omgchain

February 1, 2016

Abstract

This document describes the use of software for the automated processing and extraction of spectra from Optical Monitor images obtained with the Optical and UV grisms ("grism-filters").

1 Instruments/Modes

Instrument	Mode
OM	IMAGING

2 Use

pipeline processing	yes
interactive analysis	yes

3 Description

This package contains a PERL script which selects, from an ODF, images taken with the OM grism filters (if present) and processes them using a chained sequence of SAS tasks to produce OM grism spectral products. **Omgchain** calls a number of SAS packages that prepare the grism images for processing (e.g. insert necessary keywords into the file headers, correct images for distortion, rotate them to optimise spectral extraction, etc.), detect point-like and extended source/spectral features in the OM Science Window (OSW)s, identify and extract spectra within the OSW and then generate corresponding products (FITS-files and plots in PS and PDF formats).

3.1 Grism data

The XMM Optical Monitor can produce data in two modes: Imaging and Fast (timing). The **omgchain** package reduces the OM Imaging mode data taken with either the UV or Optical grism filters (prism filters with diffraction gratings on the back which facilitate low-resolution spectroscopy). There are two filter positions of the filter wheel in front of the OM detector which are occupied by grisms (hereafter called grism-filters), one designed for the UV spectral range and another for the visual. Thus, in the OM

Imaging mode one can obtain spectra of all the sources within the OM science window in two different spectral bands. The grism-filters are labelled as following (for instance, in the file headers):

- Grism1 (filter wheel position 1000: UV-grism);
- Grism2 (filter wheel position 200: visual grism).

The grisms disperse the incoming photons with the dispersion angle depending on the photon energy. The resulting images contain two (rarely –three) spectral orders. The zero-order spectra can be considered as point-like - they are used as reference points for the determination of wavelengths in the corresponding first-order spectra.

The specific design of the OM filter wheel and grism optical systems caused the spectral images to be rotated with respect to the pixel grid of the OM CCD detector by $\approx 6.3°$ (anticlockwise) for the UV grism and $\approx -5.6°$ (clockwise) for the visual-grism.

3.2 Grism chain capabilities

The grism chain searches for those files in the current ODF whose FILTER keyword is set to *Grism1* or *Grism2*. If such files (grism images) are found, **omgchain** calls a number of OM-specific SAS tasks in order to correct these images for distortion, align them with respect to the pixel grid (for spectral extraction optimisation), and identify and extract spectra of sources for which both the zero- and first orders lie within the OSW. Where no grism images exist or when there are no usable spectra in the OSW, **omgchain** finishes smoothly, generating a warning message. Thus, running **omgchain**, the user does not have to specify input or output files in the task parameter list, although a number of parameters may be set by the user (**omgchain** has no mandatory parameters). **Omgchain** will attempt to extract a spectrum from a default region specifically related to the location of the target object (based on the specified target coordinates in the ODF files), even when no corresponding spectrum is identified by the automated detection.

4 Grism chain procedure

Omgchain employs seven SAS tasks (Fig.1). The first four tasks are preparatory. For instance, if the OSW image is split into sub-windows, then these sub-windows are combined into a single file (see section 4.1). The necessary keywords, if missed, are inserted into the file headers. These tasks also implement the above-mentioned corrections to the image.

The other three tasks execute the source detection and spectral identification procedure, and produce corresponding output.

4.1 Combination of sub-windows in a single image

The SAS task **omcomb** is called by **omgchain** when the current ODF contains files obtained in the Engineering-2 (full frame, low resolution) mode of OM. In this mode the data from the OSW is split into four sub-windows. For instance, the 2048 × 2048-pixel image can be placed in four image files, each containing a vertical 512 × 2048-pixel strip. Combining the Engineering-2 files into one image is the default option, which, if desired, can be switched off.

