



aXe User Manual version 2.3

M. Kümmel, J. Walsh, H. Kuntschner (ST-ECF)
and H. Bushouse (STScI)

June 22nd, 2011

Contents

1 Description	9
1.1 What is aXe?	9
1.2 Slitless Spectroscopy	10
1.3 Apertures and Beams	11
1.4 Pixel Extraction Tables (PET)	12
1.5 Generating 1D spectra	13
1.6 Sky Background	14
1.6.1 Global Background Subtraction	14
1.6.2 Local Background Subtraction	15
1.7 Contamination	15
1.7.1 Geometrical contamination	15
1.7.2 Quantitative contamination	16
1.8 Drizzling of PETs	19
1.9 Optimal weighting	22
1.10 Extraction parameters in aXe	24
1.11 Flux conversion	26
1.12 aXe Visualization	26
1.13 Acknowledgement	28
2 Installing aXe	29
2.1 Package Structure	29
2.2 Requirements	29
2.3 aXe Distribution	29
2.4 Installing an aXe source distribution	29
2.5 Validating the aXe installation	30
2.6 aXe Support	30
3 Using aXe	31
3.1 The input data	31
3.2 Preparing the extraction	32
3.2.1 MultiDrizzle	32
3.2.2 Master catalogue	33
3.2.3 Preparing the Input Object Lists	34
3.2.4 Preparing the fluxcube files	35
3.3 Extracting spectra	36
3.3.1 Reduction Strategy	36
3.3.2 Input Image List	37
3.3.3 The aXe Configuration Files	38

3.3.4	Example reductions for the different scenarios	40
3.4	The requirements	42
3.5	Tuning tips	43
3.5.1	Wavelength dependence of the PSF	43
3.5.2	Extraction of higher grism orders	43
3.5.3	Limits for contamination	43
4	aXe tasks	47
4.1	IOLPREP	47
4.1.1	Usage	48
4.1.2	Parameters	48
4.1.3	Output	48
4.2	FCUBEPREP	48
4.2.1	Usage	48
4.2.2	Parameters	49
4.2.3	Output	49
4.3	AXEPREP	49
4.3.1	Usage	50
4.3.2	Parameters	50
4.4	AXECORE	51
4.4.1	Usage	51
4.4.2	Parameters	52
4.4.3	Output	53
4.5	DRZPREP	54
4.5.1	Usage	54
4.5.2	Parameters	54
4.5.3	Output	55
4.6	AXEDRIZZLE	55
4.6.1	Usage	56
4.6.2	Parameters	56
4.6.3	Output	57
4.7	SEX2GOL	58
4.7.1	Usage	58
4.7.2	Parameters	58
4.7.3	Output	59
4.8	GOL2AF	59
4.8.1	Usage	60
4.8.2	Parameters	60
4.8.3	Output	60
4.9	BACKEST	60
4.9.1	Usage	61
4.9.2	Parameters	61
4.9.3	Output	62
4.10	AF2PET	62
4.10.1	Usage	62
4.10.2	Parameters	62
4.10.3	Output	63
4.11	PETCONT	63
4.11.1	Usage	63
4.11.2	Parameters	63

CONTENTS	5
4.11.3 Output	64
4.12 PETFF	64
4.12.1 Usage	65
4.12.2 Parameters	65
4.12.3 Output	65
4.13 PET2SPC	65
4.13.1 Usage	65
4.13.2 Parameters	65
4.13.3 Output	66
4.14 STAMPS	66
4.14.1 Usage	67
4.14.2 Parameters	67
4.14.3 Output	67
4.15 DRZ2PET	67
4.15.1 Usage	68
4.15.2 Parameters	68
4.15.3 Output	68
4.16 AXEGPS	68
4.16.1 Usage	69
4.16.2 Parameters	69
4.16.3 Output	69
5 Configuration of aXe tasks	71
5.1 Environment Variables	71
5.2 Configuration Files	72
5.2.1 Main Configuration File	72
5.2.2 Example of a Main Configuration file	78
6 aXe Calibration Files	79
6.1 Sensitivity Curve	79
6.2 Flat field	79
7 File Formats	81
7.1 Input Images	81
7.2 Fluxcube file	82
7.3 Input Image List	83
7.4 Input Object List	84
7.5 Grism Object List	85
7.6 Aperture File	85
7.7 Background Estimate File	86
7.8 Pixel Extraction Table	87
7.9 The Drizzle Prepare File	88
7.10 The 2D Drizzled Grism Image	88
7.11 Extracted Spectra File	89
7.12 Stamp Image File	90
7.13 Weight Image File	90
7.14 Contamination File	90
A Virtual slits	91

Preamble

The Space Telescope - European Coordinating Facility (STECF) was responsible for supporting the slitless spectroscopic modes of the Hubble instruments ACS and WFC3 until late 2010.

As of January 2011, all support (software, calibration and user support) is being provided by the Space Telescope Science Institute (STScI) with help@stsci.edu as the central contact.

The related web pages for slitless spectroscopy are also now hosted at:
<http://axe.stsci.edu>.

Chapter 1

Description

1.1 What is aXe?

The aXe software was designed to extract spectra in a consistent manner from all the slitless spectroscopy modes provided by the **Wide Field Camera 3 (WFC3)** and the **Advanced Camera for Surveys (ACS)**, which were installed on the Hubble Space Telescope in May 2009 and February 2002, respectively. What we refer to as aXe is in fact a PyRAF/IRAF package with several tasks that can be used to produce extracted spectra. There exist two classes of aXe tasks (see Figure 1.1):

1. The Low-Level Tasks work on individual grism images. All input and output refers to a particular grism image.
2. The High-Level Tasks work on data sets. Their aim is to perform a series of processing steps for a set of images.

The High-Level Tasks often use the Low-Level Tasks to do a certain reduction step on each frame (see Figure 1.1).

The High-Level Tasks were designed to cover all steps of the aXe reduction without any restriction in functionality. All tasks are controlled through a set of configuration files , which can be edited by the user and optimized for a given data set.

The core of the software package is written using ANSI C and **Python** and is highly portable from one platform to another. aXe uses the third-party libraries CFITSIO, GSL, and WCSLIB, which have been successfully employed under Linux, Solaris, and MacOS X.

aXe is distributed as part of the STSDAS software package. Within STSDAS, aXe is located under the subpackages 'analysis.slitless.axe'.

aXe has been used successfully in several large science programs, such as GRAPES (ACS/WFC, [14]) and PEARS (ACS/WFC, [15]). The aXe software was central in extracting ACS/G800L and, using a customized version of aXe, NICMOS/G141 data within the corresponding Hubble Legacy Archive (HLA) projects (see [7] and [2]).

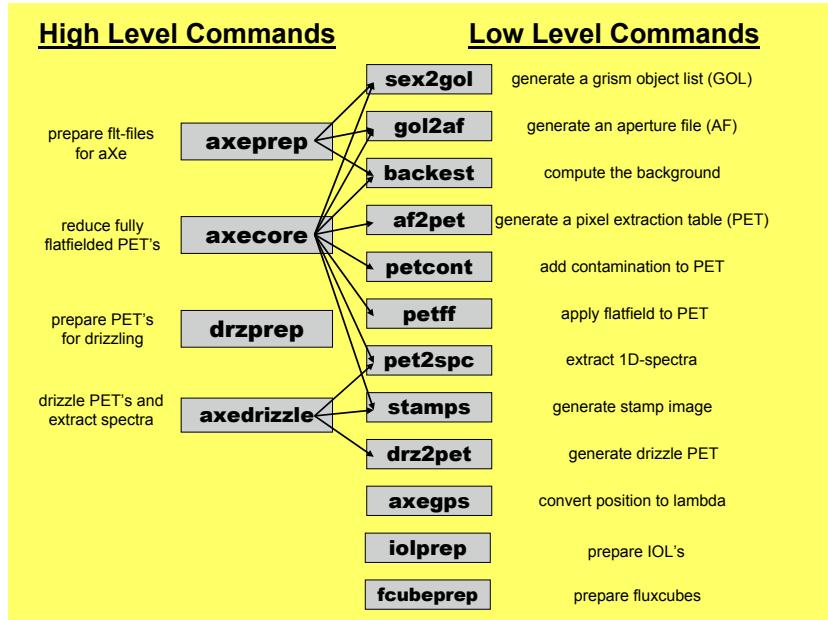


Figure 1.1: The list of aXe tasks. The arrows indicate the interaction between the High Level and the Low Level Tasks

1.2 Slitless Spectroscopy

In conventional spectroscopy, slits or masks are used to allow only the light from a small portion of the focal plane of the telescope to enter the dispersing device (e.g. grism, grating or prism). This results in an unambiguous conversion between pixel coordinates on the detector and wavelength.

In slitless spectroscopy there is no unique correspondence between pixel coordinates and wavelength. Consequently, the spectral reduction on the basis of the spectroscopic data alone is very difficult. Additional information concerning the positions of the object must be added to facilitate the spectral reduction. In aXe this is done by providing an Input Object List (IOL) at the beginning of the reduction process. In the Input Object List the object positions are given in the image-coordinate system or the world coordinate system. This allows the determination of the so called *reference pixel* for every object. The reference pixel is the undispersed object position in image coordinates on the grism data. For each individual object, it is then possible to assign a wavelength to each pixel.

In conventional spectroscopy the extraction of the 1D spectra from the 2D data is done along the direction of the slit or mask. In slitless spectroscopy, such a predefined extraction direction does not exist. It is in fact possible to define a different extraction direction for each object individually by adjusting the wavelength assignment to be constant along the chosen extraction direction (see Fig. 1.2). In aXe the default action is to set the extraction direction to be parallel to the object position angle as given in the Input Object List.

The absence of slits and masks also dramatically enhances the probability that spectra of different sources overlap each other. Even at large distances along

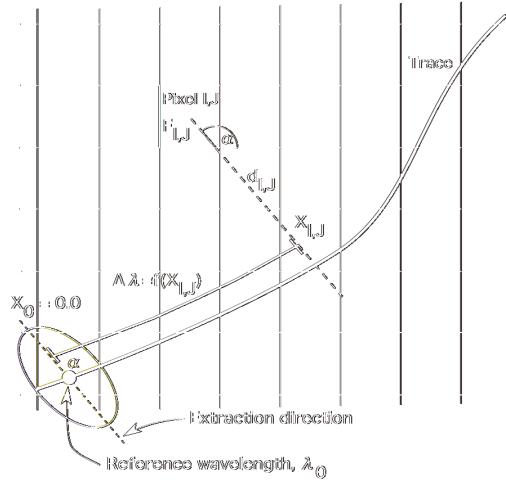


Figure 1.2: The geometry in a beam. A single wavelength is assigned to each pixel within a beam. An extraction along an arbitrary direction α with respect to the pixel grid is allowed. This optimizes the resolution of the final extracted spectrum. In this figure we show α as the orientation of an extended object.

the dispersion direction, the different orders of two objects still can overlap and create confusion problems. aXe can mark and extract several dispersion orders per object to properly record the source confusion or contamination for every order of each object.

1.3 Apertures and Beams

The extraction process in aXe is done on the basis of so called *BEAMs*. Each beam comprises one dispersion order of one object. The collection of all beams (dispersion orders) of one object is called the *APERTURE*. The aperture is characterized by the aperture number, which is identical to the object number in the Input Object List. The beams are named with a single character. The alphabetical sequence (**A**, **B**, **C**, ...) follows the sequence of beams defined in the Configuration File. Each beam is defined by the coordinates of the quadrangle which contains the pixels that are extracted together to form the spectrum of one dispersion order of an object.

The beams follow the spectral trace of the spectrum, which is defined in the Configuration File (see Chapt. 5.2 for a detailed description). While the length of a beam is set by the length of the corresponding dispersion order, its width is adapted to the extraction width set by the user.

The left panel in Figure 1.3 shows an HRC/G800L grism exposure reduced in the HUDF HRC Parallels Program (cleaned from cosmic-ray hits). In the right panel, some of the beams marked and extracted in aXe are indicated. The numbers give the spectral order, and the letters denote the correspondence with the beam in the configuration file. The bright areas mark regions where beams overlap and contaminate their spectra mutually. The different extraction angles for the objects result in different shapes of the marked regions. For each beam

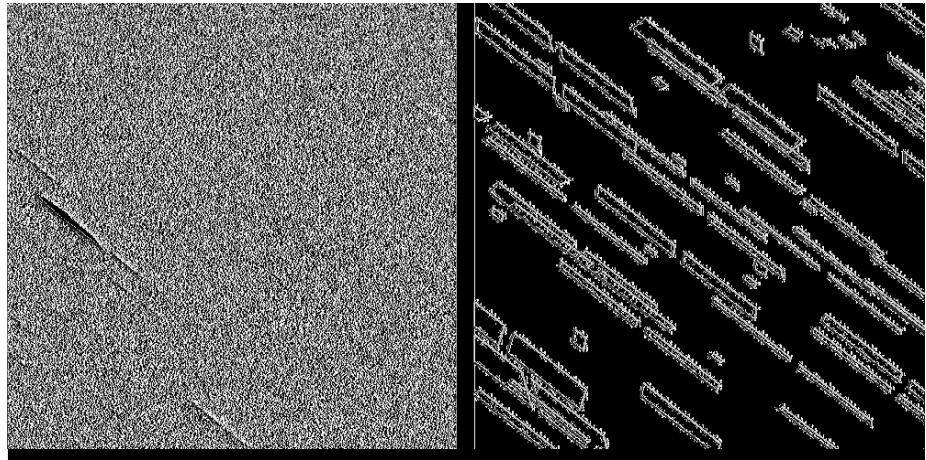


Figure 1.3: A grism image of the HUDF HRC Parallels Program (left panel) and the aXe-beams therein (right panel). The numbers and characters give the spectral order and the beam label in aXe. The bright areas mark the overlap of several beams.

the description of the spectral trace and the wavelength assignment is set up and the spectral reduction is done independently.

1.4 Pixel Extraction Tables (PET)

An important step in the aXe reduction process is the generation of the so called *Pixel Extraction Table* (PET). A PET is a multi-extension fits-table which stores in each extension the complete spectral description of all pixels of one beam. Figure 1.2 illustrates the geometry in a beam and shows various quantities stored in the PET. Important pixel information stored in the PET is:

- the *section point*, defined as the point where the spectral trace intersects a line drawn through the centre of the pixel along the extraction direction
- the distance to the section point d_{ij}
- the trace distance $X_{i,j}$ of the pixel (which is equal to the trace distance of the section point)
- the wavelength attributed to the pixel (derived by inserting the trace distance into the dispersion function stored in the configuration file)

The PETs are read and manipulated by many aXe tasks. For example, a flat-field correction is applied to the pixel values stored in the PETs. Since flat-fielding is a wavelength dependent operation, the assignment of a wavelength to each pixel is required before the correction values, derived from a 3D flatfield cube (see Chapt. 6.2), are applied.

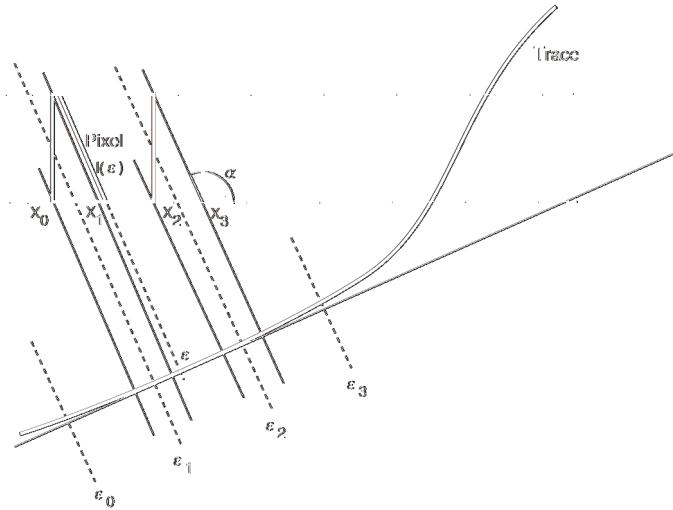


Figure 1.4: How a one-dimensional spectrum is created using information from the Pixel Extraction Table (see Chapt. 7.8). Each pixel in the table is projected onto the trace into separate wavelength bins. The number count of each pixel is weighted by the fractional area of that pixel which falls onto a particular bin.

1.5 Generating 1D spectra

The geometry required to convert the contents of the PET to a set of one dimensional spectra stored in the Extracted Spectra File (SPC) (see Chapt. 7.11) is shown in Figure 1.4. The method accounts for the geometrical rotation of the square pixel with respect to the spectral trace and appropriately projects each pixel onto the trace. To do this a weighting function is used which is the fractional area of the pixel which, when projected onto the trace, falls within the bin points ϵ_1 and ϵ_2 . The flux contained in each BEAM pixel is weighted by this weighting function as it is projected onto separate bins (ϵ_0 to ϵ_1 , ϵ_1 to ϵ_2 , and ϵ_2 to ϵ_3 in Fig. 1.4) along the spectral trace. The weight is computed by integrating over the length of the segments such as $l(\epsilon)$ shown in Figure 1.4. The length of these segments is nonzero from x_0 to x_3 , reaches a maximum value of $1/\sin(\alpha)$, and rises and decreases linearly such that it can be described by:

$$l(x) = \begin{cases} m(x - x_0) & \text{if } x_0 \leq x \leq x_1 \\ l_{max} & \text{if } x_1 \leq x \leq x_2 \\ m(x_3 - x) & \text{if } x_2 \leq x \leq x_3 \\ 0 & \text{otherwise} \end{cases}$$

where $m = l_{max}/(x_1 - x_0)$

Integration over this function $l(x)$ to compute $w(\epsilon_0, \epsilon_1)$, $w(\epsilon_1, \epsilon_2)$, and $w(\epsilon_2, \epsilon_3)$ is trivial once x_0, \dots, x_3 have been computed, which are derived from simple trigonometry.

Once the one dimensional spectra have been generated, the final step of flux calibrating can be performed by applying a known sensitivity curve for the observing mode which was used. The output product of the aXe extraction

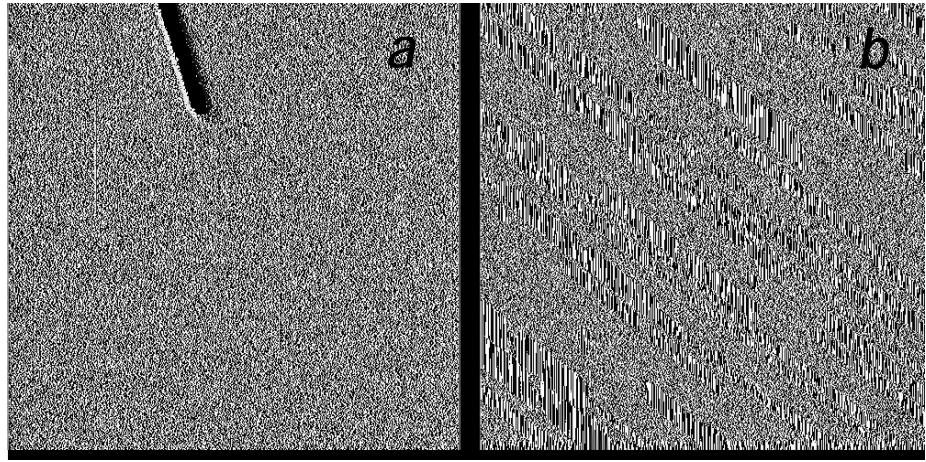


Figure 1.5: Left Panel (a):The master sky for the HRC. The holder for the coronograph results in the ‘arm’ with low sensitivity at the top. Right Panel (b): Background estimate for the grism image shown in Fig. 1.3 with the object regions masked.

process is a FITS binary table containing the set of extracted and calibrated spectra (Extracted Spectra File, see Chapt. 7.11).

1.6 Sky Background

aXe has two different strategies for removal of the sky background from the spectra.

The first strategy is to perform a global subtraction of a scaled “master-sky” frame from each input spectrum image at the beginning of the reduction process. This removes the background signature from the images, so that the remaining signal can be assumed to originate from the sources only and is extracted without further background correction in the aXe reduction.

The second strategy is to make a local estimate of the sky background for each BEAM by interpolating between the adjacent pixels on either side of the BEAM. In this case, an individual sky estimate is made for every BEAM in each science image.

The strategy of estimating a local sky background can also be applied after a global sky background subtraction for very difficult cases or instruments (e.g. NICMOS G141 data, see [2]).

1.6.1 Global Background Subtraction

The homogeneous background of HST grism exposures makes the global background subtraction from the pipeline processed science images (i.e. `.flt.fits` files) feasible. Master sky images for both ACS (WFC and HRC) and WFC3 are available from the instrument web pages at <http://www.stsci.edu/hst/acs/analysis/STECF> and http://www.stsci.edu/hst/wfc3/analysis/grism_obs/, respectively.

All master sky images were created by combining many grism images from different science programs. The object signatures on the science images were removed using several techniques, including a two step median combination, to derive a high signal-to-noise image of the sky background. Figure 1.5a shows the ACS/HRC master sky image.

Scaling and subtraction of the master sky is done with the aXe task `axeprep` (see Fig. 1.1). Before scaling the master sky to the level of each science frame, the object spectra are masked out on both the science and the master sky image.

When reducing a dataset consisting of many individual exposures, it may be desirable to check the sky subtraction by co-adding all the sky-subtracted grism images (e.g. with the `MultiDrizzle` task). The co-added image also provides a way to quickly assess the quality of the background subtraction. Any deviations from zero in the mean background level of the combined image will also affect the spectra derived with the aXe reduction.

1.6.2 Local Background Subtraction

The second option for handling the sky background is to make a local estimate of the background for each object. In this case, aXe creates an individual background image for each object on the spectrum image. On the background image the pixel values at the positions of the object beams are derived by interpolating in each column between the pixel values on both sides of the beam. The number of pixels used in the interpolation as well as the degree of the interpolating polynomial can be chosen by the user. Figure 1.5b shows the background image corresponding to the grism image displayed in Fig. 1.3.

The background images are then processed in much the same way as the science images, resulting in a *Background Pixel Extraction Table* (BPET) for all BEAMs in a grism image. Thus, every PET has its corresponding BPET, derived from the background image, with the spectral information of the identical objects and beams in it. Finally, the BPET is subtracted from the PET and the background subtracted spectra are extracted.

1.7 Contamination

In conventional spectroscopy an overlay of spectra from different sources can occur only if two or more objects fall within the aperture defined by a slit or mask element. However in slitless spectroscopy there is no spatial filtering of sources. This allows both overlap of spectra from near neighbours in the cross dispersion direction as well as from more distant sources in the dispersion direction. For this reason spectral overlap or contamination is an ubiquitous issue for slitless spectroscopy, which must be explicitly taken into account in the data reduction.

