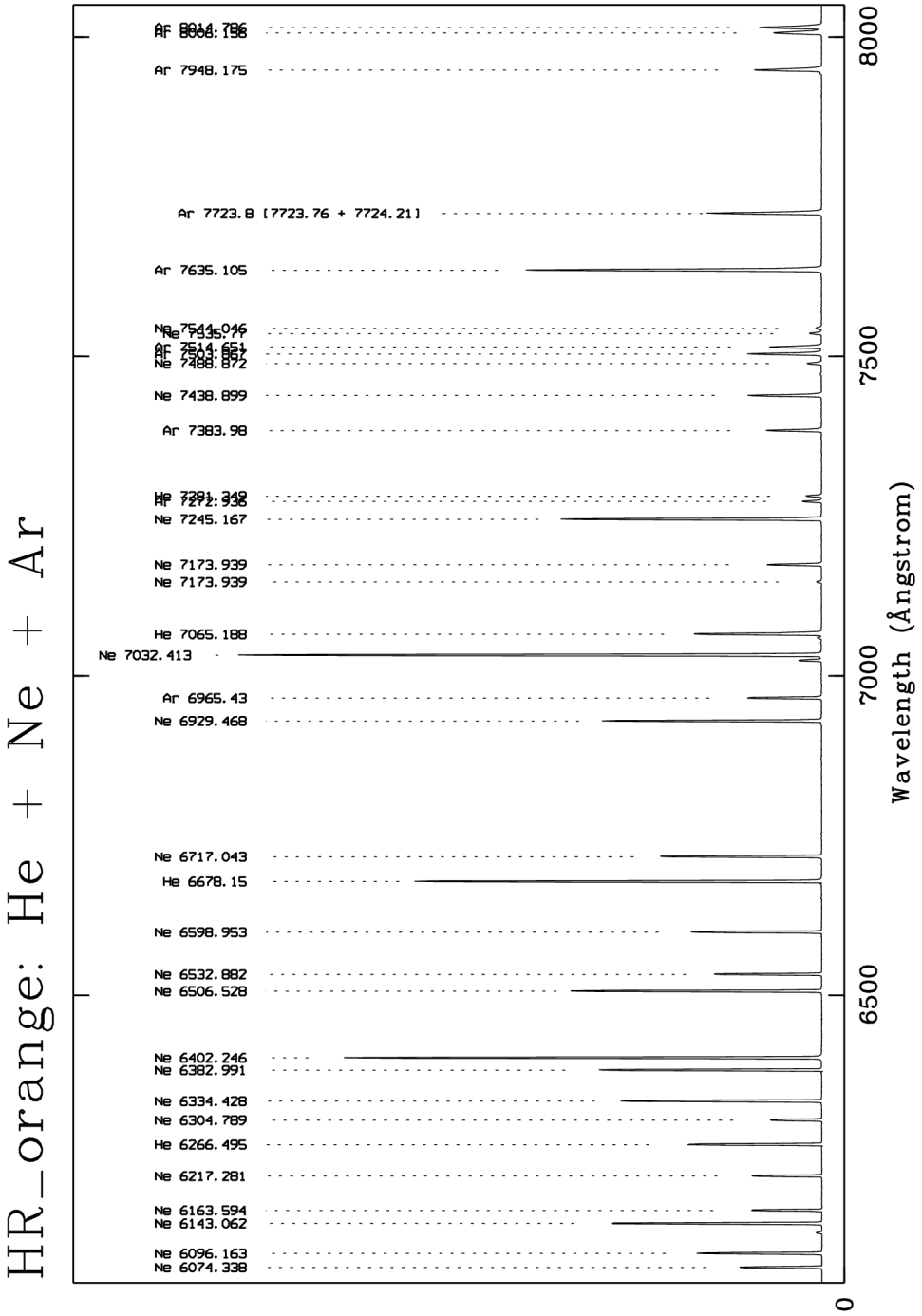

Figure 9.14: *HR orange arc line spectrum from 6050 to 8050 Ångstrom. The lines used for calibration are indicated.*

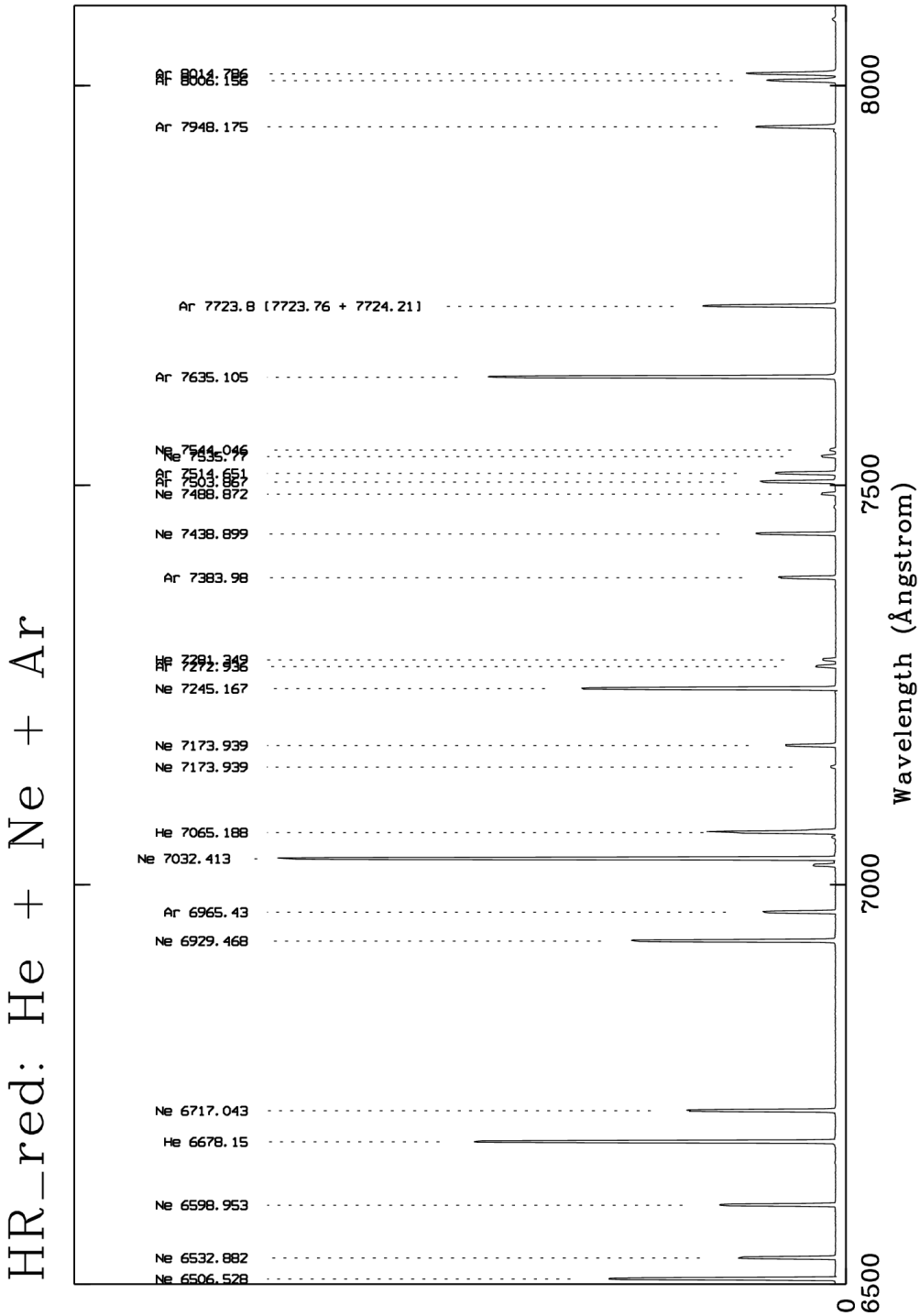

Figure 9.15: *HR red arc line spectrum from 6500 to 8100 Ångstrom. The lines used for calibration are indicated.*