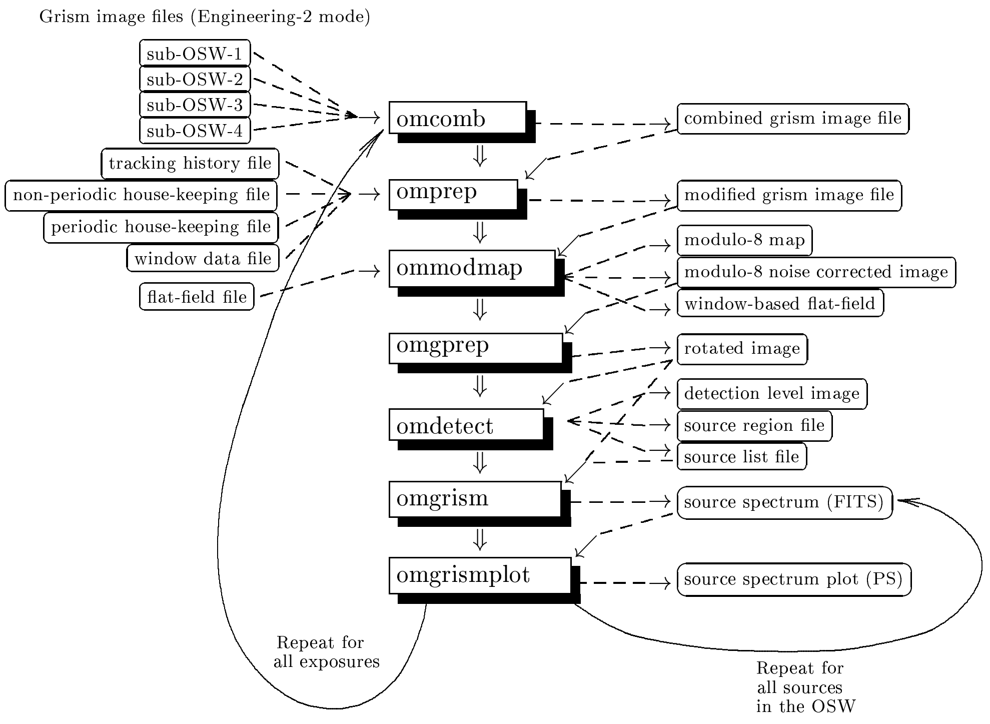

Figure 1: OM Grism Mode Chain

4.2 Header preparation

The task **omprep** inserts necessary header information into the input file (e.g., the spacecraft pointing direction, frametime, dead fraction of the frametime, etc.)

4.3 Modulo-8 noise correction

OM images are affected by a fixed modulo-8 pattern originating from the on-board centroiding of photon events to one eighth of the physical CCD pixel (see [1] or [3])

The task **ommodmap** implements modulo-8 pattern correction by redistributing the photon counts over the 8 × 8-pixel areas of the image. Currently there are three available correction methods, and the user can switch between them in order to achieve optimal results. The first (default) method generates the mod-8 map by searching for the regular pattern in the image, derived from 'source' free regions, and then the photon events are evenly resampled within each 8 × 8 cell of the found modulo-8 map. This method, which is employed in the OM Imaging chain, maintains the total number of events within each cell, thus the photometry is not affected. However, images with sparse backgrounds (short exposures, or far UV images) may contain insufficient counts to generate a modulo-8 map of adquate accuracy.

The second method exploits the well populated, cummulative OM flat field image (which can be taken from the corresponding CCF). The pattern can be well characterised from this image and the correction applied to the ODF data – it assumes the pattern is stable from observation to observation. In this case the modulo-8 pattern map is created using the OM flat field, and then the photon events are redistributed in accordance with this map. With this, the total number of the photon events is conserved, as in the first method.

The third method uses multiplication of the input image by the flat field (the latter is assumed to be normalised to unity).

The user can choose the modulo-8 correction method by specifying the parameter *mod8correction* (see Section 5).

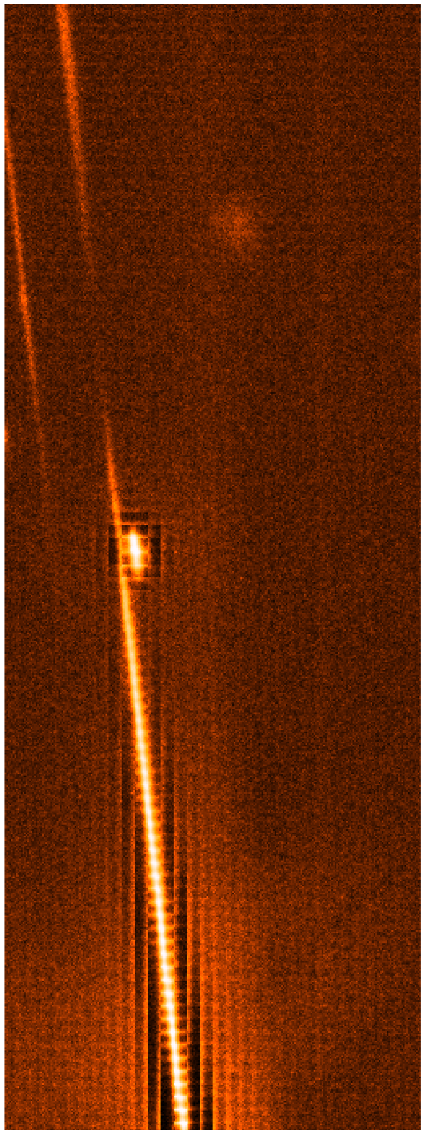

Figure 2: OM grism image with a bright spectrum affected by the modulo-8 noise and coincidence losses.