1.7.1 Geometrical contamination

In *geometrical contamination* the areas covered by the different orders of all objects are recorded on a so-called contamination image. Figures 1.3 and 1.6 show the contamination image for data taken in the Hubble Ultra Deep Field with the HRC and WFC, respectively. In both figures the regions marked black

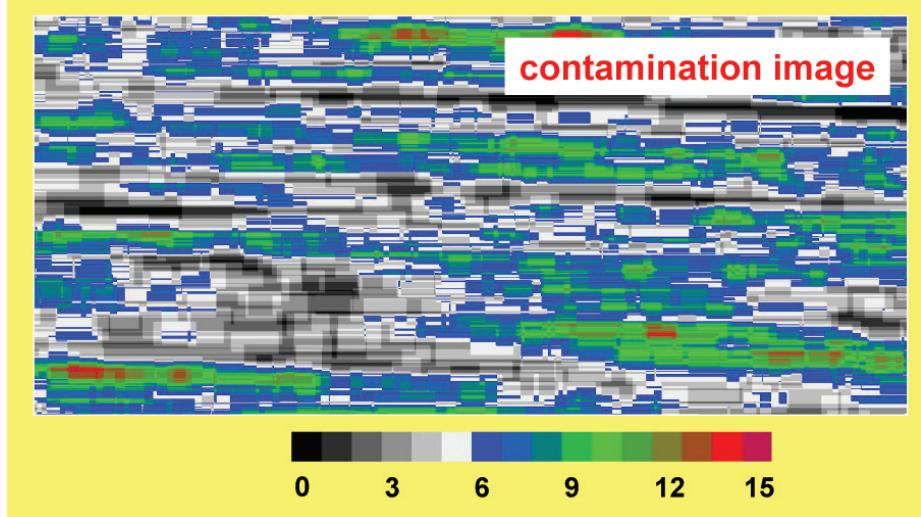


Figure 1.6: The contamination image compiled for data taken in the Hubble Ultra Deep. The different colours give the number of spectral orders which contaminate each other.

are covered by no spectrum at all, the white or red areas show regions which are covered by several (up to 15 in Fig. 1.6) overlapping spectra, which contaminate each other.

The information on the number of contaminating sources in Fig. 1.6 is stored in the object PET and fully propagated in the 1D extraction of the individual object spectra. As a final result each spectral element is accompanied by a flag which indicates whether its input pixels were also part of other object spectra. The regions of 1D spectra where the contamination flag is set must be used with care, since neighbouring sources also contribute to the extracted flux.

This contamination scheme is fast and very efficient in identifying problematic regions in the individual object spectra, however there is no information on the severity of the contamination.

The quantitative contamination introduced below assesses the contamination from neighbouring sources and helps to decide whether the contaminated spectrum might still be suitable for further scientific analysis.

1.7.2 Quantitative contamination

The *quantitative contamination* gives, for each spectral element, an estimate on the contaminating flux from all other sources. Based on this quantitative contamination estimation, the user has a better tool to decide which data points can be trusted.

The basis of the quantitative contamination estimation is a model which estimates the dispersed contribution of every object to the grism image. The contributions of the individual objects are then coadded to a 2D contamination image, which is a quantitative model of the examined grism image. In the 1D extraction of the individual object spectra, the model contribution of the object itself is subtracted (to avoid self-contamination), and then the data from the

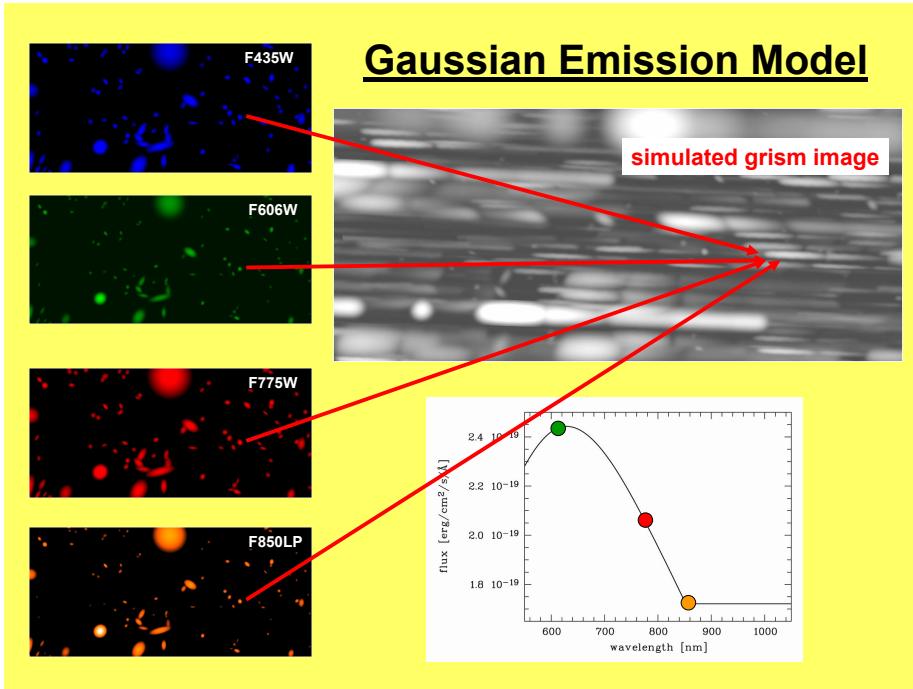


Figure 1.7: The Gaussian emission model for ACS/WFC: photometric information in the four filters shown on the left is employed to compute the model grism image (right). The object morphologies are approximated by 2D Gaussians. The arrows connect the direct image positions of one object to its first order grism spectrum. The photometric values are transformed to flux and interpolated as shown on the lower right.

modelled grism image is processed in parallel with the data from the real grism image.

As a result two spectra for every object are derived: one extracted from the real grism image; and a second one extracted from the modelled grism image. Since the model contribution of the object itself was excluded in the extraction of the latter spectrum, this spectrum is a quantitative estimate of the contamination from all other sources to the object spectrum in question. The accuracy of the contamination spectrum is set by the accuracy of the *emission model* which is needed as an input to compute the modelled grism image.

Two different emission models have been implemented, called the *Gaussian Emission Model* and the *Fluxcube Model*

The Gaussian Emission Model In the Gaussian emission model, the object morphologies are approximated by Gaussians with widths taken from the Input Object List. The Input Object List must also contain photometric information, which is provided by the total AB-magnitude in at least one filter passband or wavelength. In this mode the column name of the magnitude columns must indicate the wavelength with a simple format (such as MAG_F850LP for an AB-magnitude determined at 850nm, see Chapt. 7.4 for details on the column

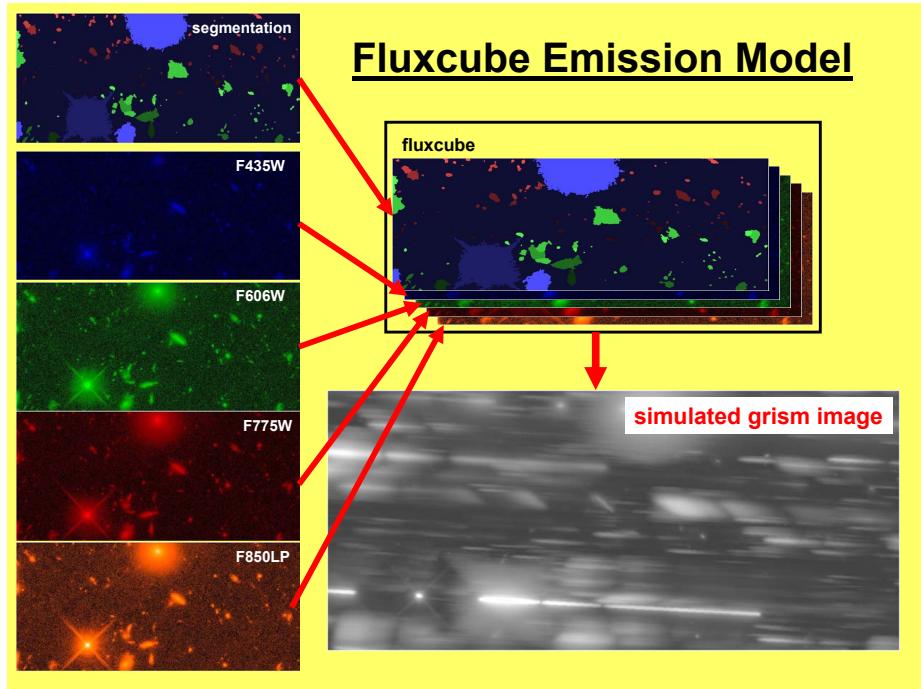


Figure 1.8: The Fluxcube emission model for ACS/WFC: real images in four filters (left) are converted to flux images and combined with the segmentation image to a fluxcube file (upper right). The model grism image (lower right) is computed using the data in the fluxcube.

names). With a proper name for the magnitude column, the Input Object List, which is required to run aXe, contains all the data to compute the contamination with the Gaussian emission model.

Figure 1.7 displays on the left side the ACS/WFC direct images in the four observed filters as seen in the Gaussian emission model, which means all objects have Gaussian shapes. The upper right panel of Fig. 1.7 shows the modelled ACS/WFC/G800L grism image computed from the morphological and photometric information. The arrows point from the direct image positions of one object to the position of its first order spectrum in the modelled grism image. In order to compute the contribution of this object to the grism image, the four photometric values (AB-magnitudes at 435, 606, 775 and 850 nm) were transformed to flux and then interpolated with a cubic spline as shown in the lower right panel of Fig. 1.7. Outside the range of the photometric data a constant extrapolation of the last available data point is used.

The images in Fig. 1.7 cover the same area as the contamination image in Fig. 1.6. The direct images in Fig. 1.7 were only created for illustration purposes. In real aXe runs, each filter is just represented by a column in the Input Object List which gives the total AB-magnitude of the objects.

The Fluxcube Model In the Fluxcube emission model both the object morphologies as well as the spectral information are taken from the fluxcube file

associated with every grism image. A fluxcube file is a multi-dimensional fits image with one or several flux images taken at different wavelengths as extensions. The basis of the flux images are normal 2D images in [*counts/sec*], which must be transformed to flux in [*erg/cm²/s/Å*] using the appropriate zeropoints. All extensions of the fluxcube image must cover the same area as the corresponding grism image.

The flux extensions in the fluxcube provide sufficient information to compute a model grism image. In the determination of the quantitative contamination however it is essential to derive the individual contribution of each object to the modelled grism image. This addition is necessary to be able to subtract the self contamination and to isolate the contamination from other sources for each individual object.

For this reason the first extension of a fluxcube image must contain the so called "Segmentation Image". In the segmentation image each pixel value is the (integer) number of the object to which the pixel is attributed. The SExtractor software provides the possibility to create a segmentation image (parameter setting: `CHECKIMAGE_TYPE SEGMENTATION`) as an additional output product of the source extraction.

The fluxcube files necessarily follow a rather complicated file format. To support the user in the creation of fluxcube files an aXe task has been implemented. The task `fcubeprep` works in a standard scenario with a multidrizzled grism image, one or several multidrizzled direct images and a segmentation image as input.

As an illustration of the Fluxcube model for ACS/WFC, Figure 1.8 shows on the left side the segmentation image and the filter images used to create the fluxcube. The lower right part of Fig. 1.8 displays the modelled ACS/WFC/G800L grism image derived by the fluxcube emission model. All images in Fig. 1.8 cover the identical area of Figs. 1.6 and 1.7 in the HUDF.

More details on quantitative contamination are given in [6] and [8].

1.8 Drizzling of PETs

The aXe reduction scheme described up to now produces one spectrum for each individual beam in each science image. However, datasets, such as those obtained with ACS, often consist of several images with small position shifts (dithers) between them. The direct approach of co-adding the 1D spectra extracted from each image to form a combined, deep spectrum has several disadvantages:

- The data is (non-linearly) rebinned twice, once when extracting the spectrum from the image and again when combining the individual 1D spectra;
- A complex weighting scheme is required to flag cosmic ray affected and bad pixels;
- Low level information on the cross dispersion profile is lost when many 1D extracted spectra are combined to a deep spectrum.

To circumvent these drawbacks, there is a more advanced reduction scheme available, whereby all the individual 2D spectra of an object are coadded to a

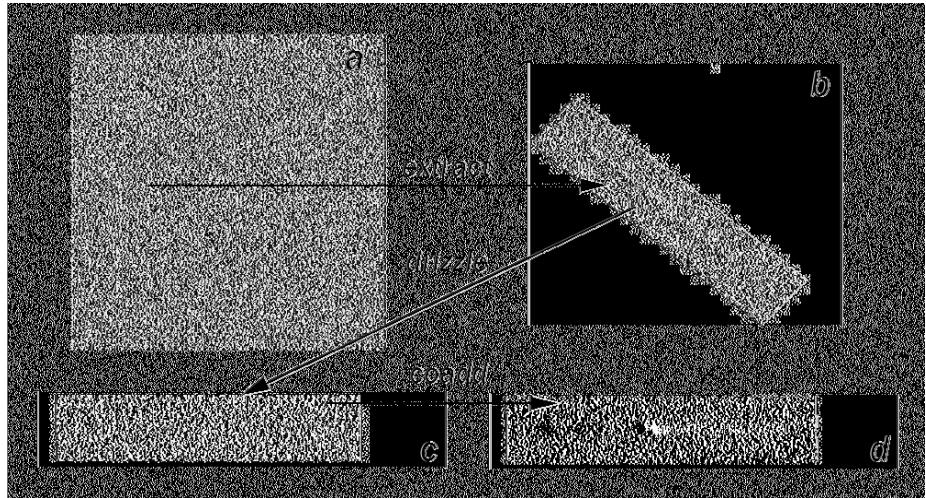


Figure 1.9: Drizzling in aXe: The object marked in panel (a) is extracted as a stamp image (b). The stamp image is drizzled to an image with constant dispersion and constant pixel scale in cross dispersion direction (c). The deep 2D drizzled image (d) is then used to extract the 1D spectrum.

single deep 2D spectrum. The final, deep 1D spectrum is then extracted from this combined 2D spectral image. The combination of the individual 2D spectra is done with the “Drizzle” software, Fruchter & Hook (2002), which is available in the STSDAS package within IRAF.

The advantages of this technique as applied to slitless spectra can be summarised as follows:

- Regridding to a uniform wavelength scale and a cross-dispersion direction orthogonal to the dispersion direction is achieved in a single step;
- Weighting of different exposure times per pixel and cosmic-ray affected pixels are correctly handled;
- There is only one linear rebinning step to produce a 2D spectrum;
- The combined 2D spectra can be viewed to detect any problems.

These advantages come at the expense of a greater complexity of the reduction and significantly longer processing time. Also, the aXe drizzle reduction currently supports only first-order spectra.

The drizzling within aXe is fully embedded in the aXe reduction flow and uses data products and tasks created and used in the non-drizzling part of aXe. The input for the drizzle combination consists of flatfielded and wavelength calibrated PETs extracted for each science image, which are converted to *Drizzle PrePare files* (DPP) using the `drzprep` task. Every first order beam in a PET is converted to a stamp image stored as an extension in a DPP. The `drzprep` task also computes the transformation coefficients which are required to drizzle the single stamp images of each object onto a single deep, combined 2D spectral image. These transformation coefficients are computed such that the combined

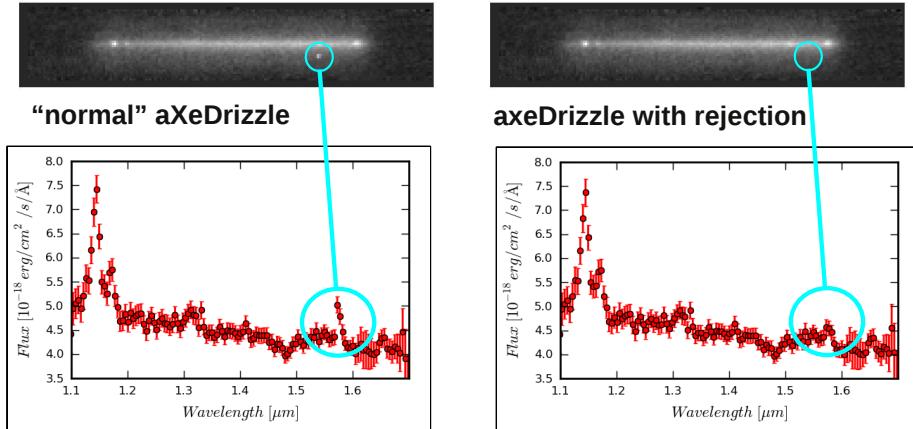


Figure 1.10: The comparison of a 2D drizzled image (WFC3/G141) produced with the “normal” aXedrizzle (left column) and the new aXedrizzle that can reject deviant pixels (right column). Hot or cosmic ray affected pixels without a correct flag in the data quality array are detected and masked out in the new aXedrizzle.

drizzle image resembles an ideal long slit spectrum, with the dispersion direction parallel to the x-axis and cross-dispersion direction parallel to the y-axis. The wavelength scale and the pixel scale in the cross-dispersion direction can be set by the user with keyword settings in the aXe Configuration File.

To finally extract the 1D spectrum from the deep 2D spectral image, aXe uses an (automatically created) adapted configuration file that takes into account the modified spectrum of the drizzled images (i.e. orthogonal wavelength and cross-dispersion and the $\text{\AA}/\text{pixel}$ and $\text{arcsec}/\text{pixel}$ scales).

A detailed discussion of the drizzling used in aXe is given in Kümmel et al. (2004a).

Figure 1.9 illustrates the aXedrizzle process for one object. Panel *a* shows one individual grism image with an object marked. Panel *b* displays the stamp image for this object out of the grism image. Panel *c* shows the derived drizzled grism stamp image, and the final coadded 2D spectrum for this object is given in panel *d*. Panel *d* shows an image combined from 112 PETs with a total exposure time of 124 ksec. In both panels *b* and *c*, the ‘holes’ resulting from the discarded cosmic ray-flagged pixels in this individual exposure are clearly visible.

For aXe version 2.1 we have extended aXedrizzle, and the new version offers to detect pixels with deviating values such as MultiDrizzle [5] does in direct imaging. As is shown in Figure 1.10, the new aXedrizzle is able to detect and mask out deviant pixels (right panels), thus reducing the risk to produce an emission line which is an artefact (left panels).

The new aXedrizzle can only be applied if the sky background has been subtracted off via global background subtraction (see Chapt. 1.6.1). This new method of combining the 2D grism stamp images has been developed on the basis of and for WFC3 G102 and G141 data. In principle, the new aXedrizzle could also be applied to ACS G800L data, however there exist other methods

to reliably detect cosmic rays in these modes (see Chapt. 3.2.1). The new aXedrizzle has certainly the potential of delivering better and cleaner spectra, as can be seen in Fig. 1.10. But if the alignment of the images has not been done properly, the new aXedrizzle can massively mask out good pixels and thus damage the resulting spectra. The results of the aXedrizzle with pixel rejection (task `axedrizzle` with `driz_separate='YES'`, see Chapt. 4.6) should only be taken as valid after a carefully comparison with the spectra from the “basic” aXedrizzle process (task `axedrizzle` with `driz_separate='NO'`, see Chapt. 4.6).

1.9 Optimal weighting

The use of unequal weights in the 1D extraction of spectral data can enhance the signal-to-noise ratio of the extracted spectra. The improvement is achieved by attributing lower weights to pixels which, due to the larger distance from the spectral trace, contain only a small fraction of the object flux. The optimal weighting technique was originally introduced in Horne (1986) ([4]), and the basic equation of the spectral extraction using optimal weights is (see e.g. Rodriguez-Pascual et al. 1999, [17]):

$$f(\lambda) = \frac{\sum_x [f(x, \lambda) - b(x, \lambda)] * \frac{p(x, \lambda)}{\sigma(x, \lambda)^2}}{\sum_x \frac{p(x, \lambda)^2}{\sigma(x, \lambda)^2}} \quad (1.1)$$

The variables are:

- λ : the coordinate in the spectral direction
- $f(x, \lambda)$: the data value at pixel (x, λ)
- $b(x, \lambda)$: the background value at pixel (x, λ)
- $\sigma(x, \lambda)$: the noise value at pixel (x, λ)
- $p(x, \lambda)$: the extraction profile at pixel (x, λ)
- $f(\lambda)$: the extracted data value at λ

In the original descriptions of optimal weighting, the extraction profile $p(x, \lambda)$ is computed from the object spectrum itself by e.g. averaging the pixel values in wavelength direction. In Horne (1986, [4]) optimal weighting (or optimal extraction, as named there) is even an iterative procedure which, starting from a normal extraction procedure using equal weights, produces improved results for sky background, extraction profile and, of course, the extracted spectrum.

In ACS slitless spectroscopy such an approach is not viable since

- an iterative approach on the sometimes hundreds or even thousands of spectra on a slitless image would require too much computing time;
- the signal-to-noise ratio of the sources is often too low to determine an individual extraction profile;
- the contamination phenomenon does not permit an automatic and reliable generation of extraction profiles for all sources.

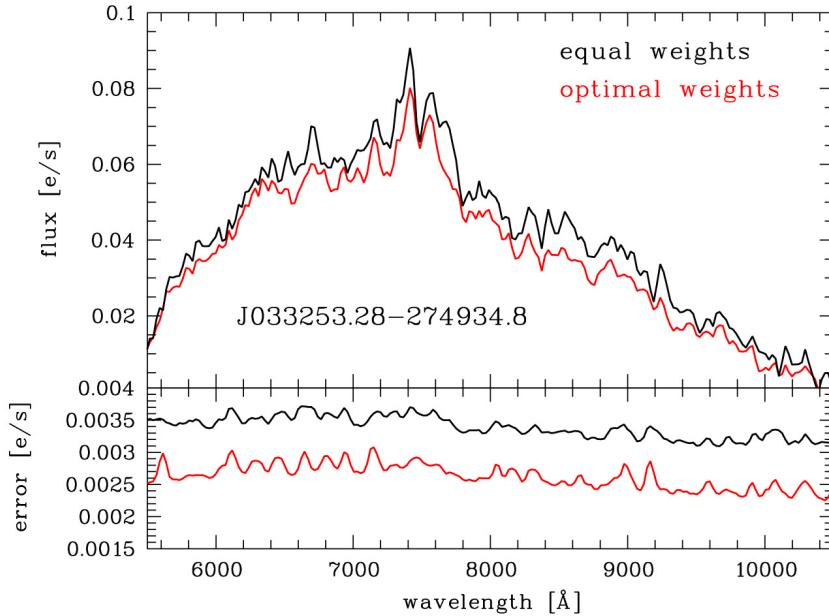


Figure 1.11: The comparison for an aXe extraction with (red) and without (black) optimal weights. The upper panel compares the object flux, the lower panel shows the associated errors.

To compute extraction profiles for all sources, the optimal weighting as implemented in aXe uses the 2D models for the dispersed objects, which were introduced in Chapt. 1.7.2 as the basis of quantitative contamination. The source-specific models computed there deliver a perfect basis to calculate the quantity $p(x, \lambda)$ in Eqn. 1.1.

The beam models are also used as an input to calculate the pixel errors $\sigma(x, \lambda)$ according to the typical CCD noise model

$$\sigma(x, \lambda) = \sqrt{mod(x, \lambda) + b(x, \lambda) + rdnoise^2} \quad (1.2)$$

with $mod(x, \lambda)$ and $rdnoise$ the beam model value at pixel (x, λ) and the CCD readout noise, respectively. Computing the quantitative contamination estimate with either the Gaussian or the Fluxcube emission model is therefore a precondition to optimal weighting.

In all extraction modes (from individual grism images or from the combined 2D drizzled grism images) aXe delivers optimal weighted spectra as an optional addition to the usual, equally weighted ones. Figure 1.11 shows a comparison between two spectra extracted from the same data using equal and optimal weights. Results from both observed as well as simulated data indicate that optimal weighting in aXe improves the signal-to-noise ratio by a small, but significant amount as expected according to Horne (1986) and Robertson (1986) ([4] and [16]).