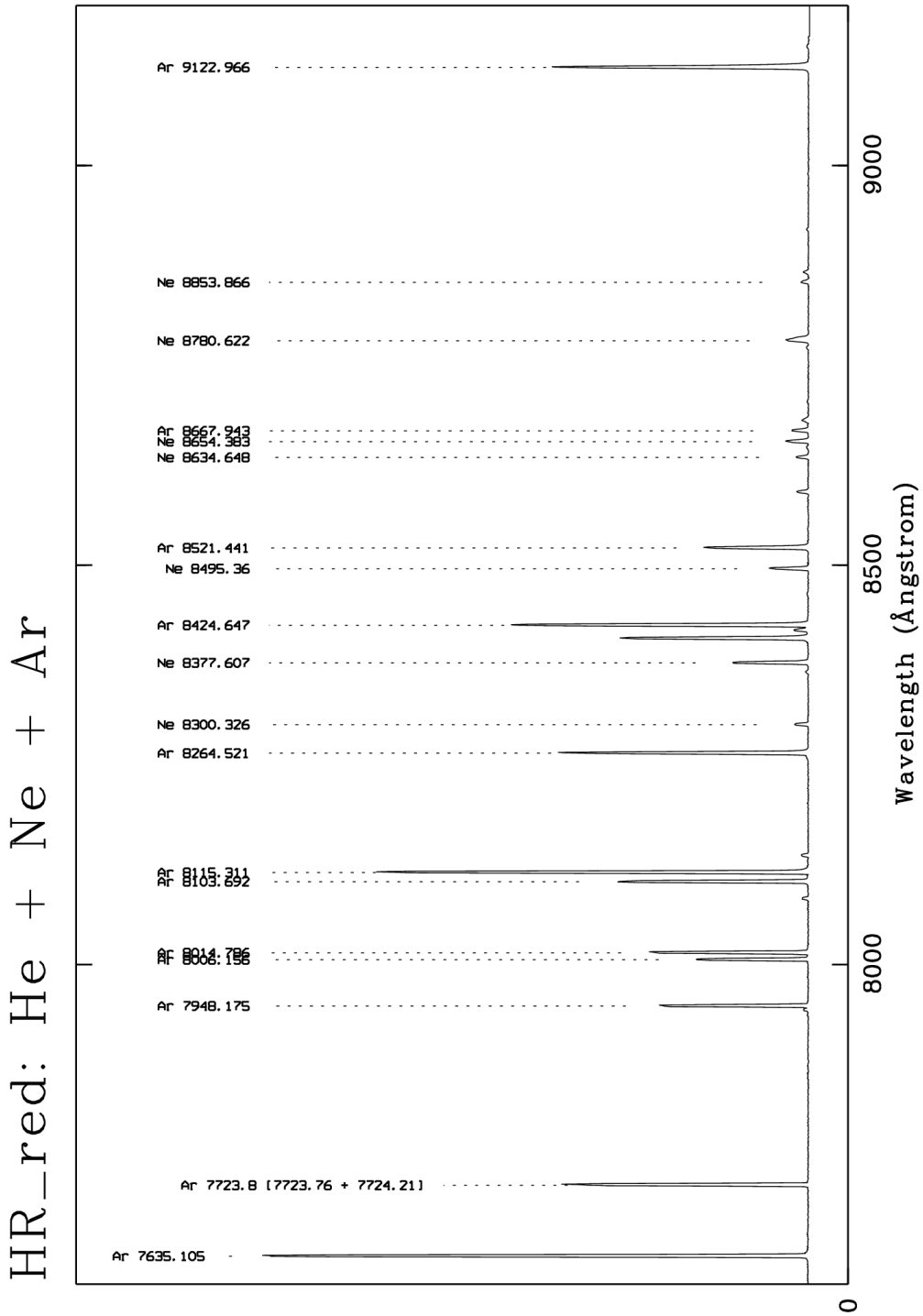

Figure 9.16: *HR red arc line spectrum from 7600 to 9200 Ångstrom. The lines used for calibration are indicated.*

9.21 vmmosscience

This recipe is used to reduce one or more scientific exposures made in MOS mode. The following fundamental steps are carried out when only one exposure is specified in input:

1. Bias subtraction (see Section 9.10).
2. Optional flat field correction.
3. Align wavelength solution to sky lines positions (optional) (see Section 9.21.1).
4. Either global or local sky modeling and subtraction (optional) (see Section 9.21.4).
5. Slit spectra rectification (see Section 9.21.2).
6. Object detection (see Section 9.21.3).
7. Sky modeling and subtraction from extracted slit spectra (optional) (see Section 9.21.4).
8. Object extraction (see Section 9.21.5).
9. In case the input exposure is from a spectrophotometric standard star, the efficiency and response curves can be derived (see Section 9.16).
10. Flux calibration of extracted spectra (see Section 9.16).

If more than one scientific exposure is specified in input, the following reduction steps are applied individually to each input exposure:

1. Bias subtraction (see Section 9.10).
2. Optional flat field correction.
3. Align wavelength solution to sky lines positions (optional) (see Section 9.21.1).
4. Either global or local sky modeling and subtraction (see Section 9.21.4).

The following steps involve all the images containing the extracted and sky subtracted slit spectra:

1. Align and combine the processed images (see Section 9.21.6).
2. Slit spectra rectification (see Section 9.21.2).
3. Iterate the object detection on the combined image (see Section 9.21.3).
4. Object extraction from the combined image (see Section 9.21.5).
5. Flux calibration of extracted spectra (see Section 9.16).

Beyond the standard reduction steps, described in some detail in the indicated sections, only few more steps need to be outlined here.

9.21.1 Align wavelength solution 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 a wavelength solution, derived from the day calibrations, to a possible variation due to mechanical flexures or to a change of temperature. For each slit spectrum, the wavelength solutions derived from the calibration data are used to look for the reference sky lines around their expected positions. A fit to the observed residuals is added to the input first-guess solution.

9.21.2 Slit spectra extraction

Each slit spectrum is read from the pre-processed input image, following the shapes of the modeled spectral distortions (see entries MOS_DISP_COEFF and MOS_CURV_COEFF, page 110). The slit spectra are always resampled along the spatial direction as described on page 108 at the MOS_ARC_SPECTRUM_EXTRACTED entry, and along the dispersion direction at the wavelength step defined by the recipe configuration parameter *--dispersion*, 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.

9.21.3 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 by the recipe configuration parameters (see Section 7.12.3, page 133), excluding cosmic ray events. The profile is analysed by looking at peak features, i. e., at least three pixels for which the central one stands above the adjacent ones. The central pixel must be above the detection threshold limit (provided by *--detection* parameter).

Each detected peak is compared with all the other peaks to determine if this peak is contaminated by any of the others. The parameter *--cont_radius* represents the minimum distance at which two point like objects of equal luminosity can stay without contaminating each other.

9.21.4 Sky modeling

Three different methods of sky modeling are applicable. They are controlled by the three configuration parameters *--skyglobal*, *--skylocal* and *--skymedian* (see Section 7.12.3, page 133),

Both the global and the local sky subtraction methods are applied *before* the slit spectra rectification, and therefore they provide the best results. They are described on page 127, at the GLOBAL_SKY_SPECTRUM entry, and in Section 7.12.3, page 133.

The median sky subtraction method is applied *after* the slit spectra rectification.

9.21.5 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 spatial curvature. The algorithm used is the one described by K.Horne (1986) [14]. 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.

Both the aperture and the optimal extraction are applied to the regions specified in the object table (see page 132, entry OBJECT_SCI_TABLE). About the way such regions are determined, see also the description of the configuration parameters *--slit_margin*, *--ext_radius* and *--cont_radius* in Section 7.12.3, page 133.

9.21.6 Align and combine the processed images

Dithered exposures can be aligned and combined (see Section 9.12) after the variable sky background has been modeled and subtracted from each individual frame. The relative offsets between exposures are determined either using the common detected objects in each frame, or the changing pointing of the telescope read from the header keywords RA and DEC. If the input frames belong to different nights, it may advisable to rely on the common detected objects for the alignment.

Alignment can be performed either to the nearest pixel offsets (reducing correlated noise, i.e., the covariances), or to a fraction of a pixel (requiring signal interpolation).

9.22 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.

9.22.1 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 9.10, page 169).

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.[47]
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.

9.22.2 Fibers identification

Please refer to Section 3.5, page 21, 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 9.17).

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.

[47] As of today (May 2005) this condition is not yet fulfilled, and therefore night calibrations are still mandatory.

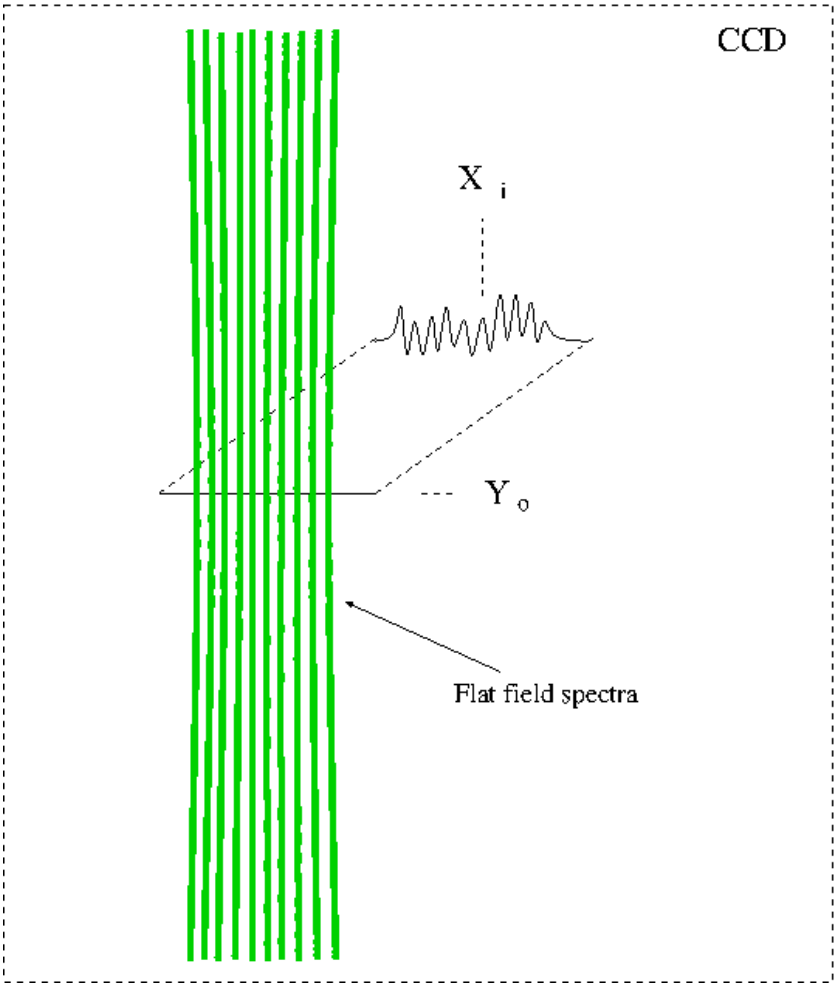

Figure 9.17: *The reference Y_o coordinate in the fiber identification procedure is conventionally fixed.*

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 7.13, page 140).