It should be emphasised that while these approaches produce acceptable results for fainter and intermediate brightness spectra, none of the modulo-8 corrections methods work well in the vicinity of bright features, in part due to the presence of coincidence loss effects.

In addition, the above methods assume that the background/flat-field completely represents the modulo-8 pattern but this is not strictly correct and so small errors at the ~ 1 sub-pixel level may be introduced. An alternative, more formally correct approach will be investigated in the near future.

As an example, Fig.2 illustrates a spectrum badly affected by coincidence losses and modulo-8 noise, along with a couple of spectra belonging to fainter sources which are not affected by the coincidence loss. The altered spectrum can readily be spotted by the presence of black square regions around the spectrum, with almost vanishing counts (down to a few counts per pixel or zero), in contrast with high levels of counts of the spectrum itself (up to a few thousand). This effectively results in modulation of the extracted spectra with the period corresponding to the modulo-8 pattern. The effect becomes noticeable when the count rate exceeds ~ 0.5 count/sec/pixel. Reaching this level depends on the brightness and spectral type of the source and cannot be rigorously quantified. Approximately it corresponds to the source's magnitudes $10^m - 13^m$.

4.4 Preparation of the grism image for spectra extraction

The task **omgprep** introduces some changes into the grism image files which are necessary for running the source detection and spectral extraction tasks. **Omgprep** inserts some missing but important keywords into the header of the grism image FITS file. The image is then corrected for distortion. The image is subsequently rotated so that the grism dispersion direction is aligned with respect to the pixel readout columns of the image. This alignment is helpful when identifying and extracting spectra. **Omgprep** also checks the positional angles of the spectra which can be useful for calibration purposes. If required, scattered light features (e.g., central enhancement) can be removed from the rotated image by using the parameter *removescatteredlight=yes*, but one must keep in mind that this operation essentialy increases the working time of *omgprep*.

The original unrotated image is shown in Fig.3. Images prepared for running the source detection and spectra extraction tasks are shown in Fig.4 (for the visual grism) and 7 (for the UV-grism). These have been processed by all **omgchain** tasks up to and including **omgprep**.

4.5 Source detection

The task **omdetect** runs on the image with spectra aligned to the pixel columns of the image, i.e. the output of **omgprep**. It creates a list of detected sources, which becomes the input for the task **omgrism**.

Practically all the sources detected by **omdetect** in the grism images are classified as extended. However, zero-order spectra can be distinguished from the first-order spectra by their major semi-axes that are given for each source in the source list produced by **omdetect**. In the case of the UV grism, the first- and the second order spectra overlap and they are usually detected as a single source with a large major semi-axis (the flux-calibrated part of the output (UV) spectra are currently truncated at ~ 3500 Ådue to the problem of separating the two orders and the parallel lack of adequate calibration).

There is also a possibility of spurious source detection due to the residual modulo-8 noise pattern. **Omdetect** incorporates a mechanism which attempts to discriminate (reject) spurious detections. However, this can cause non-detection of some spectra which are, nonetheless, detectable by visual inspection). The user can control the discrimination level through the parameter *nsigma* passsed by **omgchain** to **omdetect** (see Section 5). Further identification and cross-examination of the detected zero- and first-order spectra is left to the task **omgrism** (see the next section). An example of an image (visual-grism) with the sources detected by **omdetect** is shown in Fig.5.

4.6 Spectral extraction

Spectral extraction is implemented in the task **omgrism**, which runs over all sources in the source list (created by **omdetect**), cross-identifies the zero- and first-order spectra and decides which of them are usable for extraction. The output of the task is a FITS-file containing the spectra, each in a separate FITS extension. **Omgrism** allows spectral extraction using three possible algorithms: simple summing of counts above background in the cross-dispersion direction; one-dimensional Gaussian fit; and Optimal

Extraction (Horne's algorithm [2]). The implementation of the latter method is still under development. Examples of spectra extracted by **omgrism** are given in Fig.6 (corresponding to the lower source in Fig.5) and Fig.8 (corresponding to Fig.7).

The coincidence loss correction is currently not applied (the algorithm is under developing). The background counts are scaled to the spectrum extraction region, smoothed along the spectrum and subtracted from it. The wavelengths are computed by using the CAL routine with the distances between the centre of the zero-order image and the current spectrum cross-section as the input for the CAL routine.