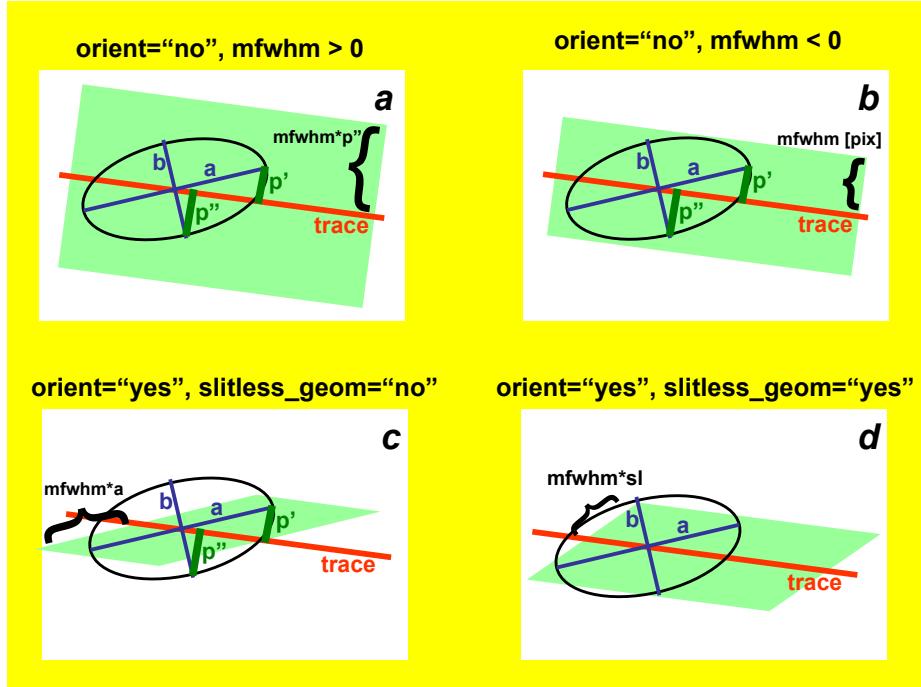


Figure 1.12: The four different methods to extract 1D spectra: (a) perpendicular to the trace, with an object specific extraction width $mfwsm * MAX(p', p'')$ and p' and p'' the projection of the major and minor half axis width onto the extraction direction; (b) perpendicular to the trace, with a fixed extraction width $mfwsm$ for all objects; (c) along the direction of the object's major half axis with extraction width of $mfwsm * a$ (a = major axis width); (d) a virtual slit with length sl , width sw and orientation so is computed from the morphological object parameters (a, b, θ) to extract with the width $mfwsm * sl$ along so .

1.10 Extraction parameters in aXe

aXe offers a large range of possibilities to specify the extraction width and extraction direction for the individual objects. Before running aXe, the user has to decide in which way the 1D spectra should be extracted from the grism images.

Fixed extraction direction or variable extraction direction With a fixed extraction direction the lines of constant wavelength and therefore the extraction direction form the angle 90° with the trace in all beams of all objects.

With variable extraction, the line of constant wavelength follows for every object a specific, marked direction. The major axis angle in the column THETA_IMAGE of the Input Object List is used in this mode to define the line of constant wavelength or extraction direction for every object individually. aXe mimics with the variable extraction direction individually oriented slits for all objects. This can help to maintain the instrumental resolution for small, ex-

Note on extraction parameters:

The optimal choice of extraction strategy depends very much on the scientific goals and the morphology of the observed sources. For stellar sources a fixed extraction direction together with a fixed extraction width is certainly highly recommended. Deep survey type observations definitely need a variable extraction width, since the object sizes usually span a large range which can not be met with a fixed extraction width. In this case the results would also benefit from a variable extraction direction to get the best possible spectral resolution.

Also in typical survey scenarios the morphological description of stellar objects and faint objects close to the detection limit rather reflects the statistical or systematic measurement errors than the true intrinsic object properties. To compensate these doubtful measurements of the quantities `A_IMAGE`, `B_IMAGE` and `THETA_IMAGE`, aXe applies, with the parameters optimized for surveys, a default extraction with an extraction direction perpendicular to the trace angle and a fixed object size to all objects smaller than a threshold given in the configuration file. Setting this threshold to the typical size of point-like objects assures a proper extraction for stellar objects and marginally resolved faint sources.

tended objects. However for small angles between the trace and the extraction direction the finite instrumental resolution limits any improvements due to the variable extraction direction, and in addition the extraction becomes numerically unstable¹. aXe can switch (with the parameter `SLITLESS_GEOM='YES'`, see below) the extraction direction from the major axis angle to a different angle which optimizes the resolution of the extracted spectra ([6], [2]).

Fixed extraction width or object specific extraction width Similar to the choice of extraction direction, aXe offers both a fixed and a variable extraction width. The fixed extraction width remains constant for all objects.

The variable extraction width is determined for each object individually to a scaled value $extrfwhm$ of the object extent in the extraction direction.

The main parameters to specify extraction width and extraction direction are `extrfwhm` (or `mfwhm` in Fig. 1.12), `orient` and `slitless_geom` in the task `axecore`. Figure 1.12 illustrates how those parameters can be used to extract the flux of an object in various ways:

- in Fig. 1.12a, the flag `orient='NO'` indicates a fixed extraction direction of 90° with respect to the trace direction. A value $mfwhm > 0$ specifies a variable extraction width, which in the case `orient='NO'` is $width = MAX(p', p'') * mfwhm$ pixels on either side of the trace (hence $2 * width$ in total). Here p' and p'' are the projection of the major and the minor axis width onto the extraction direction.

¹The parameter combination `ORIENT='YES'` and `SLITLESS_GEOM='NO'` might be reasonable in isolated cases, and its use is not prohibited. However aXe delivers a warning in case that the angle between the extraction direction and the object trace is very small ($< 1^\circ$). This warning must be seriously taken into account, since a core dump may result during later stages of the aXe reduction

- in Fig. 1.12b, the flag `orient='NO'` indicates a fixed extraction direction of 90° with respect to the trace direction. A value `mfw hm < 0` specifies a fixed extraction of $width = mfw hm$ pixels on each side of the trace (hence $2 * mfw hm$ pix in total).
- in Fig. 1.12c, the flag `orient='YES'` indicates a variable extraction direction. Since `slitless_geom='NO'` the extraction direction must follow the direction of the major axis a . The extraction width is variable with $width = mfw hm * a$ pixels on either side of the trace.
- in Fig. 1.12d, with the flag `slitless_geom='YES'` an individual *virtual slit* with the slit length sl , the slit width sw and the orientation so is defined from the major axis size `A_IMAGE`, the minor axis size `B_IMAGE` and the major axis angle `THETA_IMAGE` given in the Input Object List. Appendix A gives the equations for computing the virtual slit parameters. The shape of the virtual slit optimizes the spectral resolution for the extracted spectra and avoids angles too close with the trace direction (see [6] and [2] for details). The extraction width is then the object specific width $width = mfw hm * sl$ along the direction so .

1.11 Flux conversion

The flux conversion is done using sensitivity curves which had been derived through dedicated observations of flux standard stars ([11], [12], [13]). In extended objects, however, the spectral resolution is degraded by the object size in the dispersion direction. aXe can take into account (parameter `adj_sens='YES'` in the tasks `axecore`, `axedrizzle`, `pet2spc`) the degraded spectral resolution of extended sources by smoothing the point source sensitivity function. Based on the approximation of Gaussian object shapes, aXe uses a Gaussian smoothing kernel with the width

$$\sigma_i = f * r * \sqrt{sw_i^2 - p^2}$$

with sw_i the width of the *virtual slit* of object i (see Appendix A), the dispersion r , the point source object width p and the correction factor f , which is empirically determined for the various slitless modes. The adjusted flux conversion has been developed for and applied in the various data reduction projects within the **Hubble Legacy Archive** (HLA) program [2].

Figure 1.13 shows the effect of the sensitivity adjustment for an extracted ACS/WFC spectrum. The lower panel shows a strong upturn at both wavelength ends due to the degraded resolution. Smoothing the sensitivity function using the appropriate Gaussian kernel suppresses this effect (upper panel).

1.12 aXe Visualization

A deep ACS/WFC grism image can contain detectable spectra of hundreds to thousands of objects, and visual checking of each spectrum is very tedious. A quick-look facility is highly desirable in order to find interesting objects (e.g. high redshift galaxies, SN, etc) which can be highlighted for further study or interactive spectrum extraction. For this reason `aXe2web` was developed, a tool

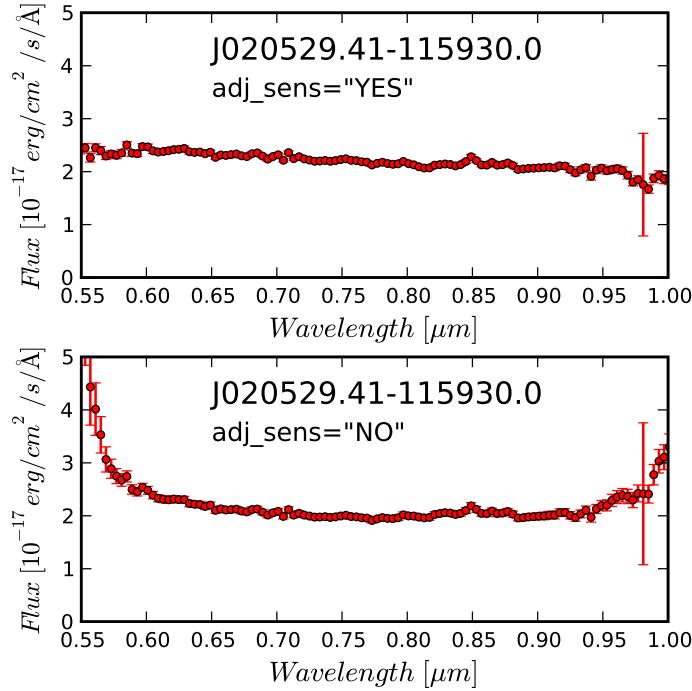


Figure 1.13: An ACS/WFC G800L slitless spectrum of an extended object reduced with (upper panel) and without (lower panel) adjusting the sensitivity curve in the flux conversion. The ‘wings’ at both wavelength ends in the lower panel are a clear sign of the decreased resolution due to the object extent.

which produces browsable web pages for fast and discerning examination of many hundreds of spectra.

Since `aXe2web` requires specific python modules, it cannot be included in the STSDAS software package. It is therefore distributed via the `aXe` webpage at <http://axe.stsci.edu/axe/installation.html> in the `aXe2html` package.

`aXe2web` uses a standard `aXe` input catalogue and the `aXe` output files to produce an html summary containing a variety of information for each spectrum. This includes a reference number, magnitude in the magnitude system of the direct object, the X and Y position of the direct object, its Right Ascension and Declination, a cut-out image showing the direct object, the spectrum stamp image showing the 2D spectrum, a 1D extracted spectrum in counts and the same in flux units.

The user can set various keywords to influence the html output. For example, it is possible to sort the objects with respect to an object property such as magnitude or Right Ascension.

In order to facilitate the navigation within a data set, an overview and an index page accompany the object pages. The overview page contains for each object the basic information sequence number, reference number, X,Y,RA,Dec and magnitude. The index page includes a table with the ordered reference number of all objects. Direct links, from both the overview page and the index

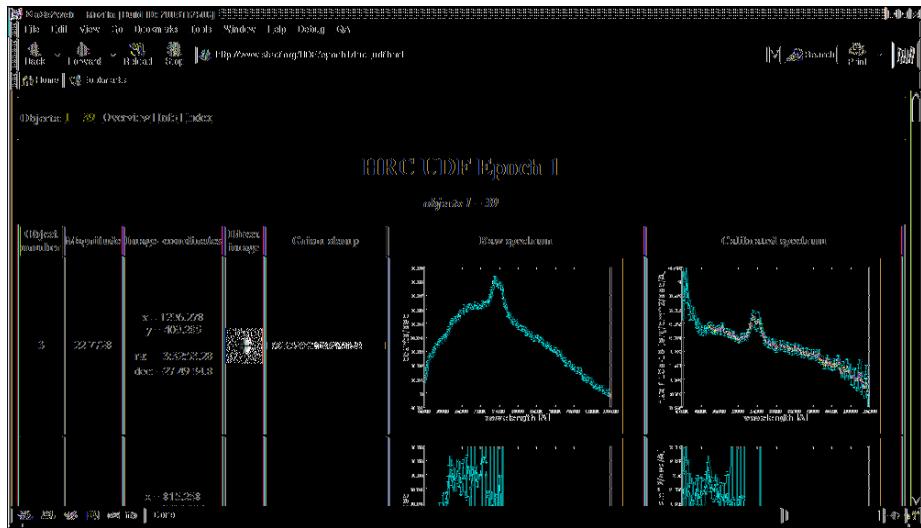


Figure 1.14: Part of a webpage created by aXe2web. The coadded 2D spectrum of the object shown here is displayed in Fig. 1.9d.

page, point to the corresponding locations of the objects in the object pages.

Figure 1.14 is a screenshot taken from Epoch 1 data of the HUDF HRC Parallels survey and shows the line covering the object whose coadded 2D spectrum is shown in Fig. 1.9d. The web pages created by aXe2web are located at the preview web pages: http://archive.stsci.edu/prepds/udf/stecf_udf/epoch1/hrc_udf.html.

1.13 Acknowledgement

In publications, please refer to aXe as:

Kümmel, M., Walsh, J.R., Pirzkal, N., Kuntschner, H. & Pasquali, A., 2009, PASP 121, 59

Chapter 2

Installing aXe

2.1 Package Structure

The aXe software is composed of a combination of routines written in ANSI-C and python. Many of the python modules use the C executables to do their work, while some perform all operations within the python module itself. Within the IRAF/STSDAS environment, the C executables reside in the `stsdas$bin` directory, while the python source routines reside in the STSDAS analysis/slitless/axe source tree.

2.2 Requirements

The following software packages are required to run aXe:

- STSDAS 3.11 or later (http://www.stsci.edu/resources/software_hardware/stsdas)
- PyRAF 1.8 or later (http://www.stsci.edu/resources/software_hardware/pyraf)

2.3 aXe Distribution

aXe is distributed as part of the STSDAS software package. The latest release of STSDAS, version 3.14 (released in July 2011), contains aXe version 2.3. You can download the latest version of STSDAS at
http://www.stsci.edu/resources/software_hardware/stsdas. If you use a computer platform that is supported by STScI, you can simply download and install one of the binary distributions.

2.4 Installing an aXe source distribution

The aXe C source modules are not distributed with STSDAS. If you use a computer platform for which STScI does not supply a binary distributions of STSDAS, please send email to `help@stsci.edu` to request a package of the aXe C modules and instructions for compiling and installing the binaries within STSDAS.

2.5 Validating the aXe installation

Test data with WFC3 and ACS grism images, as well ACS prism images, can be obtained from the aXe web site at <http://axe.stsci.edu/axe/testdata.html>. Unzip and untar the test data file in a clean directory and follow the instructions given in the README file. The ACS grism test data consist of a set of science frames taken from the HUDF HRC Parallels program. Figs. 1.3, 1.5 and 1.9 show some of the data products generated during the test reduction.

The prism test data was taken as part of the calibration proposal 10391 (PI: S.S. Larsen).

The WFC3 test data originates from the WFC3 Early Release Science program (PID: 11359, PI: O'Connell)

Reference spectra generated by running aXe on the test data are also supplied as part of the test packages. If the output obtained by running aXe on the test data is identical to these reference spectra, the proper working of aXe is assured.

2.6 aXe Support

The aXe software package was originally developed by a dedicated group at the Space Telescope - European Coordinating Facility (STECF). STECF supported the use of the aXe software and the slitless spectroscopic modes of ACS and WFC3 until late 2010. Since the beginning of 2011 support is being provided by STScI via email to help@stsci.edu.

Chapter 3

Using aXe

This chapter gives a step-by-step approach on how an aXe reduction on a given data set is performed. A short introduction to the input data is followed by an explanation of the few preparatory steps that are necessary to generate all input files. Then the different methods to produce the final, calibrated 1D spectra are shown and discussed. Both the sequence of commands within PyRAF as well as small example scripts are given for all reduction branches.

3.1 The input data

The input data consist of four direct images and eight grism images. All images were taken with the High Resolution Camera of the ACS. The direct images taken with the F555W filter are

```
j8m81cd9q_flt.fits  
j8m824toq_flt.fits  
j8m84aqkq_flt.fits  
j8m851tmq_flt.fits
```

The dispersed images observed with the G800L grism are:

```
j8m8201eq_flt.fits  
j8m8201lq_flt.fits  
j8m8201rq_flt.fits  
j8m820m4q_flt.fits  
j8m822q0q_flt.fits  
j8m822q4q_flt.fits  
j8m822qbq_flt.fits  
j8m822qhq_flt.fits
```

The data set presented here is in the HST archive. It has been taken as part of the ACS/HRC Parallels program to the ACS Ultra Deep Field. Moreover these images are also part of the test data for grism observations (see Chapt. [2.5](#)).

Spectral extraction from MultiDrizzled grism/prism images While `MultiDrizzle` can combine slitless grism/prism images, no wavelength sensitive flatfield is applied to the images (see Chapt. 6.2). Moreover, the field dependence of the grism/prism spectra and the field dependent wavelength calibration are not taken into account in the combination process. It is therefore not recommended to extract the spectra directly from the `MultiDrizzle` combined grism/prism images. The images are however very useful for field examination.

3.2 Preparing the extraction

This reduction step collects all necessary input files for the spectral extraction. In this process the ‘external’ programs SExtractor (Bertin & Arnouts 1996) and `MultiDrizzle` (part of the `stsda.dither` package) are involved. For those programs outside of aXe we do not give a detailed discussion on their usage or the exact parameter settings, but rather a description of the purpose and what the program should deliver.

3.2.1 MultiDrizzle

The first step is to run `MultiDrizzle` on both the set of direct images and the set of grism/prism images. `MultiDrizzle` is an interface for performing all the tasks necessary for registering dithered HST images. The program automatically performs cosmic ray rejection, removes geometric distortions and performs the final image combination with “drizzle”.

The grism/prism image combination is done for two reasons:

1. the combined image gives a good impression on the quality of the data and the signal-to-noise level of the various object spectra
2. `MultiDrizzle` runs a cosmic ray detection algorithm, and the dq-extenton of the flt-images is updated with the information on all cosmic rays detected in the `MultiDrizzle` run. Running `MultiDrizzle` is therefore a convenient way to perform a cosmic ray detection on the grism/prism images.

The combined direct image will be used to create a master catalogue with SExtractor. The master catalogue will then be projected back (see Chapt. 3.2.3) to generate Input Object Lists for each image in the `MultiDrizzle` combination to be used in the aXe reduction. Figure 3.1 shows an example of a combined direct image (left) and a combined grism image (right).

It is also possible to use other programs to identify cosmic ray hits on the grism/prism images. Then the information on the cosmics must be transported into the dq-extenton of the corresponding flt-image. aXe can exclude flagged pixels in the dq-extenton from the reduction. In the dq-extenton, cosmic ray affected pixels should be marked by adding the appropriate dq-flag 4096 (see ACS Data handbook) to the original dq value.

For grism images it is favourable (see Chapt. 3.2.3) to combine the direct and the grism images such that the final, `MultiDrizzled` images have the same

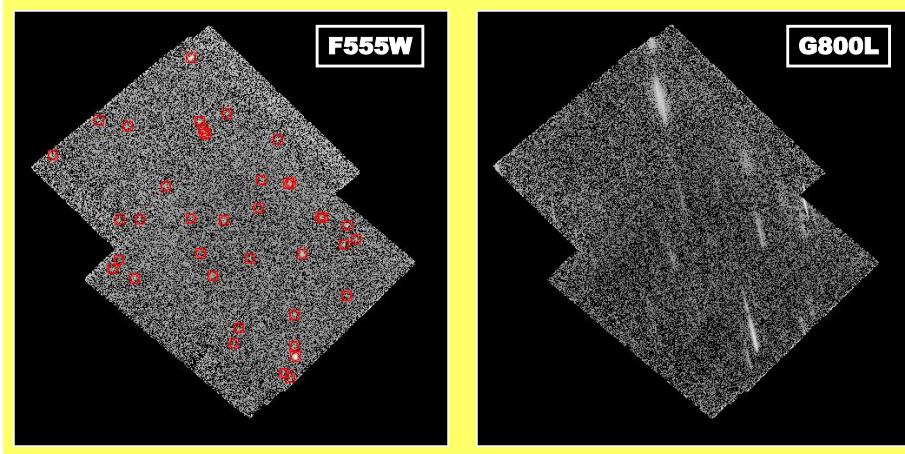


Figure 3.1: The MultiDrizzle-combined direct image (left) and the corresponding grism image (right). Objects identified in SExtractor are marked with red boxes on the direct image.

coordinate system. This means that each pixel (x, y) represents the same position (ra, dec) on the sky on both the combined direct as well as the combined grism image. The user can control this by e.g. specifying the identical center position and image size in the MultiDrizzle runs. The images in Fig. 3.1 fulfill this condition.

3.2.2 Master catalogue

The next step then is to create the master catalogue by running SExtractor on the combined direct image. Care should be taken when choosing the SExtractor parameters. Objects which are not in the master catalogue will later not be extracted from the grism images. Large numbers of fake objects or cosmics in the master catalogue on the other hand increase the computation time and simulate contributions to the contamination of real objects.

The master catalogue must contain all columns which are necessary for the spectral extraction with aXe (see format description in Chapt. 7.4). The first few lines of the master catalogue `f555w_drz.cat` extracted from the direct image in Fig. 3.1 are:

```
# 1 NUMBER           Running object number
# 2 X_IMAGE          Object position along x                  [pixel]
# 3 Y_IMAGE          Object position along y                  [pixel]
# 4 X_WORLD          Barycenter position along world x axis   [deg]
# 5 Y_WORLD          Barycenter position along world y axis   [deg]
# 6 A_IMAGE          Profile RMS along major axis            [pixel]
# 7 B_IMAGE          Profile RMS along minor axis            [pixel]
# 8 THETA_IMAGE      Position angle (CCW/x)                 [deg]
# 9 A_WORLD           Profile RMS along major axis (world units) [deg]
# 10 B_WORLD          Profile RMS along minor axis (world units) [deg]
# 11 THETA_WORLD     Position angle (CCW/world-x)           [deg]
# 12 MAG_F555W        Kron-like elliptical aperture magnitude [mag]
1 2116.6 815.9 5.322e+01 -2.781e+01 14.669 3.407 -79.3 1.0e-04 2.5e-05 -45.4 23.0
```

2	1463.7	740.1	5.322e+01	-2.781e+01	1.981	1.355	-84.5	1.3e-05	9.4e-06	-42.8	26.1
3	850.8	752.1	5.321e+01	-2.782e+01	1.877	1.749	21.9	1.2e-05	1.2e-05	37.9	24.6
4	1999.1	735.0	5.322e+01	-2.781e+01	0.952	0.465	-52.7	6.0e-06	4.1e-06	-67.4	28.0
5	760.5	761.5	5.321e+01	-2.782e+01	2.268	1.375	-54.4	1.4e-05	1.0e-05	-65.2	26.2
6	969.0	826.9	5.322e+01	-2.782e+01	3.863	1.672	37.6	2.4e-05	1.5e-05	22.2	25.0
7	781.3	831.3	5.322e+01	-2.782e+01	4.455	2.084	-58.9	2.9e-05	1.7e-05	-59.9	25.5
8	976.5	826.2	5.322e+01	-2.782e+01	2.410	0.839	-86.7	1.6e-05	5.7e-06	-41.6	26.3
9	1231.5	836.1	5.322e+01	-2.781e+01	2.522	1.177	-65.9	1.6e-05	9.3e-06	-53.8	25.9
10	981.5	834.9	5.322e+01	-2.782e+01	2.467	1.151	28.3	1.6e-05	9.5e-06	32.7	26.6

In the master catalogue the original column name `MAG_AUTO` was changed to `MAG_F555W`, a column name format which indicates the filter wavelength ($\lambda = 555.0\text{ nm}$, see Chapt. 7.4). This format allows a quantitative contamination estimate with the Gaussian emission model and the computation of optimal weights.

3.2.3 Preparing the Input Object Lists

Following user requests we have developed and introduced with aXe-1.5 the task `iolprep` (see Chapt. 4.1), a program to automatically generate Input Object Lists in a standard scenario such as described here.