9.22.3 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 9.22.2, page 204). 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

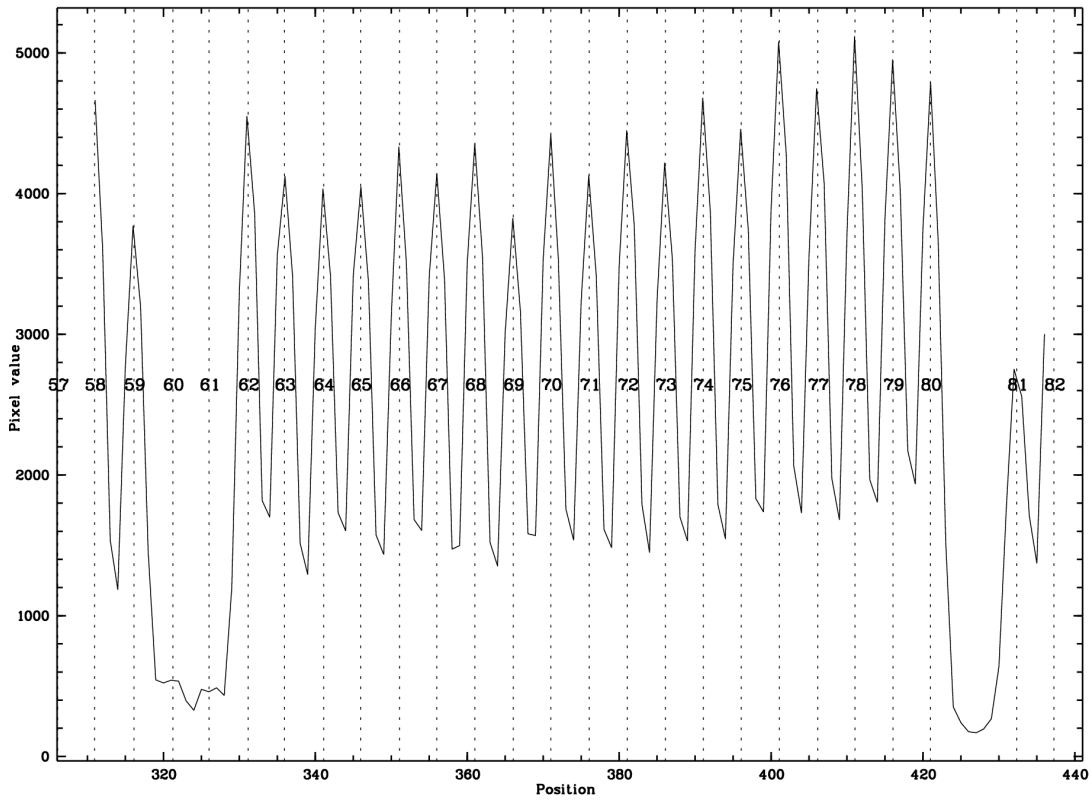

Figure 9.18: *Fiber identification along the cross-dispersion direction. The gap between the first and the second blocks of fibers is visible on the right. On the left the positions of two fibers lost to the IFU head shutter are marked.*

extraction are met.

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 9.19 and 9.20).

The tracing of all the detected flat field spectra, performed by the recipe *vmifucalib* (see Section 7.13, page 140) leads to the definition of the extraction mask that will be used in the extraction of the scientific spectra associated to the flat field.

9.22.4 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. Such extraction is performed following the spatial curvature model, that for each CCD pixel defines a *row* on the slit spectrum (see Figure 8.2, page 157).

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 8.3.5).
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 3. 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.

An alternative way to identify lines is provided, the so-called *blind* method, that doesn't need any first-guess IDS model, being based on pattern recognition. This method is described in some detail in Section 9.13, page 171.

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 3rd order for low dispersion grisms, and 4th order for higher dispersion grisms (see Section 8.3.5). This process is repeated for each row of each slit of the mask.

The global IDS model is determined by the bivariate polynomial fitting of the coefficients of the local polynomials, as described in Section 8.3.5. 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 9.8–9.16, 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 9.1 and 9.2.

9.22.5 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

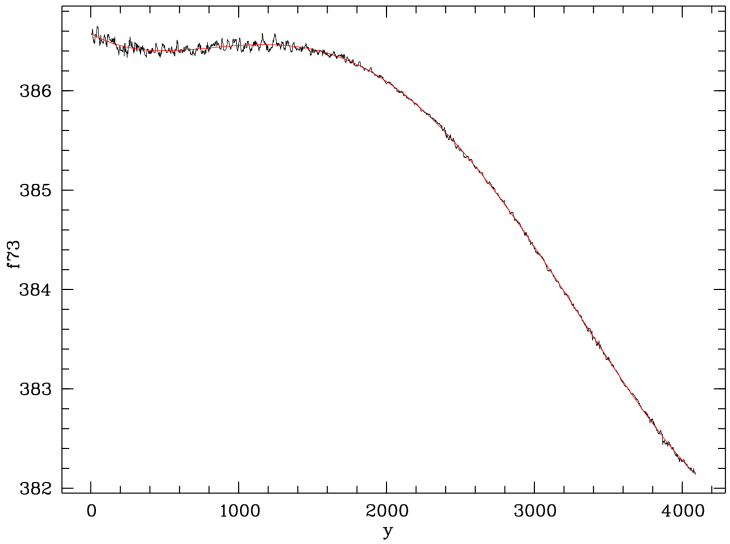

Figure 9.19: *Tracing of fiber spectrum 73 of HR_orange flat field in quadrant 3. Both axis are in pixels.*

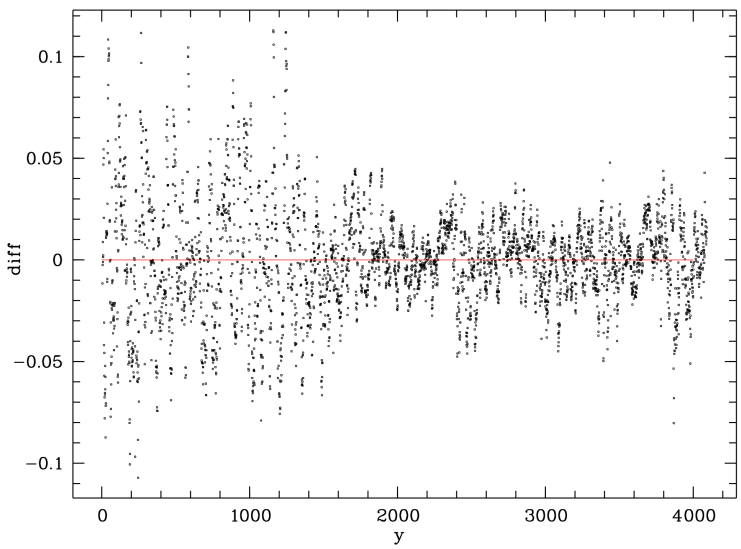

Figure 9.20: *Residuals of the tracing of figure 9.19.*

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. [48]

9.22.6 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 9.21) up to a distance of 5 pixels where the ground level is reached.

Thanks to the spatial 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 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.

The colour dependency of the reconstructed profile is negligible (less than 2% between profiles reconstructed from red and blue regions), but the differences between the ~ 100 studied fibers cannot always be neglected. In the current pipeline release the standard (numerical) model of the fiber spatial profile is adapted to the brightest (and uncontaminated) parts of the observed profile of each fiber. The fit is performed simultaneously in all wavelengths, in order not to introduce extra noise. This expedient significantly improves the spectral extraction, compensating for the differences between fibers, and managing the proper reduction of observations obtained in less than optimal conditions (*e.g.*, out of focus).

Examples of reconstructed fiber profiles are given in Figures 9.22, 9.23 and 9.24. 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

[48] This is not implemented in the current release.