5 Parameters

This section documents the parameters recognized by this task (if any).

Parameter	Mand	Type	Default	Constraints

inpdirectory	no	string	none	

Directory path-name of the input files directory - defaults to the current directory.

outdirectory	no	string	none	

Directory path-name of the output files directory - defaults to inpdirectory

comment	no	string	none	

User's comments for output

nsigma	no	real	2	≥ 1

Number of σ above the background required for a detection (this parameter is passed to **omdetect**, see the description of **omdetect**.

combine	no	boolean	yes	

Condition for combining the Engineering-2 subwindows

spectrumhalfwidth	no	real	-8.0	-20 to 20

Half-width of the spectrum extraction region (in pixels, if negative, and in FWHWs otherwise)

bkgoffsetleft	no	real	0.	-20 to 20

Offset of the left background extraction region from the edge of the spectrum extraction area; in pixels, if negative, or in FWHWs otherwise.

bkgwidthleft	no	real	-8.	-40 to 40

Width of the left background extraction region; in pixels, if negative, or in FWHWs otherwise.

bkgoffsetright	no	real	0.	-20 to 20

Offset for the right background extraction region; in pixels, if negative, and in FWHWs otherwise.

bkgwidthright	no	real	-8.	-40 to 40

Width of the right background extraction region; in pixels, if negative, or in FWHWs otherwise.

spectrumsmoothlength	no	integer	0	≥ 0

Length of the smoothing window for smoothing the extracted spectra, if necessary. Values 0 or 1 of this parameter imply no smotthing.

extractionmode	no	integer	0	≥ 0

Switch between different extraction modes. The value 0 corresponds to the normal extraction (summation of counts in the cross-dispersion direction); 1 corresponds to the Optimal Extraction (Horne's algorithm [2]); 2 corresponds to the spline smoothing of the spectrum cross-section; 3 corresponds to the Gaussian fit of the spectrum cross-section.

extractfieldspectra	no	boolean	no	

Condition for extraction either only the target object spectrum or all available spectra of the sources in the field

mod8correction	no	integer	1	0 to 3

Condition for removing the modulo-8 noise: 0: correction not applied; 1: correction applied using the modulo-8 map extracted from the input image; 2: correction applied using the modulo-8 map extracted from the OM CCF flat field; 3: correction applied multiplying the input image by the OM CCF flat field.

plotbinsize	no	integer	0	≥ 0 Å

Size of spectrum wavelength bins for the output plot (in Å)

plotflux	no	integer	2	0 to 2

Flag for plotting the spectrum only (value 0), the background only (value 1), or both of them (value 2)

scalebkgplot	no	boolean	no	

Condition for scaling the background plot differently from the spectrum plot

removescatteredlight	no	boolean	no	

Condition for removing scattered light features by the task omgprep

All parameters are optional, since **omgchain** searches for the grism images in the working directory. If at least one such image is found it is processed, otherwise **omgchain** finishes without producing any file output.

6 Input Files

Before attempting to use the **omgchain**, first run the **odfingest** SAS task. This will create a *SUM.SAS file from *SUM.ASC file that comes with the ODF. It is better to keep this in the same directory as the ODF. With this, and with the other files listed below, which should already be in the ODF, it should be possible to process the OM images with grism spectra files.

1. rrrr_iiiiiiiii_OMX00000NPH.FIT - OM Non-periodic Housekeeping file
2. rrrr_iiiiiiiii_OMX00000PEH.FIT - OM Periodic Housekeeping file
3. rrrr_iiiiiiiii_SCX00000FFX.FIT - PPS OSW Flatfield Image
4. rrrr_iiiiiiiii_SCX00000TCS.FIT - Spacecraft time correlation file
5. rrrr_iiiiiiiii_SCX00000ATS.FIT - Spacecraft attitude history file

For each exposure to be processed there are the following files:

1. rrrr_iiiiiiiii_OMSeeewwIMI.FIT - OM Image file with the grism spectra
2. rrrr_iiiiiiiii_OMSeeewwWDX.FIT - OM Priority Window Data Auxiliary file

where **rrrr** is the 4 digit XMM rev. number, **iiiiiiiii** is the 10 digit observation id, **eee** is the exposure number (e.g. 006 etc.), and **ww** is the window identifier (00 or 01).

7 Output Files

1. OSW combined image (in the case of the Engineering-2 observation; produced by **omcomb**): giiiiiiiiiiOMSeeeCIMAGE0000.FIT.
2. Intermediate image (produced by omprep) giiiiiiiiiiOMSeeeIMAGEI000.FIT.
3. Partial Flat-field image corresponding to the processed OSW (produced by the task **ommodmap**) giiiiiiiiiiOMSeeeFLAFLD000.FIT.