The task `iolprep` searches in the header of a MultiDrizzle-combined image for the names and drizzle parameters of all input images. For each input image, the pixel coordinates ($x_{\text{comb}}, y_{\text{comb}}$) of all objects in the master catalogue, which is associated with the MultiDrizzle-combined image, are projected out into the coordinate system of the input image to derive the pixel coordinates ($x_{\text{input},i}, y_{\text{input},i}$) therein. For each input image an Input Object List is generated which comprises all objects which fall on the area covered by the input image. For the projections of the object positions, this aXe task uses the STS-DAS task `tran`.

There are two general strategies to apply `iolprep`:

1. **Creating IOL's for direct images** It is always (for grism and prism data) possible to apply `iolprep` with the direct image as the MultiDrizzle-combined image and the master catalogue derived from it. On a data set as described in Chapt. 3.1, the following IOL's would be produced:

```
j8m81cd9q_flt_1.cat, j8m824toq_flt_1.cat,  
j8m84aqkq_flt_1.cat, j8m851tmq_flt_1.cat.
```

As the file names suggest, the IOL's refer to the direct images, and during the spectral extraction a direct image **must** be given for every grism image (see Chapt. 7.3).

2. **Creating IOL's for grism images** If in the case of grism data MultiDrizzle was run such that the combined direct and grism image have the same coordinate system (see Chapt. 3.2.1), the object positions in the master catalogue are also valid for the combined grism image. It is then possible to apply `iolprep` with the grism image as the MultiDrizzle-combined image. In this case the IOL's refer to the input grism images and would be named:

```
j8m8201eq_flt_1.cat, j8m8201lq_flt_1.cat,
j8m8201rq_flt_1.cat, j8m820m4q_flt_1.cat,
j8m822q0q_flt_1.cat, j8m822q4q_flt_1.cat,
j8m822qbq_flt_1.cat, j8m822qhq_flt_1.cat.
```

In this scenario the IOL's refer directly to the grism images, as their file names indicate, and in the spectral extraction no direct image is needed.

The latter strategy has small advantages, such as it is easier to make the Input Image List (see below). It is possible to include objects in the Input Object List which have positions outside of the area covered by the corresponding direct image or grism image. In the case that the spectrum of the object falls partly on the grism image, but its reference point is outside, the spectrum covered by the grism image can still be reduced and contribute to the coadded spectrum of the object. Also higher orders of bright objects outside of the grism image can cause significant contamination on the grism images. Including them in the IOL means that their contamination is properly recorded and evaluated, even if no spectrum is extracted. The parameter `dimension.info` controls the effective area for the inclusion of objects in the task `iolprep`.

Depending on whether `iolprep` is run on the direct image `f555w_drz.fits` or the grism image `g800l_drz.fits`, the task is executed as:

```
-->iolprep mdrizzle_image='f555w_drz.fits' input_cat='f555w_drz.cat'
dimension_info=0,0,0,0
```

or alternatively:

```
-->iolprep mdrizzle_image='g800l_drz.fits' input_cat='f555w_drz.cat'
dimension_info=0,0,0,0
```

3.2.4 Preparing the fluxcube files

For grism images it is possible to apply the fluxcube emission model (see Chapt. 1.7.2) in the estimation of quantitative contamination. This requires the preparation of a fluxcube file for every grism image which is analyzed in aXe. For this purpose the task `fcubeprep` was developed.

Similar to `iolprep`, the task `fcubeprep` (see Chapt. 4.2) uses MultiDrizzled direct and grism images to build the fluxcube files. In addition, the SExtractor segmentation image which is associated to the master catalogue must also be provided. `fcubeprep` searches in the header of the MultiDrizzle-combined grism image for the names and drizzle parameters of all input grism images. Using the information on wavelength and zeropoints which are part of the input, the task transforms the direct images to flux units. Then the segmentation image and all direct flux images are projected into the coordinates of each input grism image to generate cutout images which match the area of the input grism images. For each input grism image, a fluxcube image is finally created from the corresponding segmentation and flux cutout images.

All images used in the input (MultiDrizzle-combined grism image, MultiDrizzle-combined direct images and segmentation images) **must** have been combined such that they have the same coordinate system. This means each pixel (x, y)

must represent the same position (ra, dec) on the sky on all input images (see Chapt. 3.2.1).

In case there are several MultiDrizzle-combined direct images in different filters available, the user must prepare a file and give for each image the name, central wavelength and zero point separated by ',' in a row. Provided that in addition to the direct image **f555w_drz.fits**, there exists also the image **f606w_drz.fits**, this file (name **dir_ims.lis**) looks like:

```
f555w_drz.fits, 431.8, 25.157
f606w_drz.fits, 591.8, 26.655
```

Note that instead of the 'nominal' values 555 and 606 the more accurate pivot wavelength values have been used for the ACS filters F555W and F606W. With the segmentation image **f555w_seg.fits** the task **fcubeprep** is executed as:

```
--> fcubeprep grism_image='g800l_drz.fits' segm_image='f555w_seg.fits'
filter_info='dir_ims.lis' AB_zero='yes' dimension_info=0,0,0,0
```

The task creates the fluxcubes:

```
j8m8201eq_flt_2.FLX.fits, j8m8201lq_flt_2.FLX.fits,
j8m8201rq_flt_2.FLX.fits, j8m820m4q_flt_2.FLX.fits,
j8m822q0q_flt_2.FLX.fits, j8m822q4q_flt_2.FLX.fits,
j8m822qbq_flt_2.FLX.fits, j8m822qhq_flt_2.FLX.fits.
```

3.3 Extracting spectra

3.3.1 Reduction Strategy

Before actually preparing and performing the data reduction, the user must decide which data reduction strategy to follow.

The main decisions are whether aXedrizzle is used or not and whether the background subtraction is done globally with the master background or with a local background for each beam (see Chapt. 1.6 for a comparison of the two methods).

aXedrizzle is currently not supported for prism data. Global background subtraction requires a master background for the instrumental configuration with which the data were taken with. The available master background images are posted on the instrument pages (<http://www.stsci.edu/hst/acs/analysis/STECF> and <http://www.stsci.edu/hst/wfc3/analysis/grism.obs/>), and users are urged to check whether a master background is available for their data.

If possible, the recommended reduction strategy is to do a global background subtraction and to use aXedrizzle. For the typical survey type data, this is the best way to reduce ACS grism data (see e.g. the GRAPES data paper, Pirzkal et al., 2004) or WFC3 grism data. In case only individual spectra in crowded fields are to be reduced, the reduction with a background PET may have advantages.

Depending on the reduction strategy, different High Level aXe Tasks (see Fig. 1.1) have to be applied to reduce the spectra. Table 3.1 lists the tasks and the order in which to apply them for the various reduction strategies.

number	+aXedrizzle/ +master sky	-aXedrizzle/ +master sky	+aXedrizzle/ -master sky	-aXedrizzle/ -master sky
1.	axeprep	axeprep	axeprep	axeprep
2.	axecore	axecore	axecore	axecore
3.	drzprep		drzprep	
4.	axedrizzle		axedrizzle	

Table 3.1: The high level aXe tasks to be applied in the different reduction strategies.

Input Object Lists versus Input Image Lists

Although the names are strikingly similar, Input Object Lists and Input Image Lists are fundamentally different. The format of an Input Object List is identical to the SExtractor ASCII catalogue format. An Input Object List contains positions and further (SExtractor-) information for all astronomical sources on the associated direct or grism image. The Input Object List is used as an input in the Low Level aXe Task `sex2go1` to derive a Grism Object List (GOL, see Chapt. 7.5) for a grism image.

On the other hand, Input Image Lists are an input parameter in all High Level Tasks. An Input Image List contains on each row the name of a grism image and additional information which is necessary to run the aXe task on that grism image. The additional information are the name(s) of Input Object List(s) and, optionally, a direct image name and a dmag-value. The exact descriptions of Input Object Lists and Input Image Lists, plus their formats, are in Chaps. 7.4 and 7.3, respectively.

3.3.2 Input Image List

The Input Image List is consistently used as the parameter `inlist` in all High Level Tasks. The Input Image List defines the combinations of Input Object Lists, grism images and, if necessary, direct images used in the spectral extraction.

In case that the IOL's refer directly to the grism images (see item 1. in Chapt. 3.2.3), the Input Image List `axeprep.lis` for the data presented here looks like:

```
j8m8201eq_flt.fits j8m8201eq_flt_1.cat 0.0
j8m8201lq_flt.fits j8m8201lq_flt_1.cat 0.0
j8m8201rq_flt.fits j8m8201rq_flt_1.cat 0.0
j8m820m4q_flt.fits j8m820m4q_flt_1.cat 0.0
j8m822q0q_flt.fits j8m822q0q_flt_1.cat 0.0
j8m822q4q_flt.fits j8m822q4q_flt_1.cat 0.0
j8m822qbq_flt.fits j8m822qbq_flt_1.cat 0.0
j8m822qhq_flt.fits j8m822qhq_flt_1.cat 0.0
```

If the IOL's refer to direct images, (see item 2. in Chapt. 3.2.3), the Input Image List `axeprep.lis` for the data presented in here looks like:

```
j8m8201eq_flt.fits j8m84aqkq_flt.cat j8m84aqkq_flt.fits 0.0
```

```
j8m8201lq_flt.fits j8m81cd9q_flt.cat j8m81cd9q_flt.fits 0.0
j8m8201rq_flt.fits j8m81cd9q_flt.cat j8m81cd9q_flt.fits 0.0
j8m820m4q_flt.fits j8m84aqkq_flt.cat j8m84aqkq_flt.fits 0.0
j8m822q0q_flt.fits j8m851tmq_flt.cat j8m851tmq_flt.fits 0.0
j8m822q4q_flt.fits j8m824toq_flt.cat j8m824toq_flt.fits 0.0
j8m822qbq_flt.fits j8m824toq_flt.cat j8m824toq_flt.fits 0.0
j8m822qhq_flt.fits j8m851tmq_flt.cat j8m851tmq_flt.fits 0.0
```

Every grism image is paired with the direct image taken at the closest position on the sky to provide the best overlap between objects in the IOL and the area covered by the grism image. The dmag-values are all set to the default 0.0, and therefore could be neglected here.

The exact format of the Input Image List is extensively described in Chapt. 7.3. All files are expected to be located in the directory indicated by the environment variable `AXE_IMAGE_PATH` (see Chapt. 5.1).

3.3.3 The aXe Configuration Files

The aXe configuration file describes the imprint of the spectrograph on the detector and contains essential parameters such as the description of the spectral trace and the dispersion solution together with their variations over the Field of View.

Up-to-date configuration files and the calibration files for all spectral modes are posted on the instrument pages (<http://www.stsci.edu/hst/acs/analysis/STECF> and http://www.stsci.edu/hst/wfc3/analysis/grism_obs/). The appropriate configuration file for the data presented in this Chapter is given below. To save space the descriptions of the higher order beams are neglected.

ACS.HRC.Cycle11.3.conf:

```
INSTRUMENT ACS
CAMERA HRC

# Calibrations for ACS HRC for Cycle 11 onward; released June 2004
# based on calibration data taken during SMOV and Cycle 11.
# Revised (3rd order) flat field cube:
#   ACS.HRC.flat.cube.2.fits
#
# Revised 1st and 2nd order sensitivity
# New 0th order dispersion solution and sensitivity
# New -1st order dispersion solution and sensitivity
# March 2009 (MK): keywords 'POBJSIZE' 'SMFACTOR' with dummy values added

SCIENCE_EXT SCI ; Science extension
DQ_EXT DQ ; DQ extension
ERRORS_EXT ERR ; Error extension
FFNAME ACS.HRC.flat.cube.2.fits
DQMASK      16383

EXPTIME EXPTIME
RDNOISE 4.71
```

```

POBJSIZE 1.0
SMFACTOR 1.0

DRZRESOLA 24.0
DRZSCALE 0.028
DRZLAMBO 4785.0
DRZXINI 15.0
DRZROOT aXedrizzle

# PSF variations for optimal extraction
PSFCOEFFS 8.20 -8.29e-02 4.01e-04 -9.47e-07 1.18e-09 -7.44e-13 1.87e-16
PSFRANGE 100.0 1100.0

# First order (BEAM A)
BEAMA 0 185
MMAG_EXTRACT_A 25
MMAG_MARK_A 27
# Trace description, 1st order
DYDX_ORDER_A 1
DYDX_A_0 0.0 0.0 0.0 0.0 0.0 0.0
DYDX_A_1 -0.796319 7.10246e-6 9.55948e-6
# X and Y Offsets
XOFF_A 0. 0. 0.
YOFF_A -1.78463 -0.000149007 0.000436432
# Dispersion solution, 2nd order
DISP_ORDER_A 2
DLDP_A_0 4783.55 0.00657371 -0.0126691
DLDP_A_1 23.5107 -0.000677401 0.00127958
DLDP_A_2 0.00170758 1.77847e-7 1.97777e-7
#
SENSITIVITY_A ACS.HRC.1st.sens.2.fits

```

Under normal circumstances the user can apply the aXe configuration files without any modifications. Only to speed up the computation time it might be convenient to modify some keywords (see Chapt. 3.4). The location of the configuration and calibration files is the directory indicated by the environment variable `AXE_CONFIG_PATH` (see Chapt. 5.1).

Note on data with multiple science extensions

For grism data with several science extensions (e.g. ACS WFC images or WFC3 UVIS), the input given in the second column of the Input Image List (the Input Object Lists), as well as the parameter `backims` in `axeprep` and the parameter `configs` in all High Level Tasks, must be a comma separated list. Each item in the list gives the input for a specific extension. The same relative positions in those lists must specify the same extensions. As Fig. 3.2 shows, this means that e.g. the Input Object List given as the second item in the Input Image List targets the same extension as the second item in the parameter `backims` and the second item in the parameter `configs`.

The ACS Wide Field Camera and the WFC3 UVIS Camera contain two CCD chips, and the data is stored in two independent extensions of the fits file. The

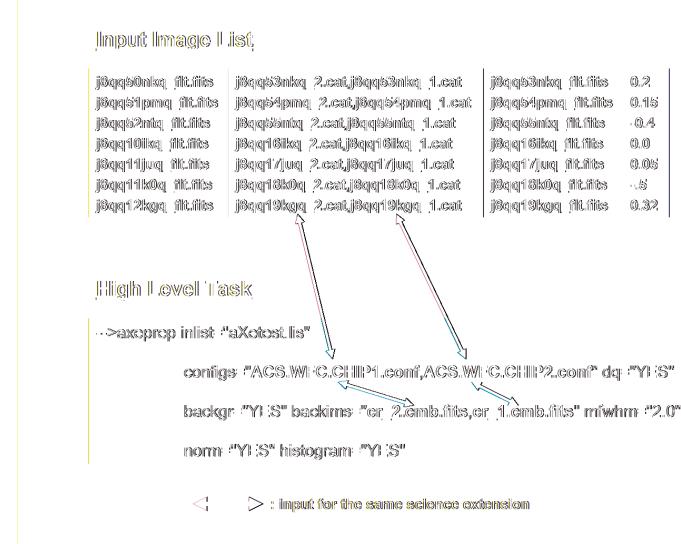


Figure 3.2: The Input Image List `axetest.lis` and a High Level aXe Task. The arrows connect input which refers to the identical science extension.

spectral reduction in aXe is done independently, using one configuration file for every science extension. In the ACS/WFC and WFC3/UVIS configuration files, the chip number is specified in the keywords “OPTKEY1” and “OPTVAL1”.

For technical reasons in both cameras the data of CCD chip No. 1 are stored in the **second** science extension version ([`sci`, 2] in PyRAF-fits notation), and the data of of CCD chip No. 2 is stored in the **first** science extension version ([`sci`, 1] in PyRAF-fits notation). Care must be taken to combine the correct files in the aXe input parameters, since the file names are often derived from these two counter-intuitive numbering schemes. While the file names of the configuration files follow the chip numbers (e.g. `ACS.WFC.CHIP1.Cycle13.2.conf` and `ACS.WFC.CHIP2.Cycle13.2.conf` are the configuration files for chip 1 and 2, respectively), the IOL’s created in `iolprep` follow the extension version number (the Input Object Lists `j8m822qhq_flt_1.cat` and `j8m822qhq_flt_2.cat` contain objects located on the fits image `j8m822qhq_flt.fits[sci,1]` and `j8m822qhq_flt.fits[sci,2]`, respectively). Figure 3.2 and the note on page 39 give further examples how to combine the input for ACS/WFC data in the various High Level Tasks.

3.3.4 Example reductions for the different scenarios

For the remainder of this section we present and describe sequences of High Level Tasks to reduce data according to the different strategies outlined in Chapt. 3.3.1. The High Level Tasks are listed with the correct syntax to be executed within an interactive PyRAF session.

aXedrizzle and Global Sky Subtraction In this reduction scenario the background is subtracted using the mastersky `HRC.back.fits`. For each object the 2D spectra on the individual grism images are combined to a deep, 2D grism spectrum with `aXedrizzle`, then the 1D spectrum is extracted from the coadded

2D grism spectrum. The quantitative contamination with the fluxcube emission model is chosen. This assumes that the fluxcube files were created beforehand (see Chapt. 3.2.4).

As in all further examples, optimal extraction is selected in the parameters. In aXe the optimal extracted spectra are always delivered **in addition** to the normal, equally weighted results. There is **no** need to run aXe twice, the optimal extractions only entails but a small additional amount of computing time.

The sequence of commands interactively applied in PyRAF is:

```
-->axeprep inlist="axeprep.lis" configs="ACS.HRC.Cycle11.2.conf"
    backims="HRC.back.fits" backgr="YES" fwhm="2.0"
    norm="YES" histogram="YES"
-->axecore inlist="axeprep.lis" configs="ACS.HRC.Cycle11.2.conf"
    back="NO" extrfwhm=4.0 drzfw=3.0
    backfwhm=0.0 slitless_geom="YES" orient="YES" exclude="NO"
    lambda_mark=800.0 cont_model="fluxcube" model_scale=3.0
    inter_type="linear" lambda_psf=555.0 spectr="NO"
    weights="NO" sampling="drizzle"
-->drzprep inlist="axeprep.lis" configs="ACS.HRC.Cycle11.2.conf"
    opt_extr="YES" back="NO"
-->axedrizzle inlist="axeprep.lis" configs="ACS.HRC.Cycle11.2.conf"
    infwhm=4.0 outfwhm=3.0 back="NO" makespc="YES"
    adj_sens="YES" opt_extr="YES"
```

The line breaks are added here for clarity, but on the actual command line each command should be given as one string. The most convenient way to specify the task parameters is with the PyRAF/IRAF `epar` mechanism.

No aXedrizzle and Global Sky Subtraction Here the background is globally subtracted using master sky images. The coaddition of the individual 2D spectra with aXedrizzle is not done. Gaussian contamination has been chosen. The command sequence is a subset of the command sequence in the last example, with small differences in the parameters:

```
-->axeprep inlist="axeprep.lis" configs="ACS.HRC.Cycle11.2.conf"
    backims="HRC.back.fits" backgr="YES" fwhm="2.0"
    norm="YES" histogram="YES"
-->axecore inlist="axeprep.lis" configs="ACS.HRC.Cycle11.2.conf"
    back="NO" extrfwhm=3.0 drzfw=0.0
    backfwhm=0.0 slitless_geom="YES" orient="YES" exclude="NO"
    lambda_mark=800.0 cont_model="gauss" model_scale=3.0
    inter_type="linear" lambda_psf=555.0 spectr="YES"
    adj_sens="YES" weights="YES" sampling="drizzle"
```

aXedrizzle and Background PET Here the background PETs are generated from background images which have interpolated pixel values at the beam positions. Both the image as well as the background are drizzled to deep 2D grism and background images, respectively (see Chapt. 1.6.2).

```
-->axeprep inlist="axeprep.lis" configs="ACS.HRC.Cycle11.2.conf"
```

```

        backgr="NO" fwhm="2.0"
        norm="YES" histogram="YES"
-->axecore inlist="axeprep.lis" configs="ACS.HRC.Cycle11.2.conf"
        back="YES" extrfwhm=4.0 drzffwhm=3.0
        backfwhm=4.0 slitless_geom="YES" orient="YES" exclude="NO"
        lambda_mark=800.0 cont_model="fluxcube" model_scale=3.0
        inter_type="linear" lambda_psf=555.0 spectr="NO"
        adj_sens="NO" weights="NO" sampling="drizzle"
-->drzprep inlist="axeprep.lis" configs="ACS.HRC.Cycle11.2.conf"
        opt_extr="YES" back="YES"
-->axedrizzle inlist="axeprep.lis" configs="ACS.HRC.Cycle11.2.conf"
        infwhm=4.0 outfwhm=3.0 back="YES" makespc="YES"
        opt_extr="YES"

```

No aXedrizzle and Background PET Both object and background spectra are extracted from each grism image individually. The background subtraction is done by subtracting the background PET from the object PET pixel by pixel. The command sequence is a subset of the command sequence given in the last example:

```

-->axeprep inlist="axeprep.lis" configs="ACS.HRC.Cycle11.2.conf"
        backgr="NO" fwhm="2.0"
        norm="YES" histogram="YES"
-->axecore inlist="axeprep.lis" configs="ACS.HRC.Cycle11.2.conf"
        back="YES" extrfwhm=3.0 drzffwhm=0.0
        backfwhm=0.0 slitless_geom="YES" orient="YES" exclude="NO"
        lambda_mark=800.0 cont_model="gauss" model_scale=3.0
        inter_type="linear" lambda_psf=555.0 spectr="YES"
        adj_sens="YES" weights="YES" sampling="drizzle"

```

3.4 The requirements

The aXe tasks are rather expensive in terms of computer time. Some of the main factors contributing to a large computational need are:

- the complete error propagation and the propagation of contamination information multiplies the computing effort per science pixel by a factor of ~ 3 , since errors as well as contamination are stored and treated similar to the science data;
- the necessary conversions of data format (image, PET, DPP, drizzled image) result in a high demand on input/output.

As a rule of thumb, each High Level Task needs around 0.3 sec of computing time per object and image on a typical Pentium five machine (2.3 GHz). For prism data with typically a few objects per image an aXe reduction is completed within a short period of time. In a survey type project, however, a typical data set consists of 10 ACS/WFC images and 1000 objects on each image. This results in around half a day of pure computing time. The minimum RAM requirement is around 2000 MB, which should not constitute a bottleneck on modern workstations.

3.5 Tuning tips

Especially for deep grism data, the computation time can be quite large, and users would like to speed up the processing time. There exist some measures to get the results more quickly.

3.5.1 Wavelength dependence of the PSF

For the ACS slitless modes, we have determined the dependency of the Point Spread Function (PSF) as a function of wavelength, and this dependency is used (via the configuration keywords `PSFCOEFFS` and `PSFRANGE`) when computing the Gaussian contamination. In general, computing the Gaussian contamination consumes a lot of processor time, especially when the wavelength dependence of the PSF is taken into account.

The aXe reduction on a particular data set is usually done several times with some small changes in the parameters to fine tune the results. The wavelength dependence of the PSF is not very large. Switching it off in the early reductions can save a lot of time without any significant influence on the results and their interpretations for the next runs.

To neglect the wavelength dependence, the keywords `PSFCOEFFS` and `PSFRANGE` must be commented out in the aXe configuration file (see Chapt.[3.3.3](#)).

For the WFC3/IR grism we were unable to detect a significant variation of the PSF with wavelength, hence there are no corresponding keywords (`PSFCOEFFS` and `PSFRANGE`) in the configuration files and aXe is always running in the “fast” mode.

3.5.2 Extraction of higher grism orders

The sensitivity of the higher grism orders (0th, 2nd, 3rd, -1st, -2nd) is typically (not WFC3 UVIS) very low compared to the first order. Most of the objects on a grism image are too faint to deliver a signal in any but the first order, and the extraction of the basically empty higher orders is very time consuming.

It is therefore very reasonable to set the extraction limits for the higher order spectra to a very low magnitude limit (setting the keywords in the configuration file e.g. `MMAG_EXTRACT_B 10`, `MMAG_EXTRACT_C 10`, ...) to prevent their extraction. Even if those higher order spectra are not extracted, they are still fully taken into account in the contamination analysis, since the brightness limits for objects to be included into the contamination analysis is controled by the keyword `MMAG_MARK_#` (see also next Chapter).

3.5.3 Limits for contamination

Similar to the extraction magnitudes, sensible limits for the contamination magnitudes (controlled by the keywords `MMAG_MARK_#`) can avoid superfluous computations. If e.g. the target has a brightness $mag = 20.0$, it may not be important to include the zeroth orders of all objects down to $mag = 25.0$ into the contamination estimate, since the contaminating contribution of the zeroth order of a $25\ mag$ object to the first order of the 100 times brighter object of $20\ mag$ is small. To find reasonable limits for the various orders, it is important to compare the throughput of the usually dominating first order spectrum with

Order name	Order letter	WFC3 ¹		ACS ¹	
		G102 [mag]	G141 [mag]	WFC [mag]	HRC [mag]
first	A	0.00	0.0	0.00	0.00
zeroth	B	0.15	0.5	1.42	1.95
second	C	3.52	3.1	3.58	2.75
third	D	5.58	5.9	4.15	3.33
-first	E	5.27	4.5	3.42	3.33
-second	F	-	-	5.04	4.00
-third	G	-	-	4.55	-

Table 3.2: The differential throughput of the various orders with respect to the first order for the WFC3/IR and ACS grisms. The quantities are given in magnitudes.

the throughput in the higher spectral orders. Table 3.2 lists the differential throughputs with respect to the first order in units [mag] for all WFC3/IR and ACS grisms. The quantities *diffmag* in Tab. 3.2 have the following meaning: For two objects 1 and 2 with magnitudes m_1 and m_2 , respectively, object 1 has, in the spectral order k , approximately the same count rates as object 2, in the first order, if

$$m_2 = m_1 - \text{diffmag}(k, \text{instr})$$

with *diffmag*(k, instr) the corresponding value for order k from Tab. 3.2. This table can be used to set reasonable limits for the keywords `MMAG_MARK_#` in the aXe configuration file.

Provided the user decides to compute the contamination down to the ratio of 1 : 10 or 2.5 mag between any contaminating beam and the target object (first order beam), and the target object has $\text{mag} = 20.0$, reasonable values for `MMAG_MARK_#` for ACS/WFC data would be

$$\begin{aligned} \text{MMAG_MARK_A} &= 20.0 + 2.5 \\ &= 22.5 \\ \text{MMAG_MARK_B} &= 20.0 + 2.5 - 1.42 \\ &= 21.08 \\ \text{MMAG_MARK_C} &= 20.0 + 2.5 - 3.58 \\ &= 18.92 \\ \text{MMAG_MARK_D} &= 20.0 + 2.5 - 4.15 \\ &= 18.35 \\ \text{MMAG_MARK_E} &= 20.0 + 2.5 - 3.42 \\ &= 19.08 \\ \text{MMAG_MARK_F} &= 20.0 + 2.5 - 5.02 \\ &= 17.48 \end{aligned}$$

¹For WFC3, the differential magnitudes were computed by simulating slitless spectra of compact objects and then comparing the brightest pixels across the various orders. For ACS, the sensitivity per resolution element was used to determine the differential magnitude for the various orders. In both instruments the “theoretical” values were compared and verified using real observations.

$$\begin{aligned} MMAG_MARK_G &= 20.0 + 2.5 - 4.55 \\ &= 17.95 \end{aligned}$$

These values would assure that all relevant beams are taken into account when computing the contamination, but also avoid the costly computation of negligible contamination contributions. The differential throughput values in Tab. 3.2 are derived from the order sensitivities applied to flat continuum sources; they may not be applicable to very red or very blue sources or emission line objects.

Chapter 4

aXe tasks

This chapter gives detailed descriptions of each of the aXe tasks and of their parameters. Examples of how to run each task are also included, as well as descriptions of the files required and produced by each task. A detailed description of all the output products can be found in Chapt. 7.

The aXe tasks use environment variables (see Chapt. 5.1) to define the locations of the direct and slitless images. In addition, all tasks are meant to work on specific extensions in the input FITS files and the file names of the output products of each aXe task reflect this by having a “_#” (where # is an extension number) appended to the original FITS file name (e.g. a grism_1.SPC.fits file will be produced if the input slitless FITS file was named grism.fits and if the science data of interest was located in extension 1). Selection of the extension to extract is defined in the configuration file (Chapt. 5.2.1) .

4.1 IOLPREP

This task produces Input Object Lists for every input image of a MultiDrizzle combined image. It projects the object positions from a master catalogue, which contains all objects in the coordinate system of the MultiDrizzled image, out into the coordinate system of each input image. For each input image an Input Object List is generated, which contains only the objects within the boundaries of the given input image.

The names and drizzle parameters of the input images are retrieved from the header of the MultiDrizzle combined image. The projection of the object positions into the coordinate system of the input images is done with the STSDAS task `wtranback`.

There is a parameter to influence the sensitive area to include objects in the IOL’s. This allows objects beyond the physical boundaries of the input image to be included in the IOL’s, to take into account partly covered objects or to include bright objects outside of the FOV in contamination estimates derived from those IOL’s.

During the task execution, the drizzle coefficient files and the input images must be available. For this reason it would be best practice to run it in the directory that was used to combine the input images with MultiDrizzle.

4.1.1 Usage

```
iolprep comb_image master_cat dim_info
```

4.1.2 Parameters

`mdrizzle_image`: the name of the combined (MultiDrizzled) image
`input_cat`: the name of the SExtractor master catalogue
`dimension_info`: four numbers to specify the modifications [left, right, bottom, top] to the target area on the input images. E.g. 100,500,10,0 would include in the Input Object Lists all objects with $-100 < x < x_size + 500$ and $-10 < y < y_size$.

Example:

```
axeprep mdrizzle_image='g8001_drz.fits' input_cat='bvri.cat'
dimension_info=200,0,0,0
```

4.1.3 Output

- `./[grism filename]_[science ext number].cat`

4.2 FCUBEPREP

The task produces fluxcubes for a set of grism images in a standard scenario. The user should have prepared a MultiDrizzled grism image and at least one MultiDrizzled direct image. Moreover there exists a segmentation map that was produced together with the master catalogue of objects in a SExtractor run. All the MultiDrizzled images must have the identical coordinate system, which means every pixel (i, j) must represent the identical position (*Ra*, *Dec*) on the sky

The task analyzes the header of the combined grism image and extracts the name and the drizzle parameters of each grism input image. The filter images are transformed from *counts/sec* to flux units, and the STSDAS task `blot` produces for each input grism image a set of cutout images from the segmentation and the flux images. The cutout images for each grism input image are finally combined to a fluxcube.

There exists the option to produce fluxcubes which have a different size than their associated grism image. This allows to include objects outside of the grism image FOV in the contamination estimate, or to restrict the contamination analysis to only a restricted area.

As with `iolprep` the drizzle coefficient files and the grism images must be available during task execution, and the best way to assure this would be running the task in the directory used for the drizzle combination of the grism images.

4.2.1 Usage

```
fcubeprep grism segmentation filtinfo zeropoint dim_info
```

4.2.2 Parameters

grism_image: the name of the combined (MultiDrizzled) grism image
segm_image: name of the segmentation image
filter_info: name, wavelength, zeropoint of the filter image(s).
 If there are several filter images the comma separated quantities are written in a file, and the name of this file is given here.
AB_zero: boolean to indicate whether the zeropoints given in the parameter 'filter_info' are in AB- or ST-magnitudes
dimension_info: four numbers to specify the modifications [left, right, bottom, top] to the target area on the grism images. E.g. 100,500,10,0 would produce fluxcube images which cover the area $-100 < x < x_size + 500$ and $-10 < y < y_size$ in the input grism images
interpol: the interpolation scheme used to compute flux values at the interpolated wavelengths

Example:

```
fcubeprep grism_image='g8001_drz.fits' segm_image='f850_seg.fits'  
filter_info='cubelisST.lis' ABzero='NO'
```

with the file cubelisST.lis:
 f435w_drz.fits, 431.8, 25.157
 f606w_drz.fits, 591.8, 26.655
 f775w_drz.fits, 769.3, 26.393
 f850lp_drz.fits, 905.5, 25.954

4.2.3 Output

- ./[grism filename]-[ext number].FLX.fits

4.3 AXEPREP

This task prepares the science files (e.g. ACS and WFC3 _flt files produced by the on-the-fly pipeline or the **calacs** task) for further processing within aXe. **axeprep** provides important keywords and is mandatory if **axedrizzle** is to be used later on.

axeprep provides three different processing steps:

- background subtraction:
 Provided that an Input Object List is given for the grism image, **axeprep** uses the tasks **sex2gol**, **gol2af** and **backest** to mark the beam areas on the grism image as well as on the master background image. Using the

IRAF task `imstat` with 3 cycles of clipping pixels with values $> 3\sigma$, the median pixel values are derived for the unmarked pixels on both the grism image and on the master background image. The master background, scaled to the level of the grism image, is finally subtracted from the grism image.

- exposure time normalization The input file is normalized by the exposure time to transform the images into counts per second.
- gain correction The input file is multiplied by the gain conversion factor (electrons/ADU) to transform the images from units of detector counts to electrons. For HST data, this is usually only necessary for NICMOS images, because ACS and WFC3 images will normally already be in units of electrons.

Every processing step can be switched on/off independently by associated boolean parameters.

For ACS/WFC and WFC3/UVIS images, the correspondence between the configuration files, the background images and the IOL's declared in the Input Image List is explained in Fig. 3.2.

The file used by `axeprep` as `inlist` can be reused again in `axecore`, `drzprep` and `axedrizzle`, perhaps extended with different `dmag`-values for the grism images.

4.3.1 Usage

```
axeprep inlist configs backgr backims mfw hm norm gaincorr
```

4.3.2 Parameters

`inlist`: Input Image List which gives on each line

- a) the name of the grism image to be processed (mandatory)
- b) the object catalog(s) (mandatory if `back='yes'`, comma separated list if there is more than one catalogue)
- c) the direct image associated with the grism image (optional)

`configs`: name of the axe configuration file. If several image extensions are to be processed (e.g. for WFC images), one configuration file per extension must be given in a comma separated list.

`background`: boolean to switch on/off background subtraction

`backims`: name of the background image. If several image extensions are to be processed (e.g. for WFC images), one background image per extension must be specified in a comma separated list.

`mfw hm`: real number to specify the extent (as a multiple of `A_IMAGE` or `B_IMAGE`) of the area that is masked out perpendicular to the trace of each object before the background level is

```

determined (see parameter mfw hm in gol2af).

norm: boolean to switch on/off the exposure time normalization

gaincorr: boolean to switch on/off the gain conversion

```

Example:

```

axe prep inlist='imlist.lis', configs='conf1.conf,conf2.conf',
    back='YES', backims='back1.fits,back2.fits', fwhm=2.0,
    norm='YES' gain='NO'

```

If `back='YES'`:

- `$AXE_OUTPUT_PATH/[slitless filename]_[ext number].MSK.fits`

ATTENTION:

The task `AXEPREP` changes the SCI (and potentially ERR) extensions of the grism images! It is highly recommended to work only on copies of the original files in order to be able to repeat the reduction with different parameters.

4.4 AXECORE

This aXe task combines the Low Level Tasks `sex2gol`, `gol2af`, `af2pet`, `petff`, `petcont`, `pet2spc` and `stamps` and offers the possibility to make a complete aXe reduction based on the individual images in one task. This also includes the reduction with background PETs (set `back='YES'`). The parameter list comprises all parameters for the individual tasks, and as a consequence is rather long. For most of the parameters the default value is appropriate, so the actual number of parameters that will normally need to be edited by the user is quite modest. In the listing below, the `axecore` parameters are organised according to the low-level aXe tasks they affect.

The Input Image List used in the task `axe prep` as `inlist` can be reused again in `axecore`, perhaps extended with individual `dmag`-values for the grism images.

The sequence of configuration files must correspond to the sequence of Input Object Lists and the sequence of background images in `inlist` (see Fig. 3.2).

If `axedrizzle` is not going to be used, the parameter `drz fwhm` should be set to 0.0. Otherwise the values of parameters `extrfwhm` and `drz fwhm` in the task `axecore` must correspond to the values to be used for parameters `infwhm` and `outfwhm` in the task `axedrizzle`, respectively. The extraction width used after drizzling, which is specified via `drz fwhm` and `outfwhm`, must be larger than the extraction width for the PET's in `axecore`, which is set by `extrfwhm` (see Chapt. 4.6).

4.4.1 Usage

```
axecore inlist configs extrfwhm drz fwhm back backfwhm orient slitless_geom exclude ...
```

4.4.2 Parameters

inlist: Input Image List which gives on each line
 a) the name of the grism image to be processed (mandatory)
 b) the object catalog(s) (mandatory)
 c) the direct image associated with the grism image (optional)
 d) dmag value (see GOL2AF) for the grism image (optional)

configs: name of the axe configuration file. If several image extensions are to be processed (e.g. for WFC images), one configuration file per extension must be given in a comma separated list.

back: Boolean to switch on/off the creation of a background PET with mfw hm=backfwhm

[The following parameters apply to GOL2AF:]

extrfwhm: mfw hm value to specify the extraction width in gol2af
 drz fwhm: mfw hm value to specify the extraction in axedrizzle
 backfwhm: mfw hm value to specify the width of the background PET
 orient: enable tilted extraction
 slitless_geom: enable the best extraction for slitless spectroscopy
 exclude: switch off the listing of faint objects
 lambda_mark: the wavelength at which to apply the cutoff magnitudes
 MMAG_EXTRACT and MMAG_MARK

[The following parameters apply to PETCONT:]

cont_model: name of the contamination model to be applied
 model_scale: scale factor for the gaussian contamination model
 interp_type: interpolation type for the flux values
 lambda_psf: wavelength [nm] at which the object widths were measured

[The following parameters apply to BACKEST:]

np: number of points for background estimation
 interp: interpolation type for background determination
 (-1: GLOBAL median; 0: local median; 1: linear fit;
 2: quadratic fit)

```

niter_med: number of kappa-sigma iterations around the median

niter_fit: number of kappa-sigma iterations around the fit value

kappa:      kappa value

smooth_length: number of adjacent pixels on each side to use when
                smoothing the background estimate in x-direction

smooth_fwhm: FWHM of the Gaussian used in the background smoothing

[The following parameters apply to PET2SPC and STAMPS:]

spectr:    enable the creation of SPCs and STPs for each of the
            grism files individually

weights:   compute and apply optimal weights

adj_sens:  adjust the sensitivity function for extended sources

sampling:  the sampling mode for the stamp images

```

Example:

```

axecore inlist='imlist.lis' configs='conf1,conf2' back='YES'
        extrfwhm=4.0 backfwhm=5.0 exclude='NO' cont_model='gauss'
        model_scale=4.0 interp_type='linear' lambda_psf=600.0
        slitless_geom='YES' np=10 interp=1 spectr='YES' adj_sens='YES'

```

4.4.3 Output

- \$AXE_OUTPUT_PATH/[slitless filename]_[ext number].cat
- \$AXE_OUTPUT_PATH/[slitless filename]_[ext number].OAF
- \$AXE_OUTPUT_PATH/[slitless filename]_[ext number].PET.fits
- \$AXE_OUTPUT_PATH/[slitless filename]_[ext number].CONT.fits

If `back='YES'`:

- \$AXE_OUTPUT_PATH/[slitless filename]_[ext number].BAF
- \$AXE_OUTPUT_PATH/[slitless filename]_[ext number].BCK.fits
- \$AXE_OUTPUT_PATH/[slitless filename]_[ext number].BCK.PET.fits

If `drzfwlm>0 and cont_model='geometric'`:

- \$AXE_OUTPUT_PATH/[slitless filename][drzfwlm]_[ext number].OAF
 this file is used to recompute the contamination in the PET's, using the
 value specified in `drzfwlm` as extraction width

If `spectr='YES'`:

- \$AXE_OUTPUT_PATH/[slitless filename]_[ext number].STP.fits
- \$AXE_OUTPUT_PATH/[slitless filename]_[ext number].SPC.fits

If `spectr='YES'` and `weights='YES'`:

- \$AXE_OUTPUT_PATH/[slitless filename]_[ext number]_opt.SPC.fits
- \$AXE_OUTPUT_PATH/[slitless filename]_[ext number]_opt.WHT.fits

4.5 DRZPREP

This task produces a set of Drizzle PrePare (DPP) files for a set of images given in an Input Image List. A DPP-file is a multi-extension fits file with a pixel stamp image, an error stamp image and a contamination stamp image for each first order beam in a grism image. DRZPREP uses the PET file to derive the pixel/error/contamination values for the stamp images and the background aperture file (BAFs) to define a common geometry for the individual objects. The need for a common geometry for all stamp images of a single object forces `drzprep` to be run always on the set of images which later are also combined with `axedrizzle`. If there is more than one set of PETs for each grism image (as in the case of WFC data), the configuration files should be given as a comma separated list in the parameter `configs`.

The task also derives and stores important keywords for `axedrizzle`. In the Input Image List given with the parameter 'inlist' the first item on each line must be the name of the grism image. Further columns/items are neglected by `drzprep`. Therefore the file used as `inlist` in `axecore` and `axeprep` can be re-used in `drzprep` again.

4.5.1 Usage

```
drzprep imagelist configs back
```

4.5.2 Parameters

`inlist`: Input Image List which gives the name of the grism image to be processed as the first item on each line.

`configs`: name of the aXe configuration file. If several image extensions are to be processed (e.g. for WFC images), one configuration file per extension must be given in a comma separated list.

`opt_extr`: boolean to generate also the necessary data for optimal extraction in `axedrizzle`

`back`: boolean to switch on the creation of background DPPs made by processing background PETs.

Example:

```
drzprep inlist='axeprep.lis' configs='aXe_config1.conf,aXe_config2.conf'
```

```
back='NO'
```

4.5.3 Output

If `back='NO'`:

- `$AXE_DRIZZLE_PATH/[slitless filename]_[ext number].DPP.fits` or

If `back='YES'`:

- `$AXE_DRIZZLE_PATH/[slitless filename]_[ext number].BCK.DPP.fits`

4.6 AXEDRIZZLE

This is the central task to the reduction method described in Chapt. 1.8. This task takes the DPPs prepared by `drzprep` as input. The extensions for the various objects are extracted from the DPP, and the extracted stamp of each object are drizzled together to form a deep, 2D drizzled grism image for each object. For a description of the drizzle algorithm, see Fruchter & Hook (2002)

The drizzle coefficients computed by `drzprep` for each stamp image are given as header keywords and are computed in such a way that the combined 2D drizzled grism image resembles an ideal grism image with a constant dispersion and a constant pixelscale in the cross-dispersion direction. The trace of the drizzled spectra is parallel to the x-axis of the image. The dispersion and the pixelscale (in cross-dispersion direction) are set in the aXe configuration file with the keywords `DRZRESOLA` and `DRZSCALE`, respectively (see Chapt. 5.2). At present, only the first order beams of images can be drizzled.

Drizzling usually creates pixels with incomplete coverage at the borders of the drizzle images. To avoid those pixels with their lower weight entering the 1D extraction, the extraction width used in the 1D extraction from the 2D drizzled grism images should be *smaller* than the extraction width used to generate the PETs in `axecore`. The extraction width (in multiples of the object `fwhm`) for the 1D extraction must be specified with the parameter `outfwhm`, while the parameter `infwhm` must be set to the value that was used in `axecore` to create the PET's and therefore the DPP's. `infwhm` and `outfwhm` in the task `axedrizzle` therefore directly correspond in `axecore` to the parameters `extrfwhm` and `drzfwhm`, respectively. Then the task `axedrizzle` can recalculate the extraction width for the 1D extraction. Typical value pairs for (`infwhm`, `outfwhm`) and (`extrfwhm`, `drzfwhm`) are (4.0,3.0) or (3.0,2.0). A value given in `axecore` cannot be corrected or changed in the task `axedrizzle`.

In addition to the 2D drizzled grism images, `axedrizzle` creates all the necessary files to facilitate the extraction of the 1D spectra with the tasks `drz2pet` and `pet2spc`. Usually, these additional steps are carried out automatically within `axedrizzle` (if `makespc='YES'`). To drizzle the background DPPs, the task `axedrizzle` must be run with `back='YES'`. If the drizzling of the background is done *after* the drizzling of the object DPPs, the background is correctly taken into account in the reduction of the 1D spectra.

The Input Image List given with the parameter `inlist` must contain the name of the grism image as the first item on each line. The name of the corresponding DPP file(s) are then derived from the grism name and the chip as specified in the configuration file(a). Further columns/items are neglected. Therefore files used as 'inlist' in `axecore` and `axeprep` can be reused in `axedrizzle` again.

With `driz_separate="YES"` aXedrizzle identifies and excludes deviant values coming from e.g. hot or cosmic ray affected pixels. In this mode, `axedrizzle` works similar to MultiDrizzle ([5]) on direct images. As indicated below, this method requires many parameters. Their name and their meaning is identical as in Multidrizzle, and user are referred to the MultiDrizzle documentation for details. `axedrizzle` with the rejection of pixels works only if the background of the grism images had been removed with global background subtraction (see task `axeprep`, Chapt. 1.6.1).

4.6.1 Usage

```
axedrizzle inlist configs infwhm outfwhm back makespc
```

4.6.2 Parameters

`inlist`: Input Image List with the input grism filename as the first item on each line.

`configs`: name of the aXe configuration file. If several image extensions (and therefore DPPs) are to be processed (e.g. for WFC images), one configuration file per extension must be given in a comma separated list.