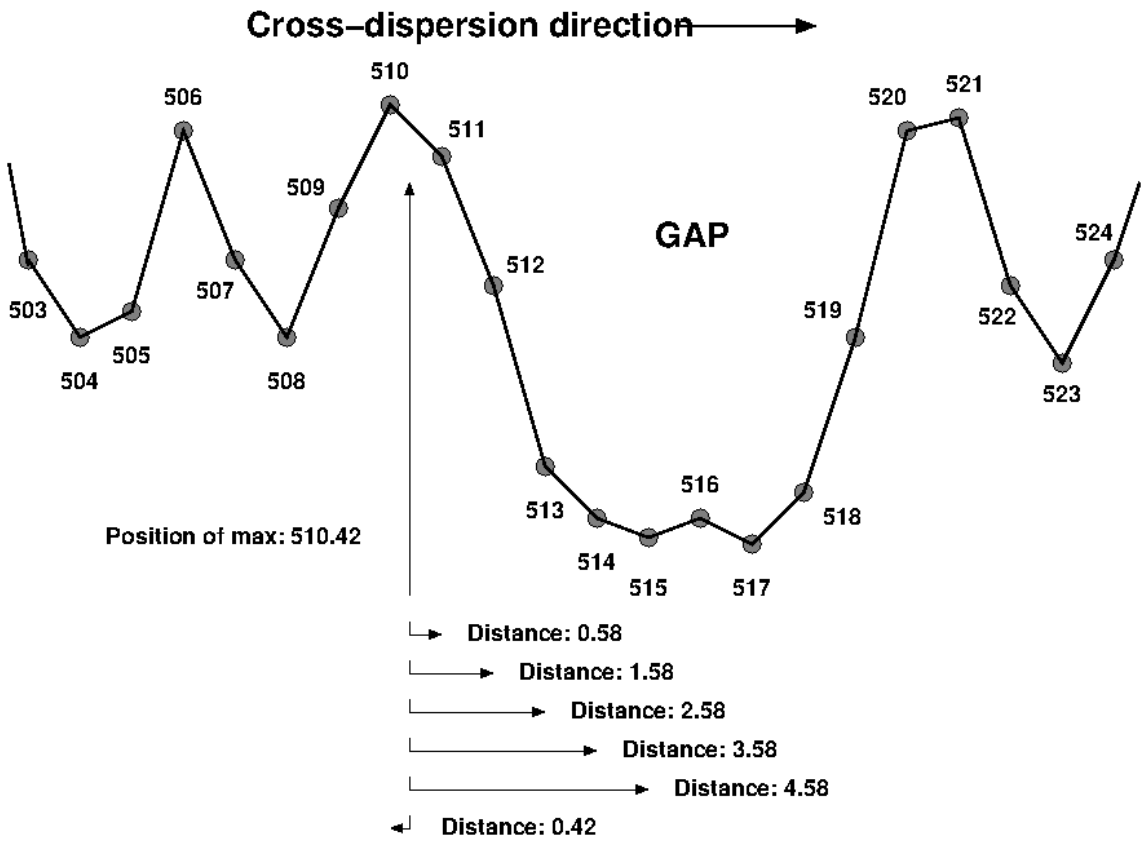

Figure 9.21: *Reading the fiber profile. In this example, some fiber profiles around a gap between two fibers blocks are shown. The last fiber profile at the left of the gap peaks at the pixel position 510.42 (obtained from the tracing solution). The pixels distances from the peak position are assigned 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).*

slightest error in the tracing solution (> 0.05 pixel) would introduce major discontinuities in plots like the ones in Figures 9.22 and 9.23.

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 9.24 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

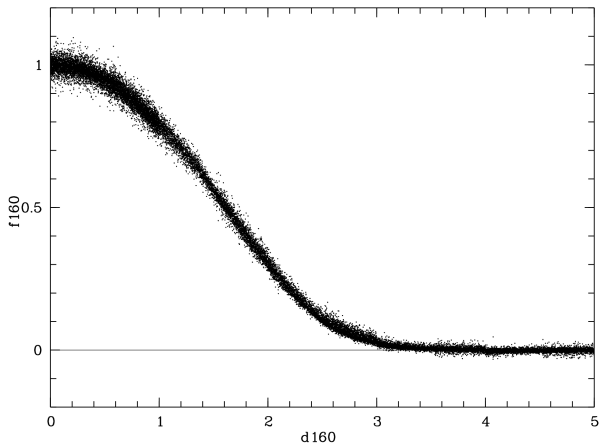

Figure 9.22: *Observed cross-dispersion profiles of spectra from fiber 160 on the central slit of quadrant 3. Orange grism.*

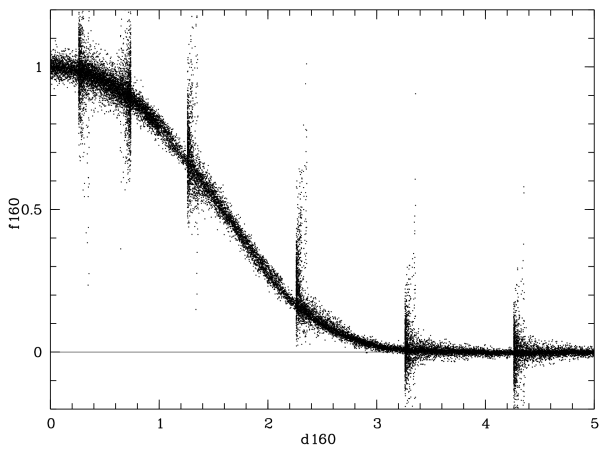

Figure 9.23: *Observed cross-dispersion profiles of spectra from fiber 160 on the central slit of quadrant 3. Blue grism.*

convolution of a gaussian with a box profile. In the current system, the same tabulated model profile is always used in the spectral extraction task (see Table 9.3). This profile was chosen as the median profile of a sample obtained from about 100 different fibers from the pseudo-slits of all quadrants.

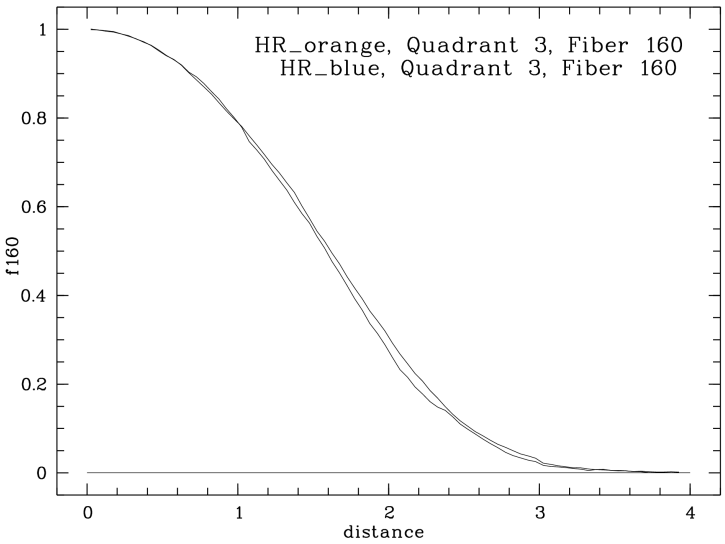

Figure 9.24: *The smoothed profiles obtained from Figures 9.22 and 9.23 are compared. The fiber profile from the HR_blue spectrum lays systematically below the fiber profile from the HR_orange spectrum, but the difference is* $< 2\%$.

9.22.7 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 3.4, page 22), giving to the objects on the reconstructed field-of-view a typical zig-zagged appearance.

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 9.3: *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).*

9.22.8 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 [14] 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 9.3, page 214), and then optimally 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).

9.22.9 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. 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.

9.22.10 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.[49]

In the current system, no flat fielding correction is applied to the scientific data. The recipe *vmifucalib* (see Section 7.13, page 140) 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).

9.22.11 Transmission correction

The spectral extraction procedure (see Section 9.22.8, page 215) 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.

9.22.12 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 7.14, page 145).

9.23 IFU WCS information

The VIMOS pipeline produces 3D IFU cubes that contain basic WCS information, including physical sky coordinates and wavelength coordinates in the third axis. Non-linear distorsions are not considered. Since the VIMOS IFU field of view is is offset with respect to the VIMOS optical axis, the pointing coordinates have to be transformed to account for that. The offsets are as follow:

$$off_x = 667.8 AlongX(WestforPA = 0, \delta = 0) off_y = -1.7 Along - Y(SouthforPA = 0, \delta = 0) \quad (19)$$

The formulae used to get the IFU coordinate center are:

[49] 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.

$$RA_ifu = RA_VIMOS - \arcsin(-cos(ofs) * \sin(ppa)/\cos(\delta_ifu)); \delta_ifu = \arcsin(cos(ppa) * \cos(ofs) * \cos(\delta_VIMOS) + \tag{20}$$

where RA_VIMOS, δ_VIMOS are the coordinates pointed by VIMOS optical axis and ppa, ofs are defined as follows:

$$ppa = -90 - \arctan((off_y/off_x) + PAofs = (90. - \sqrt{off_x * off_x + off_y * off_y}/3600.) \tag{21}$$

9.24 vmifucalib

In this section the basic steps of the data reduction procedure applied by the recipe *vmifucalib* (see Section 7.13, page 140) are described. Please refer to Section 9.22, page 203, 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 9.22.3, page 205. 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 9.22.8, page 215).
5. The wavelength calibration is obtained for each extracted arc lamp spectra (see Section 9.22.9, page 216). 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 9.22.11, page 217).

9.25 vmifuscience

In this section the basic steps of the data reduction procedure applied by the recipe *vmifuscience* (see Section 7.14, page 145, are described. Please refer to Section 9.22, page 203, 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 9.22.3, page 205).
5. The extraction mask obtained from the flat field is aligned to the available tracing solutions on the science (see Section 9.22.7, page 213).
6. The science spectra are extracted along the (possibly modified) flat field tracings (see Section 9.22.8, page 215).
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 7.16, page 149).
11. Flux calibration of extracted spectra (see Section 9.16).

9.26 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 9.22.12, page 217), and the determination of the total standard star spectrum from all the single fiber spectra. Finally, the efficiency and response curves are produced as described in section 9.15.

A Troubleshooting Guide

In the following sections, a troubleshooting guideline for the VIMOS pipeline spectroscopic recipes is given. It is assumed here that some familiarity with these recipes was already acquired.

See also Sections 7.11.3 and 7.12.3 of this Manual.

A.1 Checking the results of recipe vmmoscalib

The pipeline generally cannot handle wavelength calibration data for slits wider than 1.5″. Things can go wrong. In this Section a number of basic checks are suggested for ensuring that the *vmmoscalib* recipe worked properly. Troubleshooting is given separately, in the next Section, in order to avoid too many textual repetitions: it often happens, in fact, that different problems have the same solution. Three basic checks are described here: spectra localisation, wavelength calibration, and spectral resolution. It is advisable to perform such checks in the given order, because some results make only sense under the assumption that some previous tasks were performed appropriately. For instance, an apparently good wavelength calibration does not imply that the slit spectra were all properly traced.

A.1.1 Were all spectra detected and properly traced?

Compare (blink) the *mos_master_screen_flat.fits* and the *mos_combined_screen_flat.fits* images. The normalised flat field image can be used as a map showing where the spectra were found and how they were cut out from the CCD, while the master flat image shows where the spectra actually are. A quick visual inspection will immediately expose any badly traced, or even lost, spectrum. This kind of failure may not be so apparent in the *mos_arc_spectrum_extracted.fits* image, which includes just what has been successfully extracted.

The *mos_curv_traces.fits* table enables a closer look at the tracing accuracy. The tracings of the left and right edges of the spectrum from slit 10, for instance, are given in the table columns labeled "t10" and "b10", for each CCD pixel along the vertical direction given in column "x". Each tracing may be compared with the fitted model: for instance, the modeling of the tracing "t10" is given in the table column "t10_mod", together with the fit residuals in column "t10_res", enabling the generation of plots like those shown in Figures A.1 and A.2. In order to reduce the residuals, the degree of the fitting polynomial may be increased (using the configuration parameter "cdegree"): it is however advisable to never use polynomials above the 2nd order for low-resolution data, and above the 4th order for high-resolution data, unless the residuals are really not acceptable. In Figure A.2 the residuals are less than 3 hundreds of a pixel, and this is acceptable even if they display a systematic trend that may be easily eliminated by fitting a 3rd degree polynomial. When systematic trends in the residuals are so small (with respect to the pixel size), they can no longer be considered "physical", but rather an effect of the pixelisation of the edge changing with the position along the CCD. See also Figures 7.6, 7.7, pages 120, 121 of this Manual, and the related Section.

A.1.2 Were all spectra properly calibrated in wavelength?

Check the *mos_arc_spectrum_extracted.fits* image first. This image contains the arc lamp spectra from each slit with all the optical and spectral distortions removed. The spectral lines should all appear perfectly aligned

and vertical.

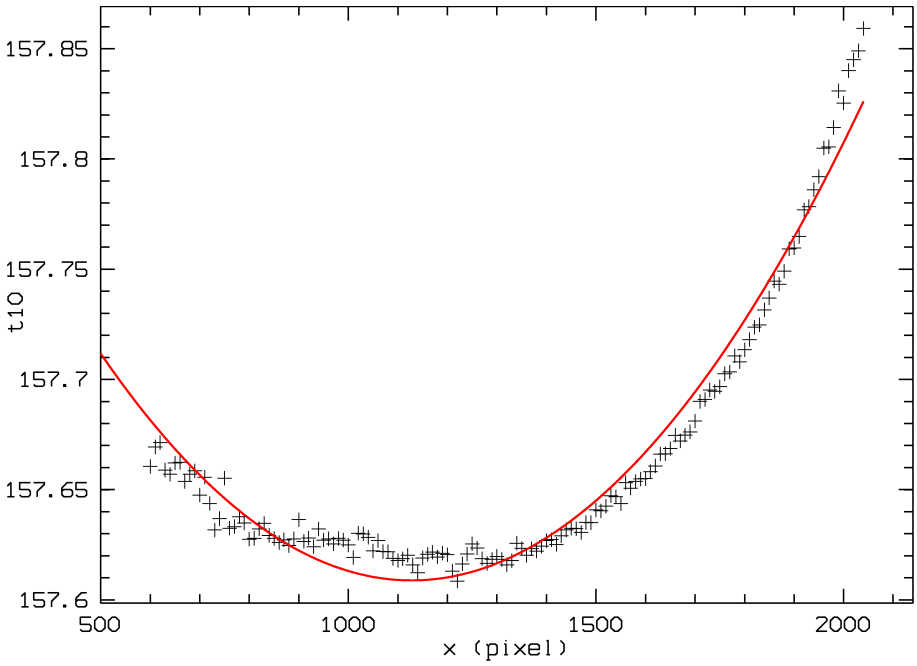

Figure A.1: *Tracing and modeling of one spectral edge tracing.*

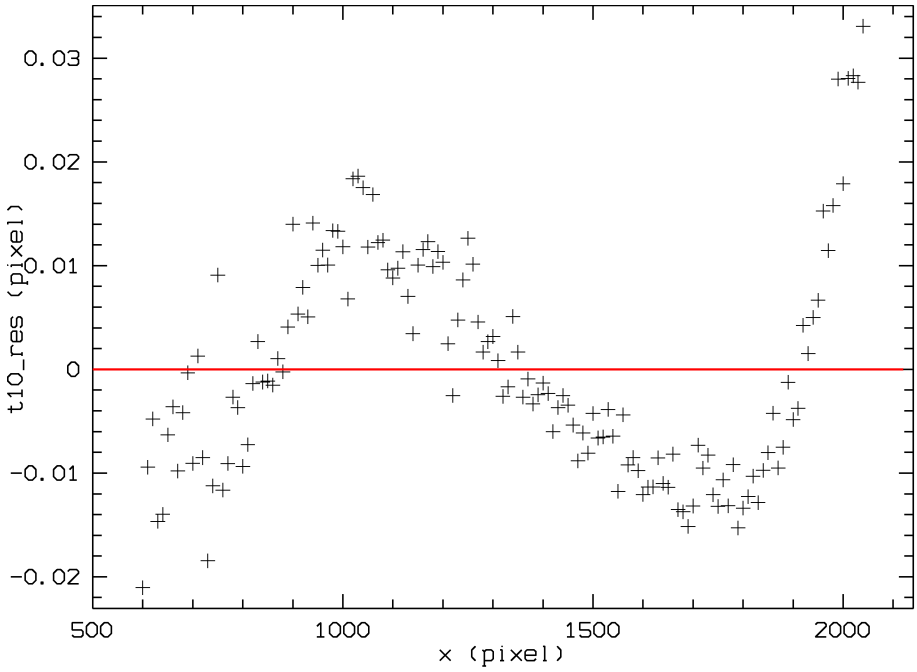

Figure A.2: *Ssystematic residuals (in pixel) of spectral edge tracing of figure A.1.*

Particular attention should be given to lines at the blue and red ends of each spectrum, where the polynomial fit is more sensitive to small variations of the signal. The calibrated slit spectra are vertically ordered as in the original CCD frame. The boundaries between individual slit spectra are generally easy to recognise: both because they are often dotted by the emission lines from nearby spectra on the original CCD frame, and because each slit spectrum may cover different wavelength intervals according to its position within the original CCD frame (see Figure 7.1, page 108). The position of each spectrum in the calibrated image is always reported in the table *mos_slit_location.fits*, at the columns "position" and "length".

More detailed checks on the quality of the solution can be made by examining other pipeline products. The image *mos_disp_residuals.fits* contains the residuals of the wavelength solution for each row of each extracted slit spectrum. This image is mostly padded with zeroes, with the only exception of the pixels where a reference line was detected and identified: those pixels report the value of the corresponding residual (in pixel). This image will in general be viewed applying small cuts (typically between -0.2 and 0.2 pixels): systematic trends in the residuals, along the dispersion direction, would appear as sequences of all-positive (white) followed by all-negative (black) residuals, in a wavy fashion, that could also be viewed by simply plotting a profile at different image rows (see Figure 7.3, page 113). Systematic residuals in the wavelength calibration are in general not acceptable, and they may be eliminated by increasing the order of the fitting polynomial.

Another product that can be used for evaluating the quality of the fit is the *vimos_disp_residuals_table.fits* file. Here the residuals are reported in a tabulated form for each wavelength in the reference lines catalog, but just for one out of 10 rectified image rows (i.e., one out of 10 solutions). In conjunction with the *mos_delta_image.fits* image, plots like the ones in Figure 7.2, page 111, can be produced.

Finally, the table *mos_disp_coeff.fits* might be examined to check how many arc lamp lines were used (column "nlines") and what is the mean uncertainty of the fitted wavelength calibration solution (column "error"), for each row of each slit spectrum. The model mean uncertainty is given at a 1-σ level, and has a statistical meaning only if the fit residuals do not display any systematic trend and have a random (gaussian) distribution around zero. Typically this uncertainty will be of the order of 0.05 pixels, i.e., much smaller than the root-mean-squared residual of the fit, depending on the number of fitted points (a fit based on a large number of points is more accurate than a fit based on few points). It should be anyway kept in mind that the model uncertainty can be much larger than that (up to 1 pixel in the worst cases) at the blue and red ends of the fitted wavelength interval). This is because in the pipeline the wavelength solution is obtained by fitting a polynomial, rather than a physical model of the instrument behaviour.

If the parameter "wmodemos" is set to 1 (the default), the wavelength calibration can be much more accurate than that, even at the extremes of the spectral range. The errors reported in *mos_disp_coeff.fits* always refer to the single calibrations (each CCD column is calibrated independently), but if "wmodemos" is set to 2 a global model is fitted to all the reference lines visible on the whole CCD, which may lead to a calibration accuracy of the order of 0.001 pixels (at least theoretically: systematic errors, e.g., due to physical irregularities of the slits, are not included in this estimate).

A.1.3 Is the spectral resolution as expected?

The table *mos_spectra_resolution.fits* reports on the mean spectral resolution, defined as $R = \lambda/\Delta\lambda$ (with $\Delta\lambda$ determined at half-maximum), which was measured for each reference arc lamp line (see Figure 7.5, page 116). The standard deviation from this mean is also given, together with the number of independent determinations of R in column "nlines".

A.2 Fixing pattern-recognition failures in vmmoscalib

In this Section and the following a set of possible solutions to almost any problem met with the *vmmoscalib* recipe is given. It is advisable to try them in the same order as they are listed here. It may be useful to go through this check list even in case the recipe seemed to work well: there might always be room for improvement.

In practice, almost any problem with the pipeline is caused by a failure of the pattern-recognition task. Pattern-recognition is applied to detect the slit spectra on the CCD, assuming that they all will include an illumination pattern similar to the pattern of wavelengths listed in the reference arc lamp line catalog.

For an immediate visualisation of how successful was the pattern-recognition just rerun the *vmmoscalib* recipe setting the "check" parameter to *true*. This will produce a number of extra (intermediate) products. One of them is the *mos_spectra_detection.fits* image, a by-product of the pattern-recognition task, displaying a preliminary wavelength calibration of the CCD. This image has as many rows as the CCD has columns: if at any CCD column the line catalog pattern is detected, the spectral signal is wavelength calibrated, resampled at a constant wavelength step, and written to the equivalent row of the *mos_spectra_detection.fits* image. If a row of this image is empty, it is either because the corresponding CCD column doesn't contain any spectrum, or because the pattern-recognition task failed for that column. The check image may simply be rotated and placed side by side with the original CCD exposure, in order to see if and how frequently a spectral signal was not recognised as such. A few failures (i.e., a few empty columns) are generally acceptable, as they are recovered by interpolation during the final wavelength calibration task. However, a high failure rate is probably the reason why a bad spectral localisation, or tracing, or final wavelength calibration, were possibly obtained.

What can make the pattern-recognition task fail? One or more of the following causes may be determined:

A.2.1 Some arc lamp reference lines are missing

It is possible that the searched pattern is simply not present in the data: for instance, the Neon lamp was off, so only Argon + Helium lines are present.

Solution: Change line catalog accordingly.

A.2.2 Some arc lamp reference lines are very faint

It is possible that the exposure time for the arc lamp frame is too short, or one of the lamps got too faint with age. If some of the reference lines listed in the catalog do not peak above a given threshold, they are not used by the pattern-matching task.

Solution: Specify a lower value for the "peakdetection" parameter. Alternatively, if this gets too close to the noise level, remove the faint lines from the reference line catalog.

A.2.3 The reference lines in the arc lamp exposure are very broad

If very wide slits are used, the reference lines would become accordingly wider (and would display a box-like, flat-top profile). The calibration recipe can handle this in case of well isolated lines, but if nearby lines blend together it is impossible to safely determine their positions.

Solution: None. These spectra cannot be calibrated.

A.2.4 The spectral dispersion is not what expected

The actual mean spectral dispersion is significantly higher (or lower) than expected. The first-guess spectral dispersion is specified via the parameter "dispersion", and is tabulated for each grism in the VIMOS User Manual (see also Table 3.1, page 18), or in the configuration tables which are included in the distributed VIMOS pipeline package. In general the pattern-recognition algorithm is quite robust against changes of the spectral dispersion (up to 20% from expectation), but for some grisms (such as the LR_blue one) good results can only be obtained within a much narrower window of values of the first-guess. For this reason a small change of the spectral dispersion (perhaps caused by a large temperature variation) may cause the wavelength calibration to fail.

Solution: Try different values of the "dispersion" parameter around the expected (default) value, and select the one producing the lowest failure rate of the pattern-recognition task.

A.2.5 There are spectra at very large offsets

The CCD may include spectra at such large y offsets that only part (red or blue) of their full wavelength range is really included in the CCD. If the line catalog contains too few reference lines in this region (say, less than 5), they might not be enough to define an unambiguous pattern to detect.

Solution: Add extra reference lines to the line catalog, for a more complete coverage of the bluest/reddest parts of the complete spectral range. If there are no extra lines to be used as a reference, the truncated spectra will then be definitely lost.

A.3 Fixing other possible failures in vmmoscalib

If the pattern-recognition seems to have worked properly, the reason of a *vmmoscalib* recipe failure can be found elsewhere:

A.3.1 The spectra are too tightly packed

If slits are too close to each others, there is a risk that (some of) the spectra would not be properly traced, or not traced at all, on the flat field frames. As a default, the *vmmoscalib* recipe tries to recover untraceable edges by interpolating a global curvature model based on other traceable edges (if they are available). Using this global description of the spectral curvature helps to extract also those spectra whose edges cannot be traced. In some cases however the recipe may find and accept a bad tracing as if it were good, producing a bad global curvature model, and therefore a bad spectral extraction.

Solution: Setting the parameter "cmode" to zero will suppress the usage of the global curvature model. In this case the recovery strategy of lost spectral edges will consist in replicating the trace of the other available spectral edge (opportunely shifted) of the same slit spectrum. This may improve the results in some cases: however, if a tracing is missing for both edges of a slit spectrum, the spectrum will not be extracted.

A.3.2 The wavelength calibration residuals display systematic trends

Especially if the extracted spectral range is very large, the fitting polynomial may be incapable to replicate the physical relation between pixel and wavelength. In this case, any estimate of the statistical error (such as the fit uncertainties listed in *mos_disp_coeff.fits*) will become meaningless.

Solution: Increase the degree of the fitting polynomial, using the parameter "wdegree". Beware that this may introduce overfitting, especially at the red and blue ends of the spectra (i.e., the polynomial is so poorly constrained in those regions where few points are available, that it also fits their position uncertainty, incorporating this noise into the solution: the corresponding residuals may therefore look very small, and yet the calibrated spectra will appear to be badly calibrated; an extreme case of overfitting is, for instance, fitting 4 points with a 3rd degree polynomial: the residuals will be exactly zero, and yet the obtained model will be highly inaccurate). For this reason, while applying this solution it may be also appropriate to set the parameter "wmodemos" to 2.

A.3.3 The calibrated spectra look "noisy" at their ends

This problem is symmetric to the previous one: the fit residuals may look very small, and yet the calibrated spectra will appear to be badly calibrated at their blue and red ends. This is the effect of model overfitting.

Solution: Decrease the degree of the fitting polynomial, using the parameter "wdegree". Beware that this may introduce systematic fit residuals.

A.3.4 The flat field is not properly normalised

The master flat field is normalised by dividing it by a smoothed version of itself. For various reasons the result may be judged unsatisfactory.

Solution: Change the smoothing box sizes using the parameters "dradius" and "sradius". Alternatively, instead of the default median smoothing, a polynomial may be used to fit the large scale trend: the degree of the fitting polynomial should be specified via the "ddegree" parameter.

A.3.5 Valid reference lines are rejected

Sometimes the peak detection algorithm may return inaccurate positions of the detected reference arc lamp lines. Outliers are automatically rejected by the fitting algorithm, but if those lines were properly identified, not rejecting their positions may really improve the overall accuracy of the wavelength calibration.

Solution: Increase the value of the "wreject" parameter. Extreme care should be used here: a tolerant line identification may provide an apparently good fit, but if this is based on misidentified lines the calibration would include unknown systematic errors.

A.4 Checking the results of recipes vmmosscience

In this Section a number of basic checks are suggested for ensuring that the recipe *vmmosscience* worked properly. Troubleshooting is given separately, in the next Section, in order to avoid too many textual repetitions:

it often happens, in fact, that different problems have the same solution. Four basic checks are described here: wavelength calibration, sky subtraction, object detection, and object extraction. It is advisable to perform such checks in the given order, because some results make only sense under the assumption that some previous tasks were performed appropriately. For instance, an apparently good sky subtraction does not imply that the slit spectra were all properly wavelength calibrated.

A.4.1 Were all spectra properly wavelength calibrated?

The wavelength calibration based on calibration lamps, performed at day-time, may not be appropriate for an accurate calibration of the scientific spectra: systematic differences due to instrumental effects, such as flexures, or change in temperature, may intervene in the meantime.

To overcome this, the day calibration may be upgraded by testing it against the observed positions of the sky lines in the scientific slit spectra. The alignment of the input distortion models to the true sky lines positions is controlled by the parameter "skyalign", that as a default is set to -1 (i.e., the sky lines correction will be disabled).

It is possible, naturally, that an alignment of the distortion models is unnecessary: if this were the case, it would be better to avoid it entirely (any extra manipulation increases the statistical uncertainties on the final product). In order to decide whether a sky alignment is necessary or not, the *mos_sci_skylines_offsets_slit.fits* table can be examined. This table has a column labeled "wave", listing the wavelengths of all the reference sky lines found within the extracted spectral interval, and a number of columns labeled "offset_*id*", listing the median offset in pixels for each sky line from its expected position, for the slit identified by "id" (see Figure A.3). *Beware*: the listed offsets are *not* the residuals of the final sky line alignment, but really the comparison of the sky line positions against expectations from the input distortion models. In case the sky line offsets are compatible with zero, the sky line alignment is really unnecessary, and the *vmmosscience* recipe may be run again setting the "skyalign" parameter to -1 (i.e., the sky lines correction will be disabled). This is not strictly necessary, but it is often wise to keep data manipulation to a minimum. On the other hand, observing systematic offsets would confirm that an alignment of the distortion model to the true sky lines positions was in order, and there would be no need to reprocess the data. In case the offset appears to depend on the wavelength, and there are at least 4 or 5 well separated sky lines available, it may be appropriate to set the parameter "skyalign" to 1.

The overall quality of the wavelength calibration (whether a sky line alignment was applied or not) can be examined in the *mos_science_sky_extracted.fits* image. This image contains the scientific spectra from each slit after removing the optical and spectral distortions. The visible sky lines should all appear perfectly aligned and vertical. The position of each spectrum in the calibrated image is listed in the table *object_sci_table.fits*, at the columns "position" and "length".

A further check on the quality of the solution can be made by examining the *mos_sci_disp_coeff_sky.fits* table. This table is only produced in case a sky line alignment was performed. Column "nlines" reports how many sky lines were used for the distortion model correction, while the "error" column reports the mean uncertainty of the new wavelength calibration solution for each slit spectrum row. The model uncertainty is given at a 1-σ level, and is computed as the quadratic mean of the input model accuracy and the sky line correction accuracy. Typically this uncertainty will be of the order of 0.1 pixel, i.e., much smaller than the root-mean-squared residual of the lamp calibration and of the sky line correction, depending on the number of fitted points. It should be anyway kept in mind that the model uncertainty can be much larger than that (up to 1 pixel in the worst cases) at the blue and red ends of the fitted wavelength interval. This is because in the pipeline the wavelength solution

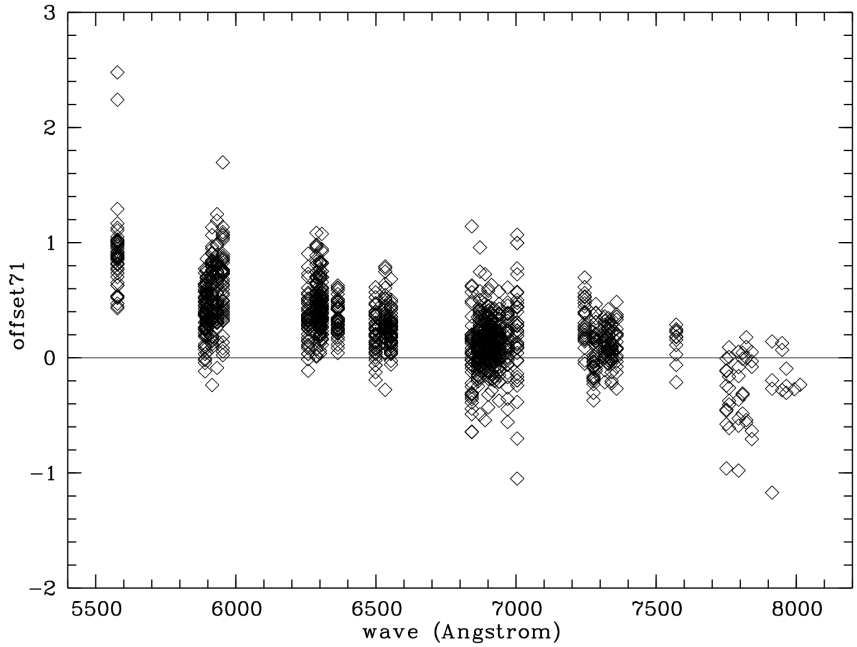

Figure A.3: *Systematic sky line offsets (in pixel) from day-calibration expectation, observed in a VIMOS HR_orange scientific exposure. The offsets from all slits are plotted. Wavelength offsets typically depend on the wavelength, because the spectral dispersion of the VIMOS spectrograph depends on the temperature, which changes between day (calibration) and night (observation). In general a sky alignment of the wavelength solution is due, and it would be appropriate in this case to set the parameter "skyalign" to* 1.

is obtained by fitting a polynomial, rather than a physical model of the instrument behaviour.

A.4.2 Is the sky background properly subtracted?

A quick check on sky subtraction can be made by examining the sky subtracted frames, *mos_science_extracted.fits* and *mos_unmapped_science.fits* (if available, depending on how the recipe was run). The spectra should have a generally smooth look, and will only appear to be noisier in those regions where bright sky lines were subtracted.

The best way to ensure that the sky was subtracted optimally, at least at the positions of the objects to extract, is to check that the residual noise is compatible with the statistical error associated to the extracted object spectra. The extracted spectra are contained in the *mos_science_reduced.fits* image (one extracted spectrum for each row). Their error spectra (at a 1-σ level) are contained in the *mos_sci_error_reduced.fits* image. The regions of the extracted spectra corresponding to a (bright) sky line will include a few noisier points, whose deviation from the spectral continuum should (almost) never pass the 3-σ deviation. If this condition is fulfilled, the sky subtraction is probably as good as it can get.

Note that the subtracted sky can be viewed in the images *mos_science_sky.fits* and *mos_sci_unmapped_sky.fits*. More useful is perhaps the image containing the extracted sky spectra, *mos_sci_sky_reduced.fits*: such spectra are extracted applying to the modeled slit sky spectra exactly the same weights that were used in the object

extraction.

A.4.3 Were all objects detected?

The answer to this question is almost always "no". The pipeline, after removing the instrument signature and the sky background from each slit spectrum, will run an object detection algorithm in order to find all the objects which need to be extracted. There will always be a detection threshold beyond which an object will not be significant enough for selection – no matter what detection algorithm is applied. Using more tolerant detection criteria would not eliminate this threshold effect, and may increase the number of false detections to the point of making the object detection task impractical.

The list of detected objects can be found in the *object_sci_table.fits* file.

A.4.4 Were all the detected objects properly extracted?

As a default the *vmmosscience* recipe would apply an optimal extraction algorithm to each detected object spectrum. This algorithm is only appropriate for point-like objects emitting over (almost) all the extracted wavelength interval, while it is not appropriate for extended objects, and it is ineffective for objects having a spectrum only consisting of emission lines with no continuum.

The statistical noise on the extracted object spectra should in principle decrease if the spectra are optimally extracted. In order to check the improvement of the noise level, it is easy to compare the *mos_sci_error_reduced.fits* images obtained by running the recipe with and without optimal extraction. A 30% increase of the signal-to-noise ratio can be obtained for faint-objects (background-noise limited), while there would be little or no improvement for brighter sources. The photometric accuracy of the optimal extraction can be checked by simply computing the ratio (or the difference) of the *mos_science_reduced.fits* images obtained once with the standard and once with the optimal extraction: the result should be a flat image, displaying no trends or systematic deviations from 1 (or 0).

A.5 Fixing failures in vmmosscience

In this Section a set of possible solutions to the most common problems with the *vmmosscience* recipe is given. It is advisable to try them in the same order as they are listed here. It may be useful to go through this check list even in case the recipe seemed to work well: there might always be room for improvement.

A.5.1 The wavelength calibration is bad

Aligning the wavelength calibration to the position of the observed sky lines may be inaccurate, especially if very few reference lines are used. If a sky line alignment is really required (see previous Section), then action needs to be taken to solve this problem.

Solution: If very few reference sky lines are used, supplying a sky line catalog including more lines (even if weak and/or blended) may help a lot.

Solution: If the wavelength calibration appears to be bad only at the blue and/or red ends of the spectra, go back to the *vmmoscalib* recipe to obtain a more stable wavelength calibration in those regions (e.g., either by adding new reference arc lamp lines, or by decreasing the fitting polynomial degree).

A.5.2 The sky alignment of the wavelength solution failed

In case a blue grism is used, or if a spectrum has a large offset toward the red, no sky lines may be visible within the observed spectral range.

Solution: None. It is however possible to modify the columns of coefficients in the input *mos_disp_coeff.fits* table, if the correction can be evaluated in some other way. For instance, the solution can be shifted by adding a constant value (in pixel) to column "c0".

A.5.3 The sky subtraction failed for resolved sources

In case of extended objects filling most or all of the slit, the evaluation of the sky may be strongly biased by the inclusion of signal that actually belongs to the object to extract. Subtracting this contaminated background would actually destroy the object spectrum.

Solution: The default sky subtraction method (parameter "skylocal") performs very well for point-like sources where plenty of sky is directly observable within the slit. An alternative method is made available for extended objects (parameter "skyglobal"). Setting "skylocal" to *false* and "skyglobal" to *true* would subtract from all spectra a supersampled model of the median sky spectrum observed in all slits. This method would perform optimally only in case the spectral resolution were the same all over the detector: in practice, this method is always less accurate than the "skylocal" method. But even if it is less accurate, this method remains the best friend of the extended sources. It is always possible to process the scientific exposures in both ways, one for processing point-like sources and the other for processing spatially resolved sources.

A.5.4 The sky subtraction failed for curved or tilted slits

Obvious residuals related to the sky subtraction are visible on the extracted slit spectra.

Solution: Change sky subtraction method: set "skylocal" to *false* and "skymedian" to *true*. The difference between the two methods is that "skylocal" would subtract the sky *before*, and "skymedian" *after* the rectification of the spectral data. The second method performs very poorly in comparison to the first, but in the case of curved or slanted slits there is at the moment no other choice than using it.

A.5.5 Cosmic rays are not removed

As a default the *vmmosscience* recipe does not remove cosmic rays hits, leaving them on the sky-subtracted slit spectra: if the optimal spectral extraction is applied, most of the cosmics are removed anyway from the extracted spectra. Optimal extraction is however not always applicable, especially in the case of resolved sources.

Solution: Set the "cosmics" parameter to *true*. This will apply a cosmics removal algorithm to the sky subtracted spectra. The removed cosmic rays hits will be included in the (modeled) sky images, *imos_science_sky.fits* and *mos_sci_unmapped_sky.fits*.

A.5.6 The sampling of the remapped scientific spectra is poor

When the slit spectra are rectified and wavelength calibrated, they are remapped undistorted to images such as *mos_science_sky_extracted.fits* or *mos_science_extracted.fits*. This remapping may be judged to undersample the signal along the dispersion direction.

Solution: Change the value of the "dispersion" parameter. This parameter doesn't need to be identical to the one used in the *vmmoscalib* recipe.

It should be noted, however, that making the sampling step smaller will not really increase the information contained in the remapped spectra. As a matter of fact, even maintaining a resampling step close to the original CCD pixel size, the remapped pixel values would still be obtained by interpolating the values from a number of original pixels that are close to the interpolation point: nearby interpolation points would surely share common information, and this is what introduces correlated noise in the result. Decreasing the resampling step would just increase this effect. In general, working on remapped spectra means to accept that the spectral signal was heavily manipulated, and it is for this very reason that the *vmmosscience* recipe also produces reduced – but still unmapped – spectra, as in the *mos_unmapped_science.fits* image.

A.5.7 The extracted spectra are normalised in time

The default behaviour of this recipe is to normalise the results to the unit exposure time.

Solution: Set the parameter "time_normalise" to *false*.

A.5.8 There are often spurious objects detected at the slit edges

As a default the *vmmosscience* recipe excludes objects that are detected within 3 pixels from the slit ends. This might not be enough in some cases.

Solution: Increase the value of the "slit_margin" parameter.

A.5.9 Some "obvious" objects are not detected

Examining the *mos_science_extracted.fits* and *mos_unmapped_science.fits* images it may appear that some clearly visible object spectra are not detected (let alone extracted) by the recipe.

Solution: Setting "cosmics" to *true* (cleaning cosmic rays hits) may help.

Solution: Try different set of values for the parameters "ext_radius" and "cont_radius".

B Software Installation

The VIMOS pipeline can be installed in different ways: via package repositories, via the `install_esoreflex` script or manually installing the software tar files.

The recommended way is to use the package repositories if your operating system is supported. The `macports` repositories support OS X 10.11 to 10.14, while the `rpm/yum` repositories support Fedora 26 to 29, CentOS 7, Scientific Linux 7. For any other operating system it is recommended to use the `install_esoreflex` script. Please note that in case of major or minor (affecting the first two digit numbers) Reflex upgrades, the user should erase the `$HOME/KeplerData`, `$HOME/.kepler` directories if present, to prevent possible aborts (i.e. a hard crash) of the esoreflex process.

B.1 System Requirements

The data reduction for VIMOS may require substantial memory resources. However, the amount of memory required to run a recipe depends on a number of factors, e.g. for imaging the number of files, stacking method, sky coverage, maximum jitter offset, etc. Below is a table with the minimum resident memory needed by the imaging pipeline to process data as a function of recipe and number of input files. Note that the number of input files here is the number of raw files, i.e. divide this by 4 to get the number of coherent sets of chips. This table assumes that all recipe parameters are set to their default and a small jitter offset between frames. In general, the execution time of a recipe can be shortened by choosing 'fast' stacking, but this will increase the memory use (see description of `stk_fast` and `stk_nfst` parameters in §9.1).

Recipe	# science frames	Min. RAM
vimos_standard_process	4	1.8 GB
vimos_science_process	12	6.2 GB
vimos_science_process	32	9.2 GB

Table B.1: *Minimum memory requirements for selected VIMOS-IMG recipes*

B.2 Installing VIMOS pipeline via `macports`

This method is supported for the macOS operating system. It is assumed that macports (*http://www.macports.org*) and java JDK 1.8 are installed. The rest of the dependencies are automatically handled by macports itself. Please read the full documentation at
http://www.eso.org/sci/software/pipelines/installation/macports.html.

B.3 Installing VIMOS pipeline via `rpm/yum/dnf`

This method is supported for Fedora 26 to 29, CentOS 7, Scientific Linux 7 operating systems. All software dependencies are automatically handled by `yum/dnf`. Please read the full documentation at
http://www.eso.org/sci/software/pipelines/installation/rpm.html.

B.4 Installing VIMOS pipeline via `install_esoreflex`

This method allows to install the VIMOS pipeline, including the Reflex workflow and Reflex itself. It can also be used to install at the same time any other pipeline distributed by ESO. The software pre-requisites for `Reflex 2.11.5` may be found at:
http://www.eso.org/sci/software/pipelines/reflex_workflows. The user has to ensure that those pre-requisites are installed beforehand.

To install the `Reflex 2.11.5` software and demo data, please follow these instructions:

1. From any directory, download the installation script:

```
wget https://ftp.eso.org/pub/dfs/reflex/install_esoreflex
```

2. Make the installation script executable:

```
chmod u+x install_esoreflex
```

3. Execute the installation script:

```
./install_esoreflex
```

   and the script will ask you to specify three directories: the download directory `<download_dir>`, the software installation directory `<install_dir>`, and the directory to be used to store the demo data `<data_dir>`. If you do not specify these directories, then the installation script will create them in the current directory with default names.

4. You will be asked whether you want to use your Internet connection. Unless you want to reuse already downloaded packages (only advanced users), use the default Yes.

5. You will be given a choice of pipelines (with the corresponding workflows) to install. Please specify the numbers for the pipelines you require (if you want to support others together with VIMOS), separated by a space, or type "A" for all pipelines.

6. For the pipelines to be installed you will be prompted for the demo data sets to be installed. Type "A" for all demo datasets. Take into account that if you are installing in a directory that already contains data, it won't be removed.

7. The script will also detect whether previous versions of the workflows or Reflex were installed and in this case you have the option to update links or remove obsolete cache directories. It is advised to use the defaults.

8. If some of the prerequisite binaries for `Reflex` are not under one of the paths indicated by the command,

```
getconf PATH
```

then you will need to add the appropriate paths as a colon separated list to the `esoreflex.path` parameter in the configuration file `<install_dir>/etc/esoreflex.rc`. This will usually be necessary when the FITS viewer (`fv`) is installed outside of `/usr/bin`. As an example, assume `fv` is installed into the directory `/usr/local/fv5.4`, the file `esoreflex.rc` should then have the line setting `esoreflex.path` look similar to the following:

```
esoreflex.path=/usr/local/fv5.4
```

In the case of OS X `/Applications/fv.app/Contents/MacOS/` is the typically installation directory. Thus, this should be similar to the following line instead:

```
esoreflex.path=/opt/local/bin:/Applications/fv.app/Contents/MacOS
```

9. To start `Reflex`, issue the command:

```
<install_dir>/bin/esoreflex
```

It may also be desirable to set up an alias command for starting the `Reflex` software, using the shell command `alias`. Alternatively, the `PATH` variable can be updated to contain the `<install_dir>/bin` directory.

C Abbreviations and acronyms

ADF	Aperture Definition File
ADM	Aperture Definition in mm
CPL	Common Pipeline Library
CRV	VIMOS spatial curvature model
DFS	Data Flow System
DO	Data Organiser
DPD	Data Processing Department
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
PSO	Paranal Science Operations
PWS	Pipeline WorkStation
QC	Quality Control
SAO	Smithsonian Astrophysical Observatory
SDD	Software Development Division
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
WCS	World Coordinate System