4. OSW Mod8-noise map (produced by **ommodmap**)

   giiiiiiiiiOMSeeeMOD8MP000.FIT.

5. Detector-coordinate image file (produced by **ommodmap**; that is, having mod-8 noise reduced)

   giiiiiiiiiOMSeeeIMAGE_000.FIT.

6. OSW rotated image (spectra are aligned with the columns of the image; produced by **omgprep**)

   piiiiiiiiiOMSeeeRIMAGE000.FIT.

7. OSW source list (produced by **omdetect**)

   piiiiiiiiiOMSeeeSWSRLI001.FIT.

8. Source region file (ASCII) corresponding to the sources detected in the rotated image (produced by **omdetect**). The regions are ellipses of the default colour (usually, green).

   piiiiiiiiiOMSeeeREGION001.ASC.

9. PPS list of the extracted spectra (FITS file produced by **omgrism**), that contains X and Y coordinates of the zero- and first-order spectra

   piiiiiiiiiOMSeeeSPECLI000.FIT.

10. Spectra region file (ASCII) corresponding to the spectra list file (output of the **omgrism**). To distinguish the spectra regions from those produced by **omdetect** the former are rectangular boxes coloured in red.

    piiiiiiiiiOMSeeeSPCREG001.ASC.

11. PPS source spectrum FITS file (produced by **omgrism**)

    piiiiiiiiiOMSeeeSPECTR000.FIT.

12. PPS spectrum plot file - Postscript (produced by **omgrismplot**)

    giiiiiiiiiOMSeeeSPECTR000.PS.

13. PPS spectrum plot file - PDF (converted from the Postscrip by *ps2pdf* standard routine)

    piiiiiiiiiOMSeeeSPECTR000.PDF.

14. Undistorted image (similar to giiiiiiiiiOMSeeeIMAGE_000.FIT but with distortion correction applied)

    uiiiiiiiiiOMSeeeIMAGE_0000.FIT

15. Plot of the spectra regions shown on the rotated grism image

    piiiiiiiiiOMSeeeSPCREG0001.PS

16. Image of the smoothed background (corresponding to the rotated image piiiiiiiiiOMSeeeRIMAGE000.FIT) used for removing scattered light features (in the case of setting the parameter *removescatteredlight* to "yes")

    giiiiiiiiiOMSeeeRIMBKG0000.FIT

8 Running The Chain

The location of the ODF files is specified by either setting the environment variable `SAS_ODF` (setenv `SAS_ODF` directory-path) to a directory or to a SAS summary file, or by specifying the input directory (omgchain inpdirectory=directory path). For the latter, **omgchain** will extract the directory path from the SAS summary file.

8.1 Command Line Examples

1. Use command 'omgchain' to process the ODF data set in the current directory or to where `SAS_ODF` has been set to, and to place the product files in the current directory.
2. Use command 'omgchain outdirectory=output_directory_path' to process the ODF data set in the current directory or to where `SAS_ODF` has been set to, and to place the output files into the specified directory.
3. Use command 'omgchain inpdirectory=input_directory_path' to process the OM ODF data set in the specified directory, and to place the product files into the current directory.
4. Use command 'omgchain inpdirectory=input_directory_path outdirectory=output_directory_path' to specify both the location of the ODF data set and the directory where the product files are to be placed.

9 Comments

- If the chain runs to completion, a number of important output files will appear. The chain product filenames have the extension '.FIT'. The names of the files start with 'p' for the products and with 'g' for the intermediate or auxiliar images and source lists.
- Some images of bright sources can be badly affected by the Mod-8 noise pattern (combined with coincidence loss). The spectra of such sources are unreliable.
- It is wise to check the output files (p* and g* files) from **omgchain**. A useful first step is to inspect the output PDF-spectra curve plots - watch out for frequent drop-outs, isolated high or substantial negative values. Establish that the background spectrum is reasonable. One should also examine the rotated image (produced by **omgprep**), for example, with the ftool, *fv* or with the *ds9* programme. Check whether all the sources are detected. For this purpose overlay the image source list onto skymap using*implot*, e.g.

  implot set= sky_image_file.FIT **withsrclistset=y srclistset=**osw_source_list_file.FIT **device=**your_device (e.g. **/XW**) **itv=1 radius=3 maxsrc=10 colour=7**.

10 Future developments

Spectrophotometry and wavelength calibration is undergoing improvement. The proper use of grism-specific CAL routines to be introduced. Further tests, checks and improvements of the spectra extraction algorithms will be made. An interactive task for visual inspection and spectral extraction is being developed. Spectrum smoothing possibility to be introduced. Coincidence loss correction to be included.

References

[1] Kawakami H., Bone D., Fordham J., and Michel R. The effect of event shape on centroiding in photon counting detectors. *Nucl. Instr. Methods in Phys. Res.*, A 348:707–712, 1994.

[2] Horne K. An Optimal Extraction Algorithm for CCD Spectroscopy. *PASP*, 98:609, 1986.

[3] Michel R., Fordham J., and Kawakami H. Fixed pattern noise in high-resolution, CCD readout photon-counting. *Mon. Not. Astron. Soc.*, 292:611–620, 1997.

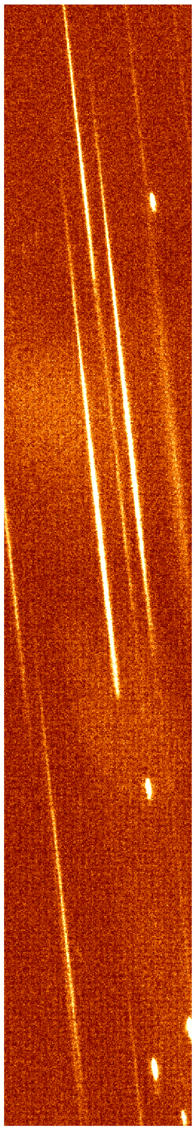

Figure 3: Original grism image (no Mod-8 correction applied, no rotation); visual spectral band

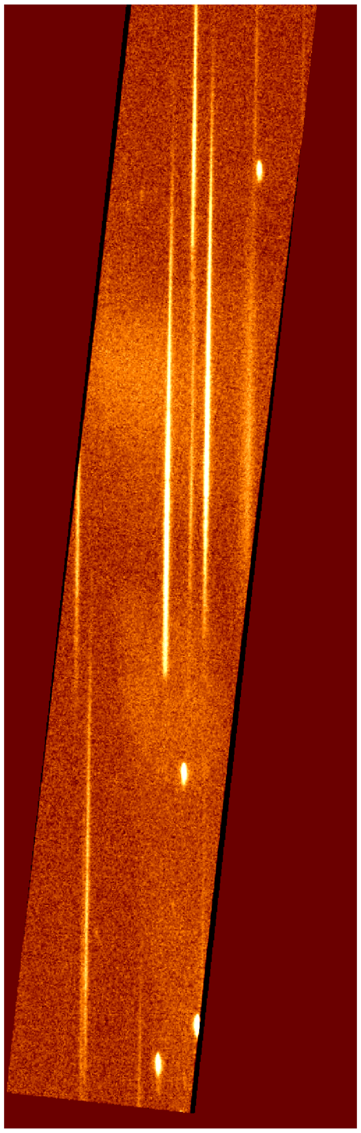

Figure 4: Mod-8 noise corrected and rotated grism image; visual spectral band

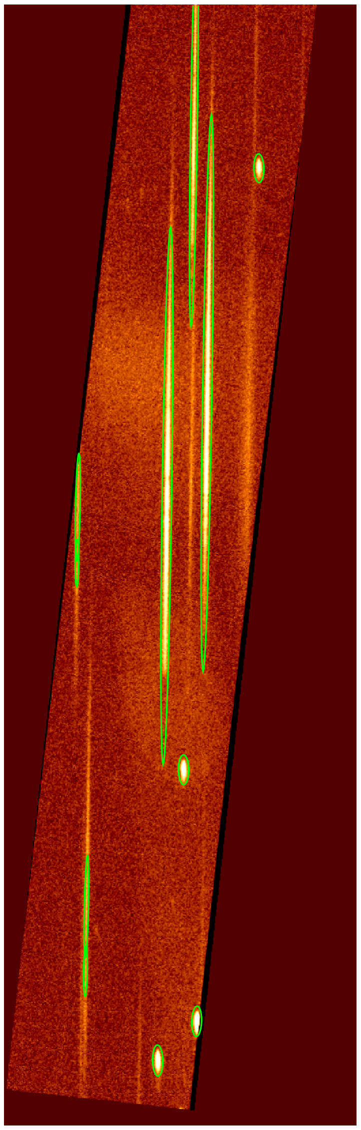

Figure 5: Visual-grism image with the sources detected by **omdetect** (the source regions are marked with the green ellipses)

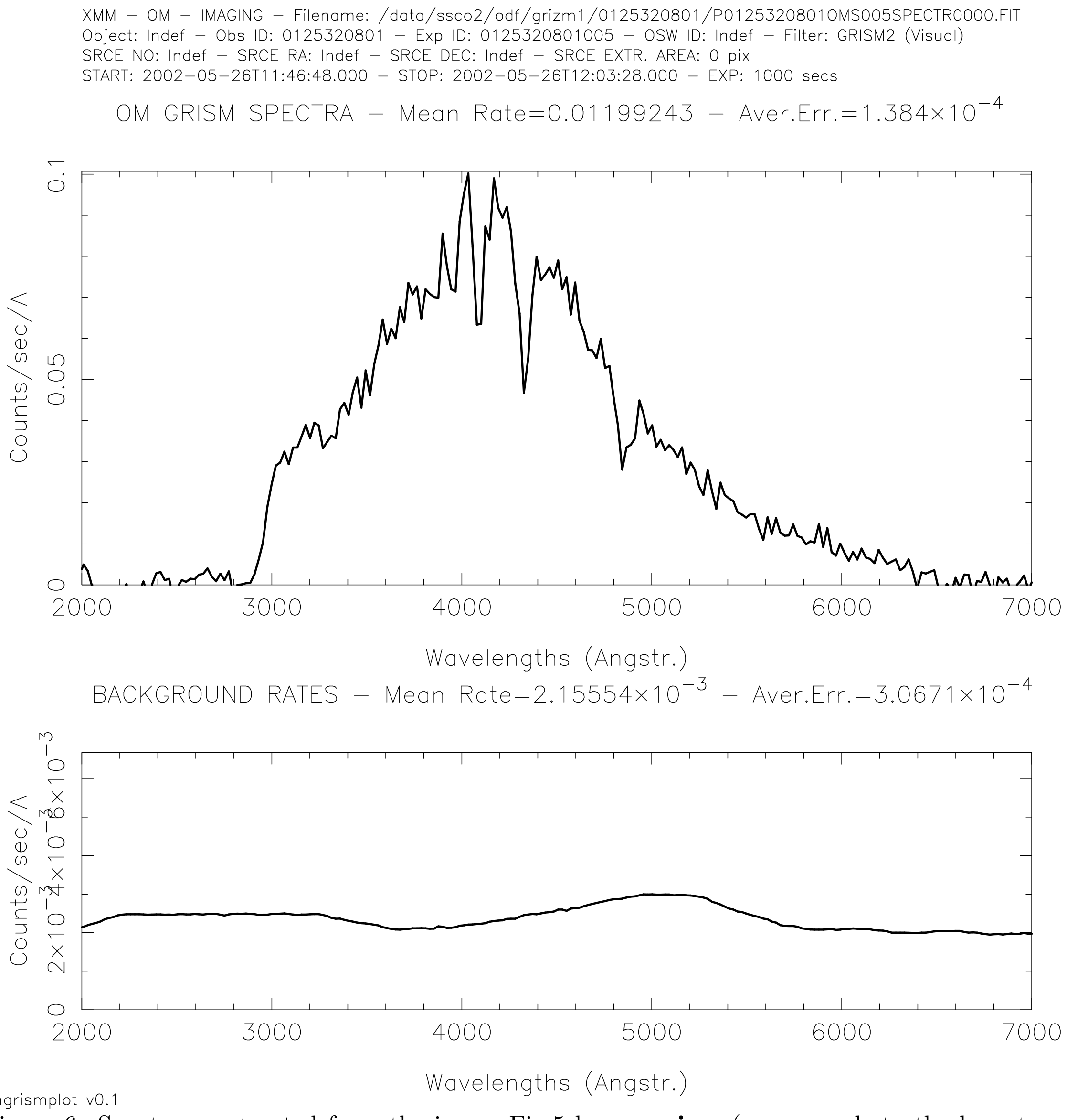

Figure 6: Spectrum extracted from the image Fig.5 by **omgrism** (corresponds to the lowest source in Fig.5)

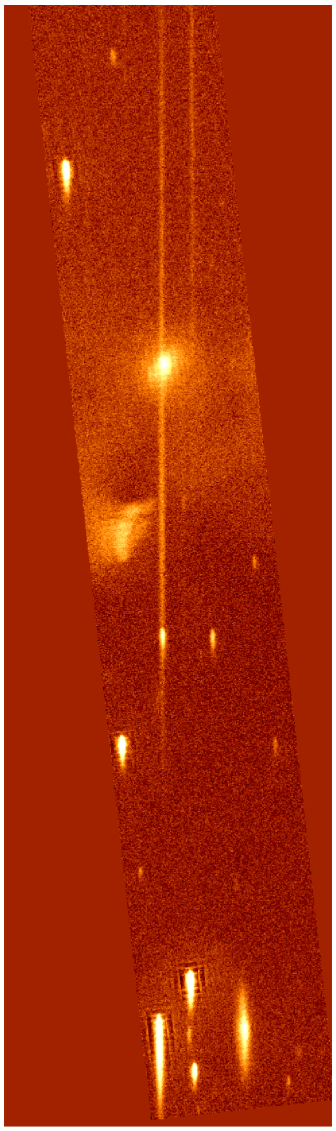

Figure 7: UV-grism image (mod-8 noise corrected; image rotated). Note two zero-order spectra overlapping the first order spectrum (target).

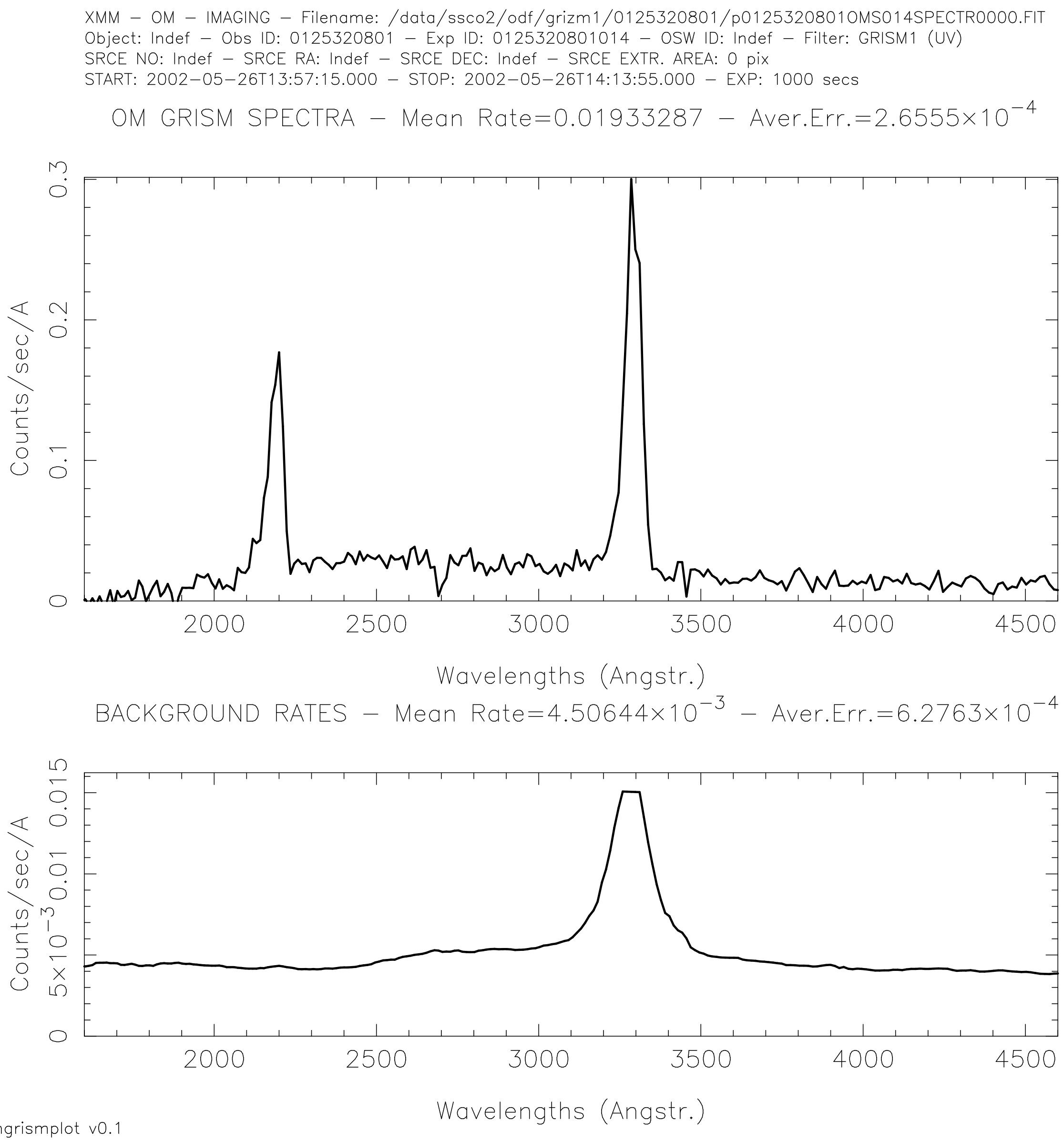

Figure 8: Spectrum extracted from the image Fig.7 by **omgrism**. Note a zero-order spectrum oberlapping with the first-order target spectrum.