`infwhm`: `mfwfwm` for the input PETs and DPPs

`outfwhm`: `mfwfwm` for the extraction of the objects in later steps

`back`: boolean to drizzle both the object and the background DPP

`clean`: boolean to remove temporary files

`makespc`: boolean to switch on/off whether SPCs shall be created directly from the coadded images

`adj_sens`: adjust the sensitivity function for extended sources

`opt_extr`: boolean to switch on the optimal extraction in addition to the regular, exposure time weighted extraction

`driz_separate`: drizzling to separate image and CCR reject ("YES") or "simple" aXedrizzle ("NO")

[The following parameters apply only for `driz_separate="YES":`]

[The following parameters apply for MEDIAN IMAGE parameters:]

```
combine_type:      type of combine operation, [median|sum|minmed|minimum]
combine_maskpt:   percentage of weight image below which it is flagged as bad
combine_nsigma:   significance for accepting minimum instead of median
                  (for combine_type="minmed")
combine_nlow:     number of low pixels to reject
combine_nhigh:    number of high pixels to reject
combine_lthresh:  lower threshold for clipping input pixel values
combine_hthresh:  upper threshold for clipping input pixel values
combine_grow:     radius (pixels) for neighbor rejection
```

[The following parameters apply for BLOT BACK:]

```
blot_interp:      interpolant [nearest,linear,poly3,poly5,sinc]
blot_sinscl:     scale for sinc interpolation kernel
```

[The following parameters apply for COSMIC RAYS REMOVAL:]

```
driz_cr_snr:      driz_cr.SNR parameter
driz_cr_grow:    driz_cr_grow parameter
driz_cr_scale:   driz_cr.scale parameter
```

Example:

```
axedrizzle inlist="axegrism.lis" configs="HUDF.HRC.conf" infwhm=4.0
outfwhm=3.0 back="NO" makespc="YES" adj_sens="YES"
driz_separate="NO"
```

4.6.3 Output

If for an input name ./[drizzle root filename]_2.list:

- \$AXE_CONFIG_PATH/[drizzle root filename].conf
- \$AXE_DRIZZLE_PATH/[drizzle root filename]_2.OAF
- \$AXE_DRIZZLE_PATH/[drizzle root filename]_mef_ID[num].fits
- BACK='YES': \$AXE_DRIZZLE_PATH/[drizzle root filename]_mef_ID[num].BCK.fits

If makespc='YES':

- \$AXE_DRIZZLE_PATH/[drizzle root filename]_2.SPC.fits
- \$AXE_DRIZZLE_PATH/[drizzle root filename]_2.STP.fits

If `makespc='YES'` and `opt_weight='YES'`:

- \$AXE_DRIZZLE_PATH/[drizzle root filename]_2_opt.STP.fits
- \$AXE_DRIZZLE_PATH/[drizzle root filename]_2_opt.WHT.fits

4.7 SEX2GOL

This task generates a Grism Object List file using an Input Object List as input. There are three different kinds of Input Object List that can be fed into aXe:

- an Input Object List (in SExtractor format) of objects on a direct image covering (roughly) the same field as the grism image
- an Input Object List in SExtractor format, which gives the objects on the grism image in world coordinates (RA, Dec and theta_sky)
- an Input Object List in SExtractor format, which lists the objects on the grism image in world coordinates and image coordinates (x_image, y_image and theta_image)

A thorough description of the Input Object List is given in Chapt. 7.4.

For the first two ways to specify object lists, the image coordinates of the objects on the grism image will be recomputed using the WCS information of the grism image and the direct image. This approach therefore relies on the accuracy of the WCS information given in those images. Refer to section 7.4 for a description of what values should be in the the input catalog and which ones can be re-constructed by SEX2GOL.

4.7.1 Usage

```
sex2gol grism config in_sex use_direct direct dir_hdu
          spec_hdu out_sex
```

4.7.2 Parameters

<code>grism:</code>	name of the grism image to be processed.
<code>config:</code>	name of the axe configuration file.
<code>in_sex:</code>	name of the object file.
<code>use_direct:</code>	boolean to indicate that the Input Object List refers to a direct image
<code>direct:</code>	name of the direct image
<code>dir_hdu:</code>	direct image extension to be used

```
spec_hdu: grism/prism image extension to be used
out_SEX: overwrites the default output object catalog name
```

Example:

```
sex2gol grism='test_grismn.fits' config='SLIM.conf.test.0'
      in_sex='test_0.cat' use_direct='NO'
```

4.7.3 Output

- \$AXE_OUTPUT_PATH/[slitless filename]_[ext number].cat

4.8 GOL2AF

This task generates an Aperture File using an input Grism Object List and a valid configuration file which defines the length, wavelength calibration and global offsets between direct and slitless images. For positive numbers in `mfwhm` the extraction width of the BEAMs is set up to be this number times the width listed in the Grism Object List. Negative numbers specify the extraction width for all objects in pixels directly (see Chapt. 1.10 for a detailed discussion). Two magnitude cutoffs are set in the Configuration File (Chapt. 5.2). Sources which have magnitudes fainter than an extraction cutoff magnitude are flagged so that they are not extracted, but will be accounted for when computing spectral contamination and the background estimates. Sources which have magnitudes fainter than another cutoff magnitude are marked so that they will be completely ignored. The `dmag` value can be used to globally adjust these cutoffs (to account for a different signal-to-noise ratio in one dataset for example without having to resort to editing of the configuration file).

This task can be used to generate both an Object Aperture File and a Background Aperture File. While these files have a similar format, it is often desirable to use different Aperture Files for the two cases. This is because the former is used to extract counts from pixels which are known to contain flux from the source, while the latter can be thought to define a zone to avoid all source flux in the slitless image when computing the background level (in the case that a master sky is not used for background subtraction, see Chapt. 1.6). In practice, a larger extraction width multiplier should be used when generating the Background Aperture File so that all the object flux is properly isolated when generating a Background Estimate File (Chapt. 7.7).

With `orient='YES'`, GOL2AF extracts the beams with an extraction angle parallel to the semi-major-axis of the object. `orient='NO'` forces a vertical extraction perpendicular to the spectral trace of the beam. For `orient='YES'` and `slitless_geom='YES'` however GOL2AF adjusts the extraction angle when the desired extraction angle forms too small an angle with the spectral trace ($|\alpha| < 35^\circ$). Then the extraction angle follows the semi-minor-axis instead of the semi-major-axis of the object which results in more pixels being extracted from the slitless image (see Chapt. 1.10 and Fig. 1.12 for more details).

4.8.1 Usage

```
gol2af grism config mfw hm dmag back slitless_geom orient exclude
sci_hdu out_af in_gol
```

4.8.2 Parameters

grism: name of the grism image

config: name of the aXe configuration file

mfw hm: the extraction width multiplicative factor

back: to generate a BAF instead of an OAF file

orient: boolean to switch on/off tilted extraction

slitless_geom: boolean to switch on/off automatic orientation
for the tilted extraction

exclude: boolean to switch on the removal of faint objects
in the result

lambda_mark: the wavelength at which to apply the cutoff magnitudes
MMAG_EXTRACT and MMAG_MARK

dmag: a number to add to the MMAG_EXTRACT and MMAG_MARK
values given in the configuration file

out_af: overwrites the default output OAF or BAF filename

in_gol: overwrites the default input catalog name

Example:

```
gol2af grism='test_grismn.fits' config='SLIM.conf.test.0' mfw hm=4.0
back='YES'
```

4.8.3 Output

If back='NO':

- \$AXE_OUTPUT_PATH/[slitless filename]_[ext number].OAF

If back='YES':

- \$AXE_OUTPUT_PATH/[slitless filename]_[ext number].BAF

4.9 BACKEST

This task uses the input slitless image and a Background Aperture File to generate a Background Estimate File (Chapt. 7.7) . This task is applicable when a

master sky is not used for background subtraction (Chapt. 1.6). The number of points to use and the order of the interpolation to use to generate the Background Estimate File can be set using online parameters. The values in the regions within each of the BEAMs listed in the Background Estimate File are replaced by the median, average, linear, or n^{th} order polynomial interpolation of pixels which are immediately above and below a BEAM (but not within any other BEAM). The number of pixels to use for fitting is by default set to 10 on each side below and above the BEAM (therefore 20 pixels in total). The value given for the `np` option can be used to change this default value. If the number of points is set to a value which is 0 or less, then the entire column of an image will be used, ignoring any pixels which are within any known BEAM. This option allows for a full column background estimate to be created, instead of a local background estimate. The type of interpolation is controlled by the parameter `interp`:

- `interp= -1` ; Median
- `interp= 0` ; Average
- `interp= 1` ; Linear fit
- `interp= (n > 1)` ; n^{th} order polynomial fit ;

In case that bad pixels or cosmics are not marked in the dq-extention of the flt-file, it is possible to execute a number of kappa-sigma klipping iterations prior to the final fit. The kappa-sigma iterations exclude extreme pixel values created by defects limit their impact on the fit.

To further suppress the noise in the background, it is possible to apply a Gaussian smoothing in x-direction on the fitted background values. This is controled by the parameters `smooth_length` and `smooth_fwhm`, which give the number of adjacent pixels used for calculating the smoothed value on each side of a background pixel and the FWHM of the Gaussian, respectively.

4.9.1 Usage

```
backest grism config np interp niter_med niter_fit kappa
```

4.9.2 Parameters

`grism`: name of the grism image

`config`: name of the aXe configuration file

`np`: the number of pixels used on each side of a beam
to compute the median/average/fitted background

`interp`: the type of interpolation to perform

`niter_med`: number of kappa-sigma iterations around the median

`niter_fit`: number of kappa-sigma iterations around the fit value

```

kappa:      kappa value

smooth_length: number of adjacent pixels on each side to use when
                smoothing the background estimate in x-direction

smooth_fwhm: FWHM of the Gaussian used in the background smoothing

mask:        create a mask image with the OAF file

in_af:       overwrite the default input aperture filename

out_back:   overwrite the default output background filename

```

Example:

```
backest grism='test_grismn.fits' config='SLIM.conf.test.0' np=10 interp=1
```

4.9.3 Output

If `mask='NO'`:

- `$AXE_OUTPUT_PATH/[slitless filename]_[ext number].BCK.fits`

If `mask='YES'`:

- `$AXE_OUTPUT_PATH/[slitless filename]_[ext number].MSK.fits`

4.10 AF2PET

This task uses the input slitless image together with an Object Aperture File to generate a Pixel Extraction Table (PET) for the input data. The same task should be used with the Background Estimate File and the same Object Aperture File to generate a Background Pixel Extraction Table containing information about the spectral background (BPET).

4.10.1 Usage

```
af2pet grism config back out_pet
```

4.10.2 Parameters

`grism:` name of the grism image

`config:` name of the aXe configuration file

`back:` generate a PET for a background image using
a BAF file instead of a OAF file and using a
background image generated by backest

`out_PET:` overwrite the default output PET filename

Example:

```
af2pet grism='test_grismn.fits' config='SLIM.conf.test.0' back='YES'
```

4.10.3 Output

If `back='NO'`:

- `$AXE_OUTPUT_PATH/[slitless filename]_[ext number].PET.fits`

If `back='YES'`:

- `$AXE_OUTPUT_PATH/[slitless filename]_[ext number].BCK.PET.fits`

4.11 PETCONT

The task computes and stores the contamination information for a given Pixel Extraction Table. There are two distinct ways to compute the contamination:

- The geometrical contamination records, for each PET pixel, how often it is a member of a different beam. If a pixel is a member of two separate beams, i.e. is in a region where two beams overlap, it is assigned a value of 1 in each of the two beam PET's, thus indicating that this pixel is also part of another beam.
- In quantitative contamination, the amount of contaminating flux from other beams is estimated for each PET pixel. This estimate is based on a model of the emitting sources. There are two different methods to establish an emission model , the **gaussian emission model** and the **fluxcube model**. Chapt. 1.7.2 gives a detailed discussion on the emission models and their implications for the contamination.

The right panel of Fig. 1.3 is a geometrical contamination image, which carries the basic information about contamination.

4.11.1 Usage

```
petcont grism config cont_model model_scale inter_type cont_map
```

4.11.2 Parameters

`grism`: name of the grism image

`config`: name of the aXe configuration file

`cont_model`: name of the contamination model to be applied

`model_scale`: scale factor for the gaussian cont. model

`spec_models`: name of the multi-extension fits table with model spectra

`object_models`: name of the multi-extension fits image with object templates.

```

interp_type: interpolation type for the flux values

lambda_psf: wavelength [nm] at which the object widths were measured

cont_map: write the contamination map into a FITS file

in_af: overwrites the input AF file name

```

Example:

```
petcont grism='test_grismn.fits' config='SLIM.conf.test.0' cont_map='YES'
```

4.11.3 Output

- Updates \$AXE_OUTPUT_PATH/[slitless filename]_[ext number].PET.fits
- \$AXE_OUTPUT_PATH/[slitless filename]_[ext number].CONT.fits

4.12 PETFF

This task uses a flat-field calibration file to flat-field the content of a Pixel Extraction Table (see Chapt. 7.8). The wavelength of a pixel is used in conjunction with a flat-fielding data cube containing the coefficients of a polynomial which can be used to compute at each pixel (x,y):

$$FF(x, y, x) = a_0(x, y) + a_1(x, y) * x + \dots + a_i * x^i,$$

where,

$$x = (\lambda - \lambda_{min}) / (\lambda_{max} - \lambda_{min})$$

The coefficients $a_0(x, y)$ are stored in the first data extension of the flat-field cube, $a_1(x, y)$ in the second, etc... The values for λ_{max} and λ_{min} are in the FITS header keywords *WMIN* and *WMAX*. The name of the flat-field cube is read from the aXe configuration file using the parameter FFNAME . Chapter 6.2 gives a detailed description of the flatfield.

Note on aXe wavelength dependent flat-fielding:

The same wavelength dependent flat-fielding should be applied to both the OPET and the BPET separately so that when subtracted, the information contained in the BPET has been flat-fielded with the same flat-field as the OPET (which contains electron count rates of objects plus electron count rates of the background). Doing so ensures that the result of subtracting the content of the BPET from the OPET are in electron count rates for all objects which have been properly wavelength dependent flat-fielded. Flat fielding is performed by dividing the pixel value by the computed flat-field coefficient. Also note that this task only computes proper flat-field coefficients for values of wavelength which are within the range *WMIN* to *WMAX*. The task takes the flat-field value computed at the wavelength *WMIN* for wavelengths smaller than *WMIN*, and the flat-field value *WMAX* for wavelengths larger than *WMAX*.

4.12.1 Usage

```
petff grism config back ffname
```

4.12.2 Parameters

grism: name of the grism image

config: name of the aXe configuration file

back: apply FF to the background Pixel Extraction Table (BPET)

ffname: overwrite the default input flat-field cube name

Example:

```
petff grism='test_grismn.fits' config='SLIM.conf.test.0' back='YES'
```

4.12.3 Output

If back='NO':

- Updates \$AXE_OUTPUT_PATH/[slitless filename]_[ext number].PET.fits

If back='YES':

- Updates \$AXE_OUTPUT_PATH/[slitless filename]_[ext number].BPET.fits

4.13 PET2SPC

This task is used to transform the content of an Object Pixel Extraction Table into a set of 1D binned spectra in an Extracted Spectra File (see Chapt. 7.11). The binning process is explained in more detail in Chapt. 1.5 and allows the application of optimal weights (see Chapt. 1.9).

The task can be used simultaneously with both an Object Pixel Extraction Table and a Background Pixel Extraction Table, in which case a background subtraction is performed. Care must be taken that both Object and Background Pixel Extraction Tables were created with the same Aperture File. Additionally, absolute flux calibration can be performed if the proper information is included in the Main Configuration File.

4.13.1 Usage

```
pet2spc grism config use_bpert adj_sens do_flux
```

4.13.2 Parameters

grism: name of the grism image

config: name of the aXe configuration file

```

use_bpet: use of a BPET file

adj_sens: adjust the sensitivity function for extended sources

weights: compute and apply optimal weights

do_flux: do flux calibration

drzpath: use AXE_DRIZZLE_PATH for IN/Output?

in_af: overwrite the default input Aperture File name

opet: overwrite the default input Object PET file name

bpel: overwrite the default input Background PET file name

out_spc: overwrite the default output SPC file name

```

Example:

```
pet2spc grism='test_grismn.fits' config='SLIM.conf.test.0' use_bpet='YES'
```

4.13.3 Output

If drzpath='NO':

- \$AXE_OUTPUT_PATH/[slitless filename]_[ext number].SPC.fits

If drzpath='YES':

- \$AXE_DRIZZLE_PATH/[slitless filename]_[ext number].SPC.fits

If weights='YES':

- \$AXE_OUTPUT/DRIZZLE_PATH/[slitless filename]_[ext number]_opt.STP.fits
- \$AXE_OUTPUT/DRIZZLE_PATH/[slitless filename]_[ext number]_opt.WHT.fits

4.14 STAMPS

This task uses the content of a Pixel Extraction Table (see Chapt. 7.8) to generate a FITS Stamp Image File (see Chapt. 7.12) containing stamp images of the BEAMs that were extracted. This task can output various types of stamp images. In addition to the usual **trace**-stamp images which display the beams as they appear on the grism images, the **rectified** stamp images order the PET pixels in a rectangular grid using the XI and DIST values of the pixels. For **drizzled** stamp images the pixels are resampled onto a rectangular grid with wavelength and trace distance as the axes and with a constant dispersion. The first order beams are resampled to the dispersion specified in the keyword DRZRESOLA or, as all other beams, to the average dispersion in the PET pixels.

The stamp images allow a quick visual check on the extraction process. Moreover the drizzled stamp images can be used as an input for alternative 1D extractions with other iraf or IDL tools.

4.14.1 Usage

```
stamps grism config sampling
```

4.14.2 Parameters

grism: name of the grism image

config: name of the aXe configuration file

sampling: the sampling type

drzpath: use AXE_DRIZZLE_PATH for IN/Output?

in_af: non standard OAF name

in_PET: non standard PET name

out_stp: non standard STP name

Example:

```
stamps grism='test_grismn.fits' config='SLIM.conf.test.0' sampling='rectified'
```

4.14.3 Output

If drzpath='NO':

- \$AXE_OUTPUT_PATH/[slitless filename]_[ext number].STP.fits

If drzpath='YES':

- \$AXE_DRIZZLE_PATH/[slitless filename]_[ext number].STP.fits

4.15 DRZ2PET

This task produces one object PET (background BPET if back='YES') from a set of images created with AXEDRIZZLE. On this PET the task `pet2spc` can then perform the extraction of the 1D spectra for the drizzled grism images.

All the necessary input files (OAF/BAF, image list, modified configuration file) are automatically created by the AXEDRIZZLE task. The sequence of the images in the image list must match the sequence of the beams in the OAF. Interactive changes to the image list and/or the OAF are not recommended.

The 1D extraction of the 2D drizzled grism spectra is usually done within `axedrizzle` by calls to the tasks `drz2pet` and `pet2spc`.

The task `drz2pet` also sets the pixel weights to reflect the different signal-to-noise (S/N) ratios in each pixel. The S/N variations are caused by the masking of bad and cosmic ray affected pixels and by the partial coverage of objects on the border of grism object. The pixels that will be co-added into a single resolution element in the 1D spectra are weighted according to their relative exposure times. Moreover it is in addition possible to compute and store optimal weights to enhance the signal-to-noise (S/N) ratio in the 1D extracted spectra.

4.15.1 Usage

```
drz2pet inlist config opt_extr back
```

4.15.2 Parameters

inlist: ascii list which gives the name of the grism image to be processed as the first item on each line.

config: name of the axe configuration file(s).

opt_extr: boolean to set the computation and storage of optimal weights

back: boolean to switch on/off the creation of background PETs made from drizzled background images.

in_af: non standard OAF name

out_pet: non standard PET name

Example:

```
drz2pet inlist='axedrizzle_2.lis' config='axedrizzle.conf' back='NO'
```

4.15.3 Output

If **back='NO'**:

- \$AXE_DRIZZLE_PATH/[drizzle root filename]_[ext number].PET.fits

If **back='YES'**:

- \$AXE_DRIZZLE_PATH/[drizzle root filename]_[ext number].BCK.PET.fits

4.16 AXEGPS

This task reports the spectral properties of a single pixel. The spectral properties for individual pixels can only be assigned with respect to a reference point or reference beam. **axegps** lists:

- the wavelength at pixel center
- the dispersion at pixel center
- the trace distance of the section point
- the distance of the pixel center to the section point
- the data value of the pixel

The task **axegps** works on the .OAF file. The corresponding OAF file and the reference beam therein must therefore exist before **axegps** can give a result.

For numerical reasons a solution can only be guaranteed within the bounding box of the specified beam. The extraction width as specified with the parameter '`extrfwhm`' in `axecore` (or '`mfw hm`' in `gol2af`) has an influence on the bounding box. In the case that the desired information for the pixel of interest is not given, a repetition of `axecore` (or `gol2af`) with a larger value of '`drzfwhm`' ('`mfw hm`') may enlarge the bounding box sufficiently to get a result from `axegps`.

Even in case of failure, the corner points which define the bounding box of the beam are listed in the output such that the user can understand why the pixel information could not be computed.

4.16.1 Usage

```
axegps grism config beam_ref xval yval
```

4.16.2 Parameters

`grism`: name of the grism image

`config`: name of aXe configuration file used to create the OAF

`beam_ref`: the beam to define the spectral solutions

`xval`: the x-coordinate of the pixel

`yval`: the y-coordinate of the pixel

Example:

```
axegps grism="j8m822qhq_flt.fits" config="HDF.HRC.conf"
      beam_ref="3A" xval=102 yval=588
```

4.16.3 Output

All output is directly printed to the standard output.

Chapter 5

Configuration of aXe tasks

The aXe tasks are configured in three different ways:

- environment Variables
- configuration files
- online parameters to aXe tasks

5.1 Environment Variables

All aXe tasks use the following environment variables:

- AXE_IMAGE_PATH: the path where the input data is located
- AXE_OUTPUT_PATH: the path where all aXe outputs, except the drizzle related files, will be directed
- AXE_DRIZZLE_PATH: the path where the drizzle outputs will be directed
- AXE_CONFIG_PATH: the path where the aXe configuration files are located

These can be set before running the aXe tasks or by a PyRAF script which runs all the aXe tasks in the desired order.

Using csh/tcsh:

```
setenv AXE_IMAGE_PATH /path/to/data/
setenv AXE_OUTPUT_PATH /path/to/output/directory/
setenv AXE_DRIZZLE_PATH /path/to/drizzle/directory/
setenv AXE_CONFIG_PATH /path/to/the/axe/config/
```

Using bash:

```
export AXE_IMAGE_PATH=/path/to/data/
export AXE_OUTPUT_PATH=/path/to/output/directory/
export AXE_DRIZZLE_PATH=/path/to/drizzle/directory/
export AXE_CONFIG_PATH=/path/to/axe/config/
```

5.2 Configuration Files

5.2.1 Main Configuration File

Many configuration parameters are read in by the aXe tasks from a single text file which serves as the primary means to configure the extraction process for a given mode of an instrument. A separate Main Configuration File should be created for each of the spectral modes of each of the instruments with which aXe tasks are to be used. This configuration file contains a basic geometrical description of where in the slitless image one would expect a given BEAM to be located relative to the position of the source object in a direct image. The character ";" can be used to add comments to this file. A general description of the format of the input data (location of the science, error and data quality arrays) is also included in this file.

General configuration

The following keywords in the Main Configuration file are used to define several parameters such as which extension of the input FITS images contain the data, which keywords should be used to determine the exposure time of the input data, etc...

- INSTRUMENT [string] The name of the instrument to which this configuration file applies (optional).
- CAMERA [string] The name of the camera (optional).
- SCIENCE_EXT [string or integer] The name of the FITS extension containing the data array (if a string) or the number of the extension containing the data array (if an integer).
- ERRORS_EXT [string or integer] The name of the FITS extension containing the error array (if a string) or the number of the extension containing the error array (if an integer). Set to "-1" if no error array is to be read in.
- DQ_EXT [string or integer] The name of the FITS extension containing the data quality array (if a string) or the number of the extension containing the data quality array (if an integer). Set to "-1" if no data quality array is to be read in.
- DQMASK [integer] This integer value determines which bits in the data quality array must not be set in order that a given pixel is considered to be good. The integer value is logically AND'ed with the actual data quality value of each pixel. If the result is non-zero the pixel will be flagged as bad and ignored in aXe tasks. The data quality value assigned to each pixel in `calacs` and updated in `axeprep` has different flag values for the various pixel deficiencies (see the ACS and the WFC3 Data Handbooks for the exact codes). The flag values for 'new hot pixels' and 'cosmic ray rejected pixels' for example are 16 and 8192, respectively. To flag both the new hot pixels and cosmic ray rejected pixels the value for DQMASK must be set to $8192 + 16 = 8208$. To flag all non-zero values in the data quality array, DQMASK must be set to 16383.

- EXPTIME [string or float] If set to a string, this keyword defines which FITS header keyword will be read in the data array FITS extension in order to define the exposure time of the data. If set to a float, then this value is used instead. The exposure time is used to compute pixel errors if an error extention in the flt image is missing and to compute the variance for optimal extraction.
- POBJSIZE [float] The size (rms) of point-like objects. for the given configuration. Used for the smoothed flux conversion and as a minimum value for the object size (see Sect. 1.10 and box on page 24).
- SMFACTOR [float] Empirical correction factor to apply in the smoothed flux conversion (see Sect. 1.11).
- RDNOISE[float] The readnoise of the detector. This quantity is used to compute pixel errors in case that there is no explicit error extention in the flt-images and to compute the variance in optimal extraction.
- PSFCOEFFS [float, float, ...] The numbers give the coefficients of the polynomial describing the variations of the PSF as a function of the wavelength in [nm]
- PSFRANGE [float, float] The two numbers give the lower and the upper wavelength range (in [nm]) for the application of the pynomial defined with the keyword PSFCOEFFS. Beyond that range the value at the border is used.
- PSF_OFFSET_# [float] Inserting a grism or prism into the optical beam might degrade the PSF of the instrument, and the object widths as measured on a direct filter image must be corrected for this by applying an offset.
- FFNAME [string] The name of the configuration file for the FITS data cube containing the flat-field model.
- OPTKEY1 [string] The name of a keyword in the FITS headers to identify the proper extension to read (e.g. "CCDTYPE")
- OPTVAL1 [string] The value that the FITS header keyword defined by OPTKEY1 must have in order to be selected (e.g. "1"). The OPTKEY1, OPTVAL1 pair allow to select the proper chip from a multi-extension ACS WFC image for example.
- REFX [int] The 2D field dependence contained in the configuration file is by default taken to be with respect to pixel (0,0). The parameter REFX and REFY can be set to different values. For example, these parameters can be used when a 2D field dependence with respect to the center of the image is required.
- REFY [int] See REFX.
- DRZRESOLA [real] The dispersion (in Å/pixel) for the drizzled first order beams.

- DRZSCALE [real] The pixelscale (in " per pixel) in the cross-dispersion direction in the drizzled beams.
- DRZLAMB0 [real] The reference wavelength (in Å) which is drizzled to the reference pixel in the drizzled beams.
- DRZXINI [real] The x-value of the reference pixel in the drizzled images. The reference wavelength given in DRZLAM0 is drizzled to this reference pixel. The y-value of the reference pixel depends on the object width and the extraction width. For a given drizzled beam, the y-value of the reference pixel is at $\text{real}(ny/2) + 1.0$ where ny is the number of rows in the drizzled beam.
- DRZPFRAC [real] The pixfrac-value used in `axedrizzle`.
- DRZKERNEL [string] The drizzle kernel to be used in `axedrizzle`. All kernels available in drizzle v2.92 are allowed. Those kernels are: [*square,point,turbo,gaussian,tophat,lanczos2,lanczos3*]. See the help for `drizzle` and `multidrizzle` and note on page 5.2.1 for more details.
- DRZROOT [string] The root name for the output files created in `axedrizzle`. The string 'hrcudf' given as DRZROOT would result in the drizzled beams 'hrcudf_ext_ID1.fits', 'hrcudf_ext_ID2.fits', ..., the OAF/BAF 'hrcudf_2.OAF/BAF', the drizzle configuration file 'hrcudf.conf', the list of drizzled images 'hrcudf_2.lis' and the dummy image 'hrcudf.fits'.

Note on the optimum settings for drizzle:

The drizzle code (see Fruchter & Hook 2002) offers several parameters to enhance the resolution of the drizzle-combined image. The most important parameters, `pixfrac` and `kernel`, can be set in the aXe configuration file as the keywords DRZPFRAC and DRZKERNEL, respectively. In the calls to drizzle, the task `axedrizzle` directly uses these corresponding parameters. In the selection of the optimal aXe settings (and therefore drizzle parameters) to enhance the image resolution in grism spectroscopy, the user has to apply the same rules as for direct imaging.

BEAM configuration

There must be a description for each of the BEAMs (i.e. dispersion orders) that are extracted. BEAMs are named using single letter characters ('A','B','C', etc., for a maximum number of 26 BEAMs). All pixel coordinates and offsets that appear in a BEAM description are in fact offsets from the reference pixel in the BEAM (REFPIXEL## in Aperture File). The following is defined for each BEAM:

- Magnitude cutoffs
- Trace description
- Wavelength calibration description
- Sensitivity

Magnitude cutoffs

- MMAG_EXTRACT [float] The maximum magnitude listed in the input object catalog for this BEAM to be extracted during the extraction process. Objects fainter than this cutoff magnitude will not be extracted. They will however be avoided when computing the background estimate and will be used to flag extracted spectra for contamination (unless otherwise determined by the MMAG_MARK parameter).
- MMAG_MARK [float] Objects that have an input catalog magnitude greater than this will be completely ignored and not accounted for. This BEAM will not be used at all for anything and will not be avoided when computing the background estimate.

Trace description

The following items apply to the BEAM "#". The character "A" through "Z" should be substituted for "#".

Note on Field Dependent Values:

Each configuration file parameter such as DLDPs, DYDXs, XOFF, or YOFF can be followed by a single [float] value. In this case, this unique value is used for the entire image, independently of position in the image. The same parameter can also be followed by a series of 3, 6,..., ($m^2/2+m/2$) values in which case these define a 2D field dependent polynomial which is to be used at a given position (x,y) of the image to actually compute the value of the parameter. A field dependence of the parameters DLDPs, DYDXs, XOFF, and YOFF can therefore be taken into account if it has been previously calibrated and the field dependence can be fitted by an m^{th} order 2D polynomial. The 2D field dependent polynomials can be different for every parameter listed in the aXe Configuration File. The 2D polynomials are by default taken to be with respect to pixel (0,0) but this behaviour can be changed by setting the parameters REFX and REFY to the appropriate values. The REFX and REFY values are first subtracted from the (x,y) coordinates of a pixel before computing values using the 2D field dependent polynomials. The order of the coefficients should always be given as shown in the following few examples:

```
n:1 a0
n:2 a0 + a1 * X + a2 * Y
n:3 a0 + a1 * X + a2 * Y + a3 * X2 + a4 * X * Y + a5 * Y2
n:4 a0 + a1 * X + a2 * Y + a3 * X2 + a4 * X * Y + a5 * Y2 + a6 * X3 + a7 * X2 * Y + a8 * X * Y2 + a9 * Y3
where X = x - REFX and Y = y - YREF
```

Note that the set of DLDP and DYDX parameters are themselves individual parameters of a set of polynomial equations (described above) which are independent of any field dependent effect. Expressing each of the n coefficients of an n^{th} order polynomial as an m^{th} order 2D (x,y) field dependent polynomial can seem a little awkward at first but allows for a maximum amount of flexibility when calibrating smoothly varying quantities.

- BEAM# [int] [int] The extent of the spectrum in the row (X) direction with respect to the reference pixel of this BEAM. The location of the reference pixel of this beam with respect to the direct image position is defined by the parameters XOFF and YOFF listed below. The beam row extent is measured independently of the position angle and always along the column direction.
- DYDX_ORDER_# [int] The order of the polynomial $\Delta y = P(\Delta x) = a_0 + a_1 * \Delta x + a_2 * \Delta x^2 + \dots \Delta x$ and $P(\Delta x)$ which determines the actual location of the trace of the spectrum in this BEAM (See description of this process in Chapt. 1.2).
- DYDX_#_0 [int] [...] For each of the orders n as specified by the DISP_ORDER_A, an entry of the form DYDX_#.n must exist. This can be a field dependent representation (see note in Chapt. 5.2.1).
- XOFF [float] A pixel row offset between the reference pixel of this BEAM

and the position of the object in the Direct Image. This can be a field dependent representation (see note on page [5.2.1](#)).

- YOFF [float] A pixel column offset between the reference pixel of this BEAM and the position of the object in the Direct Image. This can be a field dependent representation (see note on page [5.2.1](#)).

Wavelength calibration description for grisms

The wavelength calibration is handled using an n^{th} order polynomial which, as is the case for the Trace description, can be field dependent. The field dependence format is the same as for the trace description.

- DISP_ORDER_# [int] The order of the polynomial of the form $\lambda(x_i) = a_0 + a_1 * x_i + a_2 * x_i^2 + \dots$ which defines the wavelength at a distance x_i along the spectral trace.
- DLDP_#.0 [int] [...] Value of the parameter a_0 , which can be a field dependent representation as described for the Trace description.
- DLDP_#.1 [int] [...] Value of the parameter a_1 , which can be a field dependent representation as described for the Trace description.
- DLDP_#.2 [int] [...] Value of the parameter a_2 , which can be a field dependent representation as described for the Trace description.
- DLDP_#.n Value of the parameter a_n , which can be a field dependent representation as described for the Trace description.

Wavelength calibration description for prisms

The wavelength calibration is handled using an n^{th} order inverse polynomial which, as is the case for the trace description, can be field dependent. The field dependent format is the same as for the trace description.

- DISP_ORDER_# [int] The order of the inverse polynomial of the form $\lambda(x_i) = a_1 + a_2/(x_i - a_0) + a_3/(x_i - a_0)^2 + \dots$
- DLD1P_#.0 [int] [...] Value of the parameter a_0 , which can be a field dependent representation as described for the Trace description.
- DLD1P_#.1 [int] [...] Value of the parameter a_1 , which can be a field dependent representation as described for the Trace description.
- DLD1P_#.2 [int] [...] Value of the parameter a_2 , which can be a field dependent representation as described for the Trace description.
- DLD1P_#.n Value of the parameter a_n , which can be a field dependent representation as described for the trace description.
- DLD1P_#.PRANGE [int] [int] In the form of the dispersion relation given above, the singularity at $x_i = a_0$ divides the inverse polynomial into the two branches $x_i - a_0 < 0$ and $x_i - a_0 > 0$. The desired solution for the dispersion relation is on only one branch. The finite pointspread function and extended sources however require a beam definition which extends

from the valid branch over the singularity at $x_i = a_0$ partly into the second, invalid branch. To avoid that pixels from the invalid branch enter the PET and the spectra, this keyword defines the minimum and maximum values for $x_i - a_0$ which are allowed in the PET. Thus pixels from the invalid branch can be excluded.

Note on How to set up your own BEAM description:

- Choose a pixel on the spectrum as the Reference Pixel in the BEAM. With respect to this pixel the dispersion relation and trace description will be calibrated. This can be any pixel on the spectrum itself as long as it remains consistent, and the DYDX and DLDP polynomials are formulated so that they are with both respect to this pixel (i.e. in a reference frame where the Reference Pixel has coordinates (0,0)).
- Determine the XOFF and YOFF row and column offsets between the direct object coordinates in the object catalog and your Reference Pixel. These quantities can be field dependent.
- Determine the extent of this BEAM in the row direction and with respect to the BEAM Reference Pixel (e.g. -10 100 for a spectral order which extends -10 pixels to the left and 100 pixels to the right of your reference pixel)
- Calibrate the geometrical trace of this order (DYDX polynomial). This can be a field dependent relation. The resulting polynomial should return the Y column pixel offsets from the BEAM Reference Pixel as a function of X row pixels.
- Calibrate the wavelength dispersion of this order (DLDP or DLD1P polynomial). This can be a field dependent relation. The resulting polynomial should return the wavelength of a pixel lying a distance L, along the trace (as defined by the DYDX description), from the BEAM Reference Pixel.

Sensitivity

The absolute sensitivity calibration is handled by applying a sensitivity curve to the electron count rates at each wavelength.

- SENSITIVITY_# [string] The name of a sensitivity FITS file. If no sensitivity is available this keyword can be set to "None" instead of a real filename.

5.2.2 Example of a Main Configuration file

See Chapter 3.3.3.

Chapter 6

aXe Calibration Files

The aXe tasks use several calibration files in addition to the information contained in the Main Configuration File (see Chapt. 5.2.1). This section describes these files.

6.1 Sensitivity Curve

This file is a FITS binary table containing the three columns WAVELENGTH, SENSITIVITY, ERROR and listing the total system sensitivity (in e^-/s per $\text{erg cm}^{-2} \text{s}^{-1} \text{\AA}^{-1}$) as a function of wavelength (in \AA). The ERROR column should contain the estimated error in the SENSITIVITY (in e^-/s per $\text{erg cm}^{-2} \text{s}^{-1} \text{\AA}^{-1}$). Note: this sensitivity curve should be per \AA and not per pixel.

6.2 Flat field

This file is a multiple extension FITS file containing a model of the wavelength dependence of the flat-field for each pixel. Each extension i of this file contains the n^{th} polynomial coefficient of the relation $f(i, j, x) = a_0 + a_1 * x + a_2 * x^2 + \dots + a_i * x^n$, where x is a normalized value obtained with $x = (\lambda - \lambda_{min}) / (\lambda_{max} - \lambda_{min})$ and λ is the wavelength of the pixel (i,j). The values for λ_{max} and λ_{min} are in the FITS header keywords *WMIN* and *WMAX*. There are no hard limits on the number of extensions in this file, i.e. on the order of the polynomial model. The task `petff` is used to read this file, compute and apply the flat-field coefficient at each pixel contained in a Pixel Extraction Table. This is done by *dividing* the pixel value by the computed flat-field coefficient. The structure of this file is shown in figure 6.1. Note that the first extension of this file [0] contains the constant term of the polynomial.

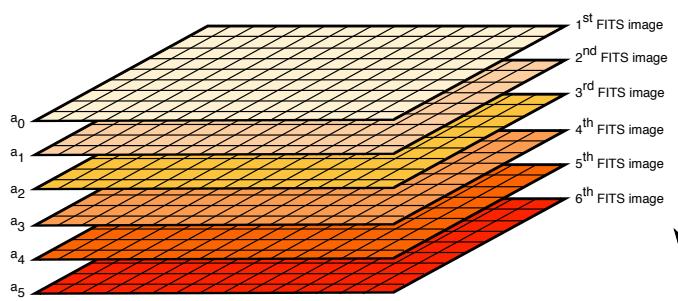


Figure 6.1: The structure of the FITS flat-field calibration file which is used by aXe to construct, at each pixel coordinate (i,j) a proper flat-field coefficient $FF(i, j, x) = a_0 + a_1 * x + a_2 * x^2 + \dots + a_i * x^n$, where x is a normalized value obtained with $x = (\lambda - \lambda_{min}) / (\lambda_{max} - \lambda_{min})$ and λ is the wavelength of the pixel (i,j)

Chapter 7

File Formats

This chapter describes the file formats of the intermediate data products generated by the aXe tasks. All files used by the aXe tasks are either ASCII files, FITS binary images with multiple extensions, or FITS binary tables containing multiple extensions. Separate BEAMs are kept by all aXe tasks in separate FITS extensions.

7.1 Input Images

The input images must be in FITS format. Any FITS file following the FITS standard with binary image extensions can be used as input to the aXe tasks. A WCS (CD matrix) should be present in the header of the FITS extension to be read for some of the aXe tasks to work properly (`sex2gol`).

All standard pipeline processed input files are multi extension fits-files with the science, error and data quality array(s) in various extensions.

There are two ways to specify a fits extension in an aXe configuration file. One way is to address the fits extension number. Here aXe follows the convention of the CFITSIO library, whereby the primary extension (which is always present) has number 1, the first image extension number 2, the second image number 3 and so on. For an HRC image, the lines

```
SCIENCE_EXT 2
ERROR_EXT    3
DQ_EXT       4
```

in an aXe configuration file specify which fits extensions to use as science, error and data quality arrays, respectively. Another way to specify a fits extension in aXe is to use the extension names. In case of an HRC image, the lines

```
SCIENCE_EXT SCI
ERROR_EXT    ERR
DQ_EXT       DQ
```

are equivalent to the ones given before.

An aXe configuration file can target only one science array plus its associated error and data quality arrays. For ACS/WFC and WFC3/UVIS images, the data from its two chips is stored in separate extensions. To fully process these

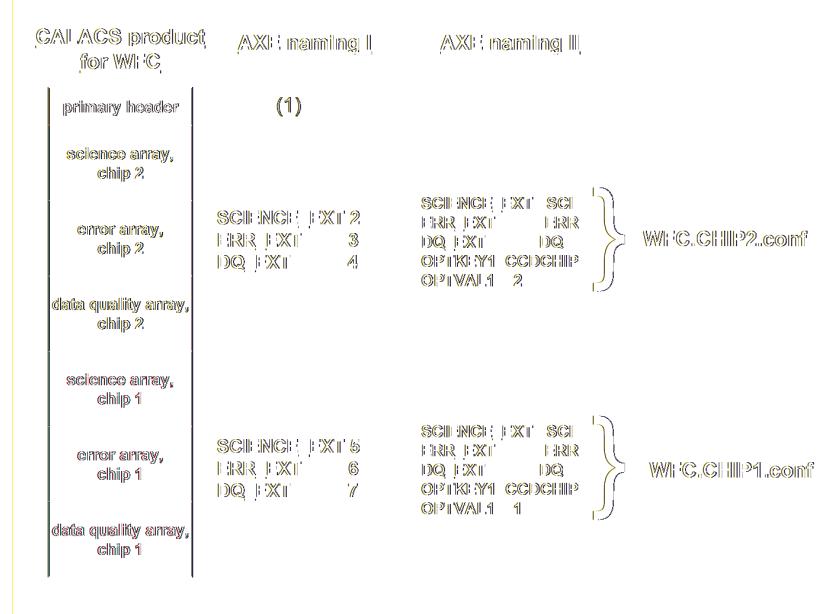


Figure 7.1: The naming methods to specify WFC fits extensions in the aXe configuration files named “WFC.CHIP1.conf” and “WFC.CHIP2.conf”

images in aXe two processing runs with two different configuration files have to be undertaken. With the fits extension numbers, the extensions can be uniquely specified using the scheme given above. In case that extension names are used, additional information must be provided, since there exist e.g. two extensions with the extension name ‘SCI’. In the aXe configuration file this additional information is the chip number, which is specified using the keywords ‘OPTKEY1’ and ‘OPTVAL1’. For ACS/WFC and WFC3/UVIS data, the chip notation defined by the archive is counterintuitive, since the data from chip 1 is stored at a higher extension number than chip 2 (see [ACS Data Hanbook](#)). To specify the data from chip 1, in addition to the extension names, the keywords ‘OPTKEY1’ and ‘OPTVAL1’ must be set to ‘CCDCHIP’ and ‘1’, respectively. For chip 2 data the keyword ‘OPTVAL1’ must be set to ‘2’. Figure 7.1 clarifies the two naming conventions that can be used in the aXe configuration files named “WFC.CHIP1.conf” and “WFC.CHIP2.conf”. The configuration files shown in Chapt. 3.3.3 also show how the different extensions in the WFC must be addressed.

7.2 Fluxcube file

The fluxcube files are a necessary input to compute a quantitative contamination estimate with the so called fluxcube emission model. This model is extensively described in Chapt. 1.7.2. The fluxcube files have the file extension `*.FLX.fits` (a typical fluxcube name is e.g. `j8mt20kiq.flx_2.FLX.fits`, which is the fluxcube for the grism image `j8mt20kiq.flt.fits[sci,1]`).

The fluxcube files are generated in the task `fcubeprep` (see Chaps. 3.2.4 and

[4.2\)](#) from MultiDrizzled direct images covering the area of the grism images. Fluxcubes are multiextension fits images. The first image extension is a so called Segmentation image. It shows for each pixel the number of the object to which the pixel belongs in the emission model of the corresponding grism image. All other extensions are flux images at different wavelengths. The wavelength of every flux image is indicated in the name of the fits extension (e.g. the fits extension `j8mt20kiq_flt_2.FLX.fits[LAMBDA769]` is a flux image at 769 nm).

In the computation of the quantitative contamination with the fluxcube emission model, a fluxcube file must exist for every grism image extension. All extensions cover approximately the identical FoV as the corresponding grism image extension. The keywords `XOFFS` and `YOFFS` in the primary fits header mark the offsets between the pixel coordinates in the grism image and in the fluxcube.

7.3 Input Image List

The Input Image List is a flexible file format used in the High Level Tasks (`axeprep`, `execore`, `drzprep`, `axedrizzle`) to specify for each grism image the necessary information for the aXe reduction. The file format is identical for all High Level Tasks. Once the user has produced an Input Image List for a particular data set, it can be used in all High Level Tasks (for the parameter *in-list*). Each row lists a grism image and the additional filenames and information to reduce the grism image with aXe.

The columns list:

1. grism image name (mandatory)
2. input object list 1, input object list 2, ... (mandatory)
3. direct image (optional)
4. dmag value (optional)

If the grism image has more than one science extension, the Input Object List corresponding to each science extension must be specified as comma separated list in the second column. If the Input Object Lists refer to a direct image instead of the grism image itself, the name of the grism image should be listed in the third row. The fourth row holds the individual dmag value for the grism image (see task `gol2af` in Chapt. [4.8](#)). The third and fourth columns are optional and can be omitted. A number in the third row will be interpreted as the dmag-value.

The following example shows some rows taken from an Input Image List for a WFC data set.

```
j8qq50nkq_flt.fits j8qq53nkq_1.cat,j8qq53nkq_2.cat j8qq53nkq_flt.fits 0.2
j8qq51pmq_flt.fits j8qq54pmq_1.cat,j8qq54pmq_2.cat j8qq54pmq_flt.fits 0.15
j8qq52ntq_flt.fits j8qq55ntq_1.cat,j8qq55ntq_2.cat j8qq55ntq_flt.fits -0.4
j8qq10ikq_flt.fits j8qq16ikq_1.cat,j8qq16ikq_2.cat j8qq16ikq_flt.fits 0.0
j8qq11juq_flt.fits j8qq17juq_1.cat,j8qq17juq_2.cat j8qq17juq_flt.fits 0.05
j8qq11k0q_flt.fits j8qq18k0q_1.cat,j8qq18k0q_2.cat j8qq18k0q_flt.fits -.5
j8qq12kgq_flt.fits j8qq19kgq_1.cat,j8qq19kgq_2.cat j8qq19kgq_flt.fits 0.32
```

7.4 Input Object List

This file is a simple ASCII file containing tabulated information about objects to be extracted. It has the same format as a SExtractor 2.x output object catalog. The first few lines contain the name and description of each of the columns in the tabulated portion of this file.

To extract the spectra, aXe must know the exact location the objects would have on the grism/prism image if a filter instead of the grism/prism would have been used. The aXe task `sex2go1` uses the Input Object Lists plus further image information to generate a Grism Object List, which contains all the necessary grism image coordinates of the objects.

The Input Object List (IOL) and therefore the SExtractor catalogue they are built from, must contain the following columns:

```
NUMBER
X_IMAGE
Y_IMAGE
A_IMAGE
B_IMAGE
THETA_IMAGE
X_WORLD
Y_WORLD
A_WORLD
B_WORLD
THETA_WORLD
MAG_AUTO
```

If the IOL refers directly to the grism/prism image and not to a direct image taken at a close position (see Chapt. 3.2.3), the values in the columns *_WORLD are ignored. They can be replaced by ###, NaN, -NaN, or +NaN.

To compute a quantitative contamination estimate with the gaussian emission model presented in Chapt. 1.7.2, the wavelength of the magnitude must be known. aXe decodes the wavelength given in a column name if it is given as:

`MAG_?<wavelength [nm]>*`

In this notation ? and * are single and multiple characters, respectively. Valid magnitude column names instead of `MAG_AUTO` are therefore e.g. `MAG_F906LP`, `MAG_R710JOHN` and `MAG_I763SL0AN`. Several magnitude columns with the brightness at various wavelengths improve the emission model and therefore the contamination estimate.

Additional columns are ignored and stripped off when generating the Grism Object List. The actual order of the columns in the Input Object List is not important as long as the header of the file properly describes its content. Blank lines and lines starting with a ';' are ignored.

Care should be taken that each object has a unique number (NUMBER column) assigned to it in an Input Object List. This is the value that will be used throughout the extraction process to identify a particular object. If you use several Input Object Lists in your aXe reduction, make sure that an individual object has the same number in all Input Object Lists. This is important for the combination of spectra extracted from different grism files with `axedrizzle`.

The object numbers must be positive, but do not have to start at a particular value and do not need to be in consecutive order.

With the task `iolprep`, aXe offers a tool to generate the set of Input Object Lists for standard data sets. This aXe task is described in Chapt. 4.1. An example of its use is provided in Chapt. 3.2.3.

7.5 Grism Object List

This file (GOL) is usually generated by aXe using the task `sex2gol`. It has exactly the same format as the Input Object List.

7.6 Aperture File

This Aperture File is an ASCII file describing the APERTUREs in the spectroscopic image. An APERTURE consists of all BEAMS of an object. A BEAM is defined as the group of pixels in the image which will be extracted and combined to produce a final 1-D spectrum. APERTUREs are numbered (e.g. APERTURE 101) using the same numbers that originally appeared in the NUMBER column of the Input Object List. Each APERTURE itself consists of one or more BEAMS (labelled A, B, C etc.). Usually, each object is assigned one aperture in the APER file and each dispersive order is assigned a different BEAM entry inside that aperture definition. In this manner, assuming that the first and second orders are labelled A and B respectively, the 2nd order of object 101 will be found in APERTURE 101, BEAM B. The aperture file is generated by the task `gol2af`.

Each BEAM entry in the APER file contains the following information (data format is indicated in `[]`):

- REFPIXEL## the position in the image of a reference pixel [2*float, x,y]
- CORNERS## the coordinates of a quadrangle defining the region of the image containing the pixel of interest [8*float, x1,y1,x2,y2,x3,y3,x4,y4]
- CURVE## a polynomial description of the dispersion relation of the form $\Delta y = P(\Delta x) = a_0 + a_1 * \Delta x + a_2 * \Delta x^2 + \dots$. Δx^n and $P(\Delta x)$ are the pixel offsets as measured from the coordinates listed in REFPIXEL##. The first number following this keyword is the order of the polynomial. It is followed by (n_1) polynomial parameters [int, (n+1)*float]
- WIDTH## the total number of pixels to extract in the cross dispersion direction [float]
- ORIENT## the orientation, in degrees counter-clockwise and with respect to the x-axis, along which the extraction should proceed [float]
- AWIDTH## the object width as in column A_IMAGE in the Input Object List [float]
- BWIDTH## the object width as in column B_IMAGE in the Input Object List [float]

- AORIENT## the object angle as in column THETA_IMAGE in the Input Object List [float]
- FLUX## the wavelength and the flux values of the object taken from the Input Object List [2n * float, λ_1 , flux₁, λ_2 , flux₂, ... λ_n , flux_n]
- IGNORE## followed by either 0 or 1. If set to 1, this BEAM will not be extracted. [int]

The following example shows one APERTURE containing two BEAMs:

```

APERTURE 2
BEAM A
REFPIXEL2A 27.841 712.643
CORNERS2A 32 714 217 568 208 564 23 710
CURVE2A 1 0.000e+00 -7.893e-01
WIDTH2A 2.811
ORIENT2A 28.083
AWIDTH2A 1.385
BWIDHT2A 0.937
AORIENT2A -61.917
FLUX2A 5.55000e+02 4.10993e-20
IGNORE2A 0
BEAM END
BEAM B
REFPIXEL2B 27.841 712.643
CORNERS2B -122 837 -97 817 -106 812 -131 832
CURVE2B 1 0.000e+00 -7.893e-01
WIDTH2B 2.811
ORIENT2B 28.083
AWIDTH2B 1.385
BWIDHT2B 0.937
AORIENT2B -61.917
FLUX2B 5.55000e+02 4.10993e-20
IGNORE2B 1
BEAM END

```

7.7 Background Estimate File

This file (BEF) is a multiple extension FITS file containing a copy of the input slitless data where the regions defined in an Aperture File have been replaced by estimates of the background (see Chapt. 4.9). This file contains one primary data array in the main extension, named 'SCI', followed by two extensions containing respectively the error array of the Background Estimate (extension 'ERR'), and the Data Quality array of the Background Estimate (extension 'DQ') where bad pixels are flagged by a non-zero value.

This file is generated by the `backest` task.

7.8 Pixel Extraction Table

This file (PET) is a FITS file containing FITS binary table extensions. The primary extension is empty and its header contains information from the header of the original FITS data file from which the PET was generated. Each of these extensions correspond to a single BEAM (as listed in the Aperture File). Each extension can be accessed using its name which is "##" (e.g. "1A" for the first BEAM of APERTURE 1). Each extension contains the information extracted using the task `af2pet` for every pixel contained in the corresponding BEAM. It is in essence a table listing all the pixels in BEAM and some of the values computed for each pixel. A description of the geometry involved can be found in Chapt. 1.

This file is generated by the `af2pet` task (see Chapt. 4.10).

Each extension contains the following columns :

- N, the number of pixels in this BEAM
- P_X, the absolute column coordinate of the pixel
- P_Y, the absolute row coordinate of the pixel
- X, the relative column coordinate of the pixel with respect to the BEAM reference pixel (REFPIXEL## in Aperture File)
- Y, the relative row coordinate of the pixel with respect to the BEAM reference pixel (REFPIXEL## in Aperture File)
- DIST, the projected distance from the center of the pixel to the section point on the trace of the spectrum
- XS, abscissa of the section point relative to the BEAM reference pixel (REFPIXEL## in Aperture File)
- YS, ordinate of the section point relative to the BEAM reference pixel (REFPIXEL## in Aperture File)
- DXS, width of this pixel along the computed trace
- XI, path length of the section point relative to the BEAM reference pixel (REFPIXEL## in Aperture File) along the trace
- LAMBDA, the average wavelength of the light collected by this pixel
- DLAMBDA, the wavelength range of the light collected by this pixel
- COUNT, the number of electron/s in this pixel
- ERROR, the error estimate in electron/s in this pixel
- WEIGHT, the extraction weight assigned to this pixel
- CONTAM, the contamination flag. Set to -1 if no contamination was computed (the task `petcont` was not run) or to the number of BEAMs in which the pixel is included. CONTAM=1 implies that the pixel is a member of exactly one BEAM and therefore not contaminated, while CONTAM=N implies that the pixel is present in N-1 BEAMs, and that contamination may therefore be a problem.

- MODEL, the signal (electron/s) in this pixel according to the quantitative contamination model.
- DQ, the data quality of this pixel.

7.9 The Drizzle Prepare File

This file is a multi-extension FITS file containing the stamp images of all first order beams in a grism image. For each BEAM there are up to five extensions in the DPP-file:

- the data stamp image with the extension name “BEAM_[aperture][beam]”
(e.g. “BEAM_117A”)
- the error stamp image with the extension name “ERR_[aperture][beam]”
(e.g. “ERR_117A”)
- the contamination stamp image with the extension name “CONT_[aperture][beam]”
(e.g. “CONT_117A”)
- the grism model stamp image with the extension name “MOD_[aperture][beam]”
(e.g. “MOD_117A”)
- the variance stamp image with the extension name “VAR_[aperture][beam]”
(e.g. “VAR_117A”)

The Drizzle Prepare File is created by the task `drzprep`. In the task `axedrizzle`, the science, error and contamination images are extracted and drizzled to build for each object the various extensions of a 2D drizzled grism image.

7.10 The 2D Drizzled Grism Image

The 2D drizzled grism images are multi-extension FITS files created in the task `axedrizzle`. There exists one 2D drizzled grism image for every object in the Input Object Lists used to start the aXe reduction. Its name is “[DRZROOT-keyword]_mef_ID[object number].fits” (e.g. `testaxe_mef_ID105.fits`) and reflects the object number used in the Input Object Lists. A 2D drizzled grism image created in `axedrizzle` has the extensions:

- SCI: the science image drizzled from the science extensions of the particular object in all DPP files
- ERR: the error image drizzled from the error extensions of the particular object in all DPP files
- EXPT: the exposure time map for the science extension
- CON: the contamination image drizzled from the contamination extensions of the particular object in all DPP files
- MOD: the grism model drizzled from the model extensions of the particular object in all DPP files

- VAR: the variance image drizzled from the variance extensions of the particular object in all DPP files
- SCIBCK: the drizzled background image (exists only if background drizzling was used)
- ERRBCK: the error of the drizzled background image (exists only if background drizzling was used)
- WHT: the weight image for the science extension

The weight extension is derived from the exposure time map in the task `drz2pet` (see Chapt. 4.15 on how the weights are computed). In `axedrizzle` the task `drz2pet` is used to generates a PET from the set of 2D grism images and to extract 1D spectra for those drizzle-coadded PETs.

7.11 Extracted Spectra File

This file (SPC) is a FITS file containing FITS binary table extensions. The primary extension is empty and its header contains information from the header of the original FITS data file from which the SPC was generated. Each of these extensions correspond to a single BEAM (as listed in the Aperture File). Each extension can be accessed using its name which is "BEAM_##" (e.g. "BEAM_1A" for the first BEAM of APERTURE 1).

This file is generated by the `pet2spc` task.

Each extension contains an extracted, binned, spectrum as produce by the task `pet2spc`. Each extension contains the following columns :

- N, the number of rows in this spectrum
- LAMBDA, wavelength in Å.
- TCOUNT, total number of counts in e s^{-1} in this wavelength bin.
- TERROR, error in the total number of counts in e s^{-1} in this wavelength bin.
- COUNT, background subtracted number of counts in e s^{-1} in this wavelength bin.
- ERROR, error in the background subtracted number of counts in e s^{-1} in this wavelength bin.
- BCOUNT, estimate of the number of counts in electron/s contributed from the background in this wavelength bin.
- BERROR, error in the estimate of the number of counts in e s^{-1} contributed from the background in this wavelength bin.
- FLUX, background subtracted flux in $\text{erg cm}^{-2} \text{s}^{-1} \text{\AA}^{-1}$ in this wavelength bin.
- FERROR, error in the background subtracted flux in $\text{erg cm}^{-2} \text{s}^{-1} \text{\AA}^{-1}$ in this wavelength bin.

- WEIGHT, number of pixels binned together into this wavelength bin.
- CONTAM, for quantitative contamination (see Chapt. 1.7.2) this column gives an estimate on the contaminating flux from other objects to the spectrum. For geometrical contamination the values are set to -1,0,1..n to give the number of source this bin is contaminated with. The value 0 means no contamination. If the contamination was not recovered, every bin has the value -1.
- DQ, the propagated data quality at this wavelength. This is computed by simply summing all the individual DQ values from the pixels contributing to this wavelength.

7.12 Stamp Image File

This file (STP) is a multi-extension FITS file containing stamp images of the BEAMs that were extracted. The primary extension of this file is empty. Each following extension contains the image of a single extracted BEAM. Extensions are named "BEAM_[aperture][beam]" (e.g. BEAM_1A). The STP-file is similar to the drizzled grism image explained in Chapt. 7.10. The STP-file contains all stamp images in a slitless image, and the drizzled grism image the stamp image of one object as combined from all slitless images. Also the drizzled grism images all have a WCS in wavelength and spatial direction.

7.13 Weight Image File

The weight image file is produced in the optimal weighted extraction of spectra from individual grism images. Its file name extension is `*.opt.WHT.fits`, corresponding to the `*.opt.SPC.fits` of the optimal extracted spectra file.

The weight image file is a multi extension fits file which contains images of the weights applied to the PET pixels during the optimal weighted extraction. The weights are computed according to Eqn. 1.1 in Chapt. 1.9. The weight images have trace length and trace distance as x- and y-axes, respectively. The extensions are named "WHT_[aperture][beam]" (e.g. WHT_1A).

7.14 Contamination File

This file (CONT) is a simple FITS image containing the contamination estimate computed by the `petcont` task.

If quantitative contamination (see Chapt. 1.7.2) was deployed, the contamination image contains the sum of all modelled beams. It is therefore a complete, quantitative model of the corresponding grism/prism image. Blinking the contamination image against the slitless image is a very good method to check whether the contamination estimates are reasonable. The contamination image should have pixel values comparable to the background subtracted slitless image.

In geometrical contamination, pixels which are not within any known beams are assigned a value of 0. Pixels which are within a single beam (i.e. not contaminated by higher spectral orders and/or other objects in the field) are assigned a value of 1. Pixels contaminated by n beams are given a value of n+1.

Appendix A

Virtual slits

The slit length sl , the slit width sw and the orientation so of the *virtual slit*, optimized to maintain the spectral resolution in slitless spectroscopy, is computed from the major axis size `a_img`, the minor axis size `b_img` and the major axis angle `theta_img` for each object (the three quantities are given in the SExtractor catalogue columns `A_IMAGE`, `B_IMAGE` and `THETA_IMAGE`).

Without loss of generality we assume that the dispersion direction is parallel to the x-axis, which means the angle between the dispersion direction and the major axis angle (defined with respect to the x-axis) is `theta_img`. It is then:

$$A_{11} = (\cos(\text{theta_img})/\text{a_img})^2 + (\sin(\text{theta_img})/\text{b_img})^2 \quad (\text{A.1})$$

$$A_{12} = \cos(\text{theta_img}) * \sin(\text{theta_img}) * (1.0/\text{a_img}^2 - 1.0/\text{b_img}^2) \quad (\text{A.2})$$

$$A_{22} = (\sin(\text{theta_img})/\text{a_img})^2 + (\cos(\text{theta_img})/\text{b_img})^2 \quad (\text{A.3})$$

$$\alpha = \arctan(A_{12}/A_{11}) \quad (\text{A.4})$$

$$sl = \sqrt{A_{11}} * \text{a_img} * \text{b_img} / \cos(\alpha) \quad (\text{A.5})$$

$$sw = 1.0 / \sqrt{A_{11}} \quad (\text{A.6})$$

$$so = \alpha + 90.0 \quad (\text{A.7})$$

For further details see Freudling et al. 2008 ([2]).

Bibliography

- [1] Bertin, E. & Arnouts, S., 1996, *SExtractor: Software for source extraction*, A&AS 117, 393
- [2] Freudling, W., Kümmel, M., Haase, J. et al. 2008, *The Hubble Legacy Archive NICMOS grism data*, A&A, 490, 1165
- [3] Furchter, A. & Hook, R., 2002, *Drizzle: A method for the linear reconstruction of undersampled images*, PASP, 114, 144
- [4] Horne, K., 1986, *An optimal extraction algorithm for CCD spectroscopy*, PASP, 98, 609
- [5] Koekemoer, A.M., Fruchter, A.S., Hook, R.N., Hack W., & Hanley, C., 2006, *Multidrizzle and Tweakshifts: Overview and Future Plans*, The 2005 Calibration Workshop, eds. A. Koekemoer, P. Goudfrooij, & L. Dressel, L., 423
- [6] Kümmel, M., Walsh, J.R., Pirzkal, N., Kuntschner, H. & Pasquali, A., 2009, *The Slitless Spectroscopy Data Extraction Software aXe*, PASP 121, 59
- [7] Kümmel, M., Kuntschner, H., Walsh, J.R., Lombardi, M., Stoehr, F., Haase, J.; Hook, R.N., Rosati, P., Micol, A., Fosbury, R., Freudling, W., 2009, *The ST-ECF ACS Grism Hubble Legacy Archive Project*, in Bohlander, D., Durand, D., Dowler, P. (eds.) *Astronomical Data Analysis Software and Systems XVIII*, ASP Conference Series, Vol. 411, p. 430
- [8] Kümmel, M., Larsen, S. & Walsh, J., 2005, *New developments in aXe*, ST-ECF Newsletter 37, 8
- [9] Kümmel, M., Walsh, J., Larsen, S. & Hook, R., 2004a, *Drizzling and the aXe slitless spectra extraction software*, ST-ECF Newsletter 36, 10
- [10] Kümmel, M., Walsh, J., Larsen, S. & Hook, R., 2004b, *The release of aXe 1.4*, ST-ECF Newsletter 37, 14
- [11] H. Kuntschner, M. Kümmel & J. R. Walsh, 2008, *Updated flux calibration and fringe modelling for the ACS/WFC G800L grism*, ISR ACS 2008-01
- [12] Larsen, S.S. 2006, *Wavelength and Flux calibration of the ACS/SBC PR110L and PR130L prisms*, ISR ACS 2006-02

- [13] Larsen, S.S., Walsh, J.R. & Kümmel, M., 2006, *Wavelength and Flux calibration of the ACS/HRC PR200L prism*, ISR ACS 2006-03
- [14] Pirzkal, N., Xu, C., Malhotra, S., Rhoads, J. E., Koekemoer, A. M., Moustakas, L. A., Walsh, J. R., Windhorst, R. A., Daddi, E., Cimatti, A., Ferguson, H. C., Gardner, Jonathan P., Gronwall, C., Haiman, Z., Kmmel, M., Panagia, N., Pasquali, A., Stiavelli, M., di Serego Alighieri, S., Tsvetanov, Z., Vernet, J., Yan, H., 2004, *GRAPES, Grism Spectroscopy of the Hubble Ultra Deep Field: Description and Data Reduction*, ApJS 154, p. 501
- [15] Pirzkal, N., Burgasser, A.J., Malhotra, S., Holwerda, B.W., Sahu, K.C., Rhoads, J.E., Xu, C., Bochanski, J.J., Walsh, J.R., Windhorst, R.A., Hathi, N.P.; Cohen, S.H., 2009, *Spectrophotometrically Identified Stars in the PEARS-N and PEARS-S Fields*, ApJ 695, 1591
- [16] Robertson, J.G., 1986, *Optimal extraction of single-object spectra from observations with two-dimensional detectors*, PASP 98, 1220
- [17] Rodriguez-Pascual, P., Gonzalez-Riestra, R., Schartel, N. & Wamsteker, W., 1999, *The IUE INES System: Improved data extraction procedures for IUE*, A&AS 139,183
- [18] van Dokkum, P.G., Brammer, G., 2010, *Hubble Space Telescope WFC3 Grism Spectroscopy and Imaging of a Growing Compact Galaxy at $z = 1.9$* , ApJ 718, L73
- [19] Walsh, J. & Kümmel, 2004, *Comfortable visualisation of ACS spectra*, ST-ECF Newsletter 35, 9

Index

- ACS, 9, 73
- AF2PET, 62, 87
- APERTURE, 11
- Aperture File, 59, 74, 85, 87, 89
- AWIDTH, 85
- aXe support, 30
- aXe tasks, 47, 71, 72, 79
- aXe test, 30
- aXe visualization, 26
- aXe webpage, 27
- aXe2web, 26
- AXECORE, 51
- AXEDRIZZLE, 55
- axedrizzle, 49, 55
- AXEGPS, 68
- AXEPREP, 15, 49
- BACKEST, 60
- Background, 59
- Background Estimate File, 59–62, 86
- Background Pixel Extraction Table, 15, 62
- BCOUNT, 89
- BEAM, 11, 59, 61, 66, 72, 74–77
- BERROR, 89
- binned spectrum, 65, 89
- BWIDTH, 85, 86
- calibrated spectrum, 14
- CAMERA, 72
- CCDTYPE, 73
- CFITSIO, 9, 81
- compilation, 29
- configuration file, 9, 38, 47, 59, 65, 71, 72, 79
- Configuration of aXe tasks, 71
- CONTAM, 87, 90
- contamination, 15
- contamination file, 90
- CORNERS, 85
- COUNT, 87, 89
- CURVE, 85
- direct image, 47, 59, 72, 77
- DISP_ORDER, 77
- Dispersion Order, 74
- DIST, 87
- DLAMBDA, 87
- DLD1P, 77
- DLD1P_RANGE, 77
- DLDP, 77
- dmag, 59
- DQ, 88, 90
- DQ_EXT, 72
- DQMASK, 72
- drizzle, 20, 55, 74
- Drizzle PrePare files, 20, 54
- drizzled stamp images, 66
- DRZ2PET, 67
- DRZKERNEL, 74
- DRZLAMB0, 74
- DRZPFRAC, 74
- DRZPREP, 20, 54
- DRZRESOLA, 55, 66, 73
- DRZROOT, 74
- DRZSCALE, 55, 74
- DRZXINI, 74
- DXS, 87
- DYDX, 76
- DYDX_ORDER, 76
- emission model, 17, 63
- environment variables, 47, 71
- ERROR, 87, 89
- ERRORS_EXT, 72
- Example, 59, 60, 62–67
- exposure time, 72, 73
- EXPTIME, 73
- Extracted Spectra File, 13, 65, 89
- extracting spectra, 36
- extraction parameters, 24
- FCUBEPREP, 48

FERROR, 89
 FFNAME, 64, 73
 field dependence, 76, 77
 flat field, 64, 79
 FLUX, 89
 flux conversion, 26
 fluxcube, 48
 fluxcube file, 82
 Fluxcube Model, 17
 Gaussian Emission Model, 17
 geometrical contamination, 15, 63
 GOL2AF, 59
 Grism Object List, 58, 59, 85
 grism wavelength calibration, 77
 GSL, 9
 High Level Tasks, 9, 36, 83
 HST, 9
 Hubble Ultra Deep Field, 28
 IGNORE, 86
 input image, 81
 Input Image List, 37, 83
 Input Object List, 10, 37, 47, 58, 84
 INSTRUMENT, 72
 interpolation, 61
 IOLPREP, 47
 LAMBDA, 87, 89
 Linux, 9
 Low Level Tasks, 9
 magnitude cutoffs, 75
 master catalogue, 33
 master sky image, 14
 MMAG_EXTRACT, 75
 MMAG_MARK, 75
 MODEL, 88
 MultiDrizzle, 15, 32
 multiple extensions, 39
 N, 87, 89
 Note on aXe wavelength dependent flat-fielding, 64
 Note on Field Dependent Values, 76
 Note on How to set up your own BEAM description, 78
 object orientation, 11
 object position, 77
 online parameter, 61, 71
 optimal weighting, 22, 65, 67, 90
 OPTKEY1, 73
 OPTVAL1, 73
 ORIENT, 85
 P_X, 87
 P_Y, 87
 PET2SPC, 65, 89
 PETCONT, 63, 90
 PETFF, 64
 Pixel Extraction Table, 12, 13, 62–64, 66, 87
 pixel weight, 25
 POBJSIZE, 73
 preamble, 7
 preview webpages, 28
 prism wavelength calibration, 77
 PSF_OFFSET, 73
 PSFCOEFFS, 73
 PSFRANGE, 73
 PyRAF, 29
 Python, 9
 quadrangle, 11, 85
 quantitative contamination, 16, 63
 RDNOISE, 73
 reduction strategy, 36
 reference pixel, 10, 74, 76, 77
 REFPIXEL, 85
 REFX, 73
 REFY, 73
 SCIENCE_EXT, 72
 section point, 12
 segmentation image, 35
 segmentation map, 48
 SENSITIVITY, 78
 sensitivity curve, 13, 74, 78, 79
 SEX2GOL, 58, 81, 85
 SExtractor, 48
 sky background, 14
 slitless image, 47, 59, 60, 62, 72
 slitless spectroscopy, 10
 SMFACTOR, 73
 Solaris, 9
 spectral trace, 11, 13, 59, 75–77
 Stamp Image File, 66, 90
 STAMPS, 66

STSDAS, 9, 29
STSDAS download, 29
support, 7

TCOUNT, 89
TERROR, 89
trace distance, 12
tuning aXe, 43

virtual slit, 26, 91

wavelength bins, 13, 89, 90
wavelength calibration, 59, 77–79
WCS, 58
WCSLIB, 9
WEIGHT, 87, 90
weight image file, 90
WFC3, 9, 21
WIDTH, 85

X, 87
XI, 87
XOFF, 76
XS, 87

Y, 87
YOFF, 77
YS, 87