
The Swift UVOT Grism Documentation

Release 2.0.3

Paul Kuin

December 19, 2014

CONTENTS

Here you can find the description of the *Swift* UVOT grism calibration in much detail as well as the latest version of the software (in Python) and the grism calibration files.

INTRODUCTION TO THE UVOT GRISMS

- [UVOT description \(MSSL UVOT website\)](#)
- [The UVOT filter wheel \(MSSL UVOT website\)](#)

1.1 The UVOT MIC image intensifier

The detector is a Microchannel plate (MCP) Intensified Charge coupled device (CCD) or MIC [Fordham et al.](#) . Each photon incident on the S20 multi-alkali photocathode can release an electron which is amplified a million-fold using a three stage MCP. The cloud of electrons excites photons in a P46 fast-phosphor screen which are fed through a fibre taper to a CCD operated in frame transfer mode. The exposed area corresponds to 256x256 CCD pixels, but after readout the photon splash is centroided to 8 times higher resolution, providing an effective image that is 2048x2048 pixels square. The nature of the centroiding process is such that the effective size each of the 8x8 sub-pixels on the sky is not exactly the same, leaving a modulo-8 (MOD-8) pattern in the untreated image which can be corrected for in data processing. However information loss that occurs when more than one photon splash is registered on a CCD pixel within the same CCD readout interval (coincidence loss) can cause some pattern to remain after correction for bright sources.

As noted above, the finite time over which each exposure is integrated on the CCD, the *frame time*, results in coincidence losses if the photon arrival rate is high enough. Statistically, there is a chance that multiple photons arrive within one CCD frame with spatially overlapping pulse profiles, in which case only one arrival will be recorded. This means that fewer source photons are detected than are incident on the detector, resulting in a non-linear response with source brightness. Making use of the statistical nature of the effect, the coincidence loss can be corrected, and an expression for point sources has proved very effective in UVOT photometry, e.g., [Poole et al.](#) , [Breeveld et al.](#) . Extremely bright sources, above the brightness limit for coincidence loss correction, suffer a further loss due to interference from events registered in neighbouring CCD pixels.

The background due to dark current in the detector is very low; instead the sky background is the limiting factor for faint sources. The sky background in the grisms is comparable to that in the UVOT white (clear) filter since both grisms transmit the $2800\text{--}6800\text{\AA}$ optical band, as well as shorter wavelengths in the case of the UV grism.

1.2 Description of the grisms

The UVOT uses a modified Ritchey-Chrétien optical design where light from the telescope is directed towards one of two redundant detectors using a 45-degree mirror. A filter wheel allows selection of either a UV or optical lenticular filter, a “white”/clear filter, a UV grism, a visible grism, or a blocked position. Behind the filter wheel is an image intensifier configured to detect each photon event with a 2048x2048 pixel resolution.

The *Swift* UVOT grisms were originally the flight spares for the *XMM OM* instrument. The grisms for both instruments were designed using a *zemax* optical model. The *Swift* UVOT instrument design and build procedure was modified to avoid the molecular contamination which impaired the *XMM OM* UV sensitivity. Therefore the sensitivity of the

UVOT grism is much better in the UV than in the *XMM OM* grisms. The UVOT visible grism optics were blazed at 3600 Angstrom. However, the UVOT UV grism optics were not blazed to obtain better UV throughput; therefore the second order spectrum of this grism is significant and has to be accounted for in the analysis when orders overlap.

Each of the two grisms can be operated in two modes. The so-called “nominal” mode is where the filter wheel is rotated so that the grism is positioned in direct alignment with the telescope optical light path. However, in order to reduce the contamination by zeroth order emission of the background and field sources, in the so-called “clocked mode” the filter wheel is turned so the grism is partially covered by the telescope exit aperture which restricts the field of view, blocking some field stars and reducing the background light. In effect this mimics the properties of a slit spectrograph with a broad ‘slit’. In the clocked mode the first order spectra of many stars in the field of view lie in the area uncontaminated by background or the zeroth order spectra of other field stars, see figure *figure 1: visible grism image*.

1.2.1 figure 1: visible grism image

The clocked mode has proven to be the preferred mode of observation. A comparison of clocked and nominal images in figure *figure 2: UV grism (clocked mode) image* shows how effective the reduction of zeroth order contamination is from other sources in the field for those spectra falling on the left hand side of the image.

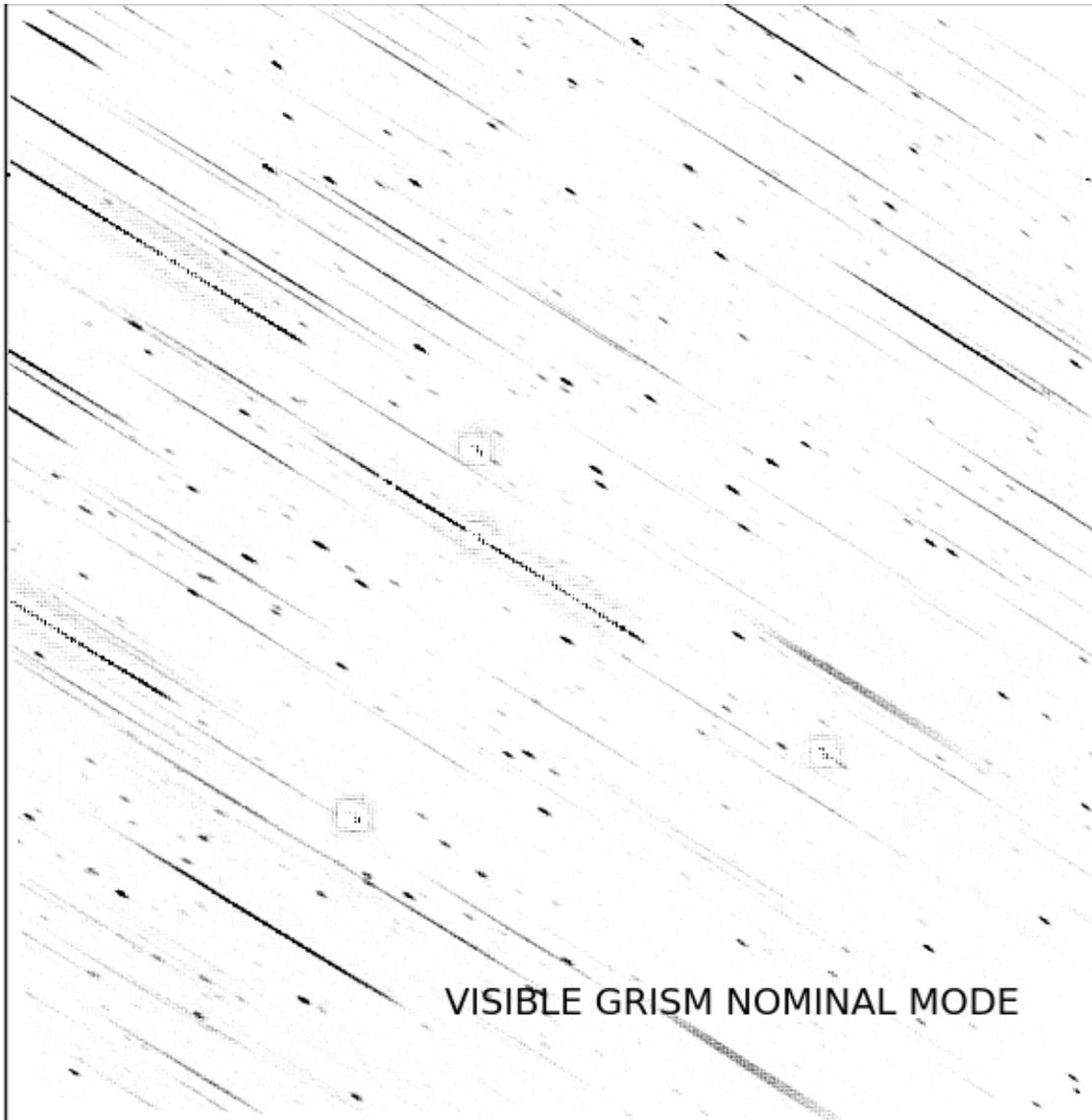
1.2.2 figure 2: UV grism (clocked mode) image

The “grisms” are actually made up of two optical elements: a grism and a tilt compensator, see figure *figure 3: Diagram with rays*. The grism provides on-axis dispersion by means of a prism and transmission grating, but the focal plane is tilted. The tilt compensator element flattens the focal plane. The UV grism is direct ruled on a *Suprasil* substrate, while the visible grism is replicated *epoxy*. The grating in the visible grism is blazed to maximise the transmission of the first order. The UVg grating was not blazed. Because the optics are in the converging beam of the telescope, which uncompensated would induce a shift in focus, the leading surface is slightly convex. The dispersed light is refocussed before leaving the grism. The grism design was optimised for the UV grism in the first order around 2600 Angstrom, and in the visible grism around 4200 Angstrom.

1.2.3 figure 3: Diagram with rays

1.3 Summary of the Swift UVOT grism properties

The Ultraviolet(UV) and Optical Telescope on board the *Swift* mission provides images of a field 17x17’ large on the sky, with a spatial resolution of 0.5” in seven filters, two grisms, and a magnifier. The information below documents the latest information of the UVOT Grisms for the redundant optics which have been used since start of operations. The magnifier is out of focus in the operational configuration and has not been used. The grisms are mounted on a filterwheel and can be used in two modes called ‘nominal’ and ‘clocked’ as explained elsewhere.



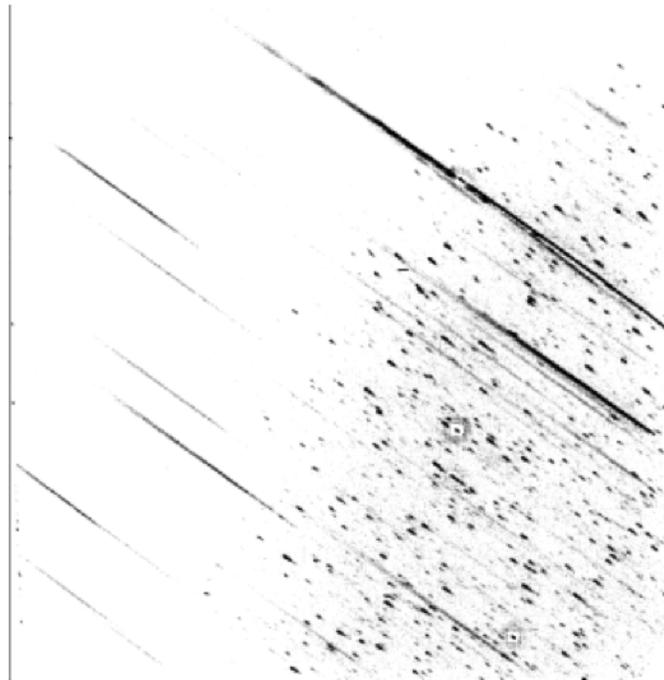
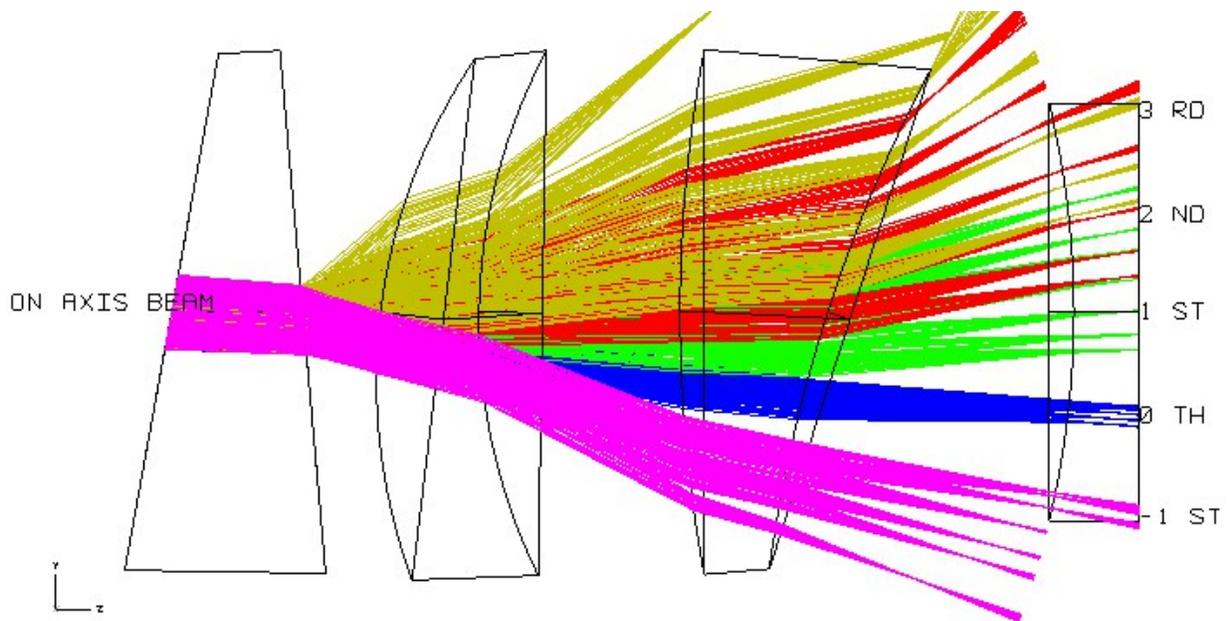


Figure 1.2: As in figure 1, but for the uv grism.



1.3.1 Summary of capabilities

Visual grism properties	values
Spectral resolution	100 at 4000A
Effective wavelength range (first order)	2900-6600A
wavelength accuracy	12A
order overlap second order to first	5700A in first order
effective magnitude range	13-17 V mag
astrometric accuracy (first order anchor point)	1.2 arcsec
scale	0.57 arcsec/pixel
dispersion	5.9 A/pixel at 4200A
coincidence loss above (depends on background)	$\sim 10^{-13}$ erg/cm ² /s/A
zeroth order magnitude zero-point	17.7 mag

UV grism properties	values
Spectral resolution	75 at 2600A
Effective wavelength range (first order)	1730-6700A
Effective wavelength range (second order)	1700-2250A
Effective wavelength range (zeroth order)	3600-5500A
wavelength accuracy	22A
order overlap second order to first	>2750A in first order
effective magnitude range	11.5-15.5 V mag
astrometric accuracy (first order anchor point)	3.5 arcsec
scale	0.57 arcsec/pixel
dispersion	3.2 A/pixel at 2600A
coincidence loss above (depends on background)	$\sim 5 \times 10^{-13}$ erg/cm ² /s/A
zeroth order magnitude zero-point	19.0 mag

The magnitude zero point given here for the zero order is the isophotal magnitude.

Some example data:

Raw Images: UV image and grism of the same field.

Raw images: V image and Visual grism of the same field.

Most noticeable are the differences between the nominal and clocked mode.

The small dotted streaks are zeroth orders of sources. In the clocked mode the zeroth orders are confined to only part of the detector image. In addition the background is also more prominent where the zeroth orders are. Since in long exposures the background can drive up the effect of coincidence loss, and in bright fields sometimes avoid data loss from buffer overflows, the clocked mode is usually the best for observing.

More difficult to notice is that in the UV grism some of the spectra are slightly curved and a partial separation of the second order from the first order in the top right and bottom left corners of the image. In the V grism part of the second order is sometimes found as an extension of the first order. UV bright sources in the UV grism can show a separated UV tail to the zeroth order.

Some first orders on the detector have no accompanying zero order. In that case the zero order did fall outside the detector. In fact, because the gratings were optimised for the first order, the first orders fall mostly on the detector, but part of the zero orders do not. In rare cases, a minus one order is present in the UV grism on the right hand side of the image as a thick streak.

If a source is bright, the zeroth order will present a large halo, with a modulo-8 pattern. That is due to the combined effect of the coincidence loss of the detector and the on-board detection algorithm. If a source is very very bright, the first order may also degenerate into a mod-8 pattern and nearly all spectral and flux information will be lost.

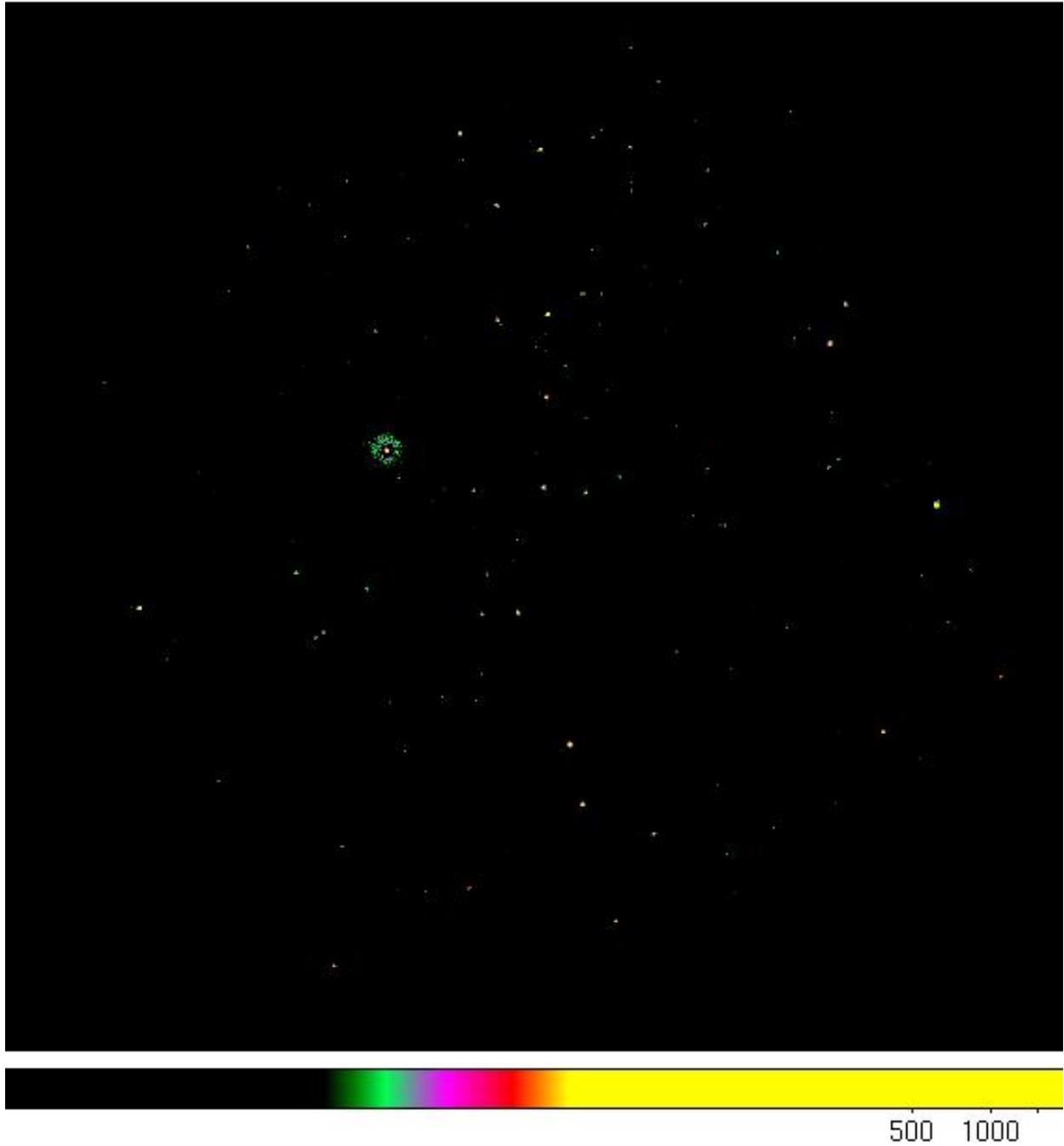


Figure 1.3: The field of which the uv grism is taken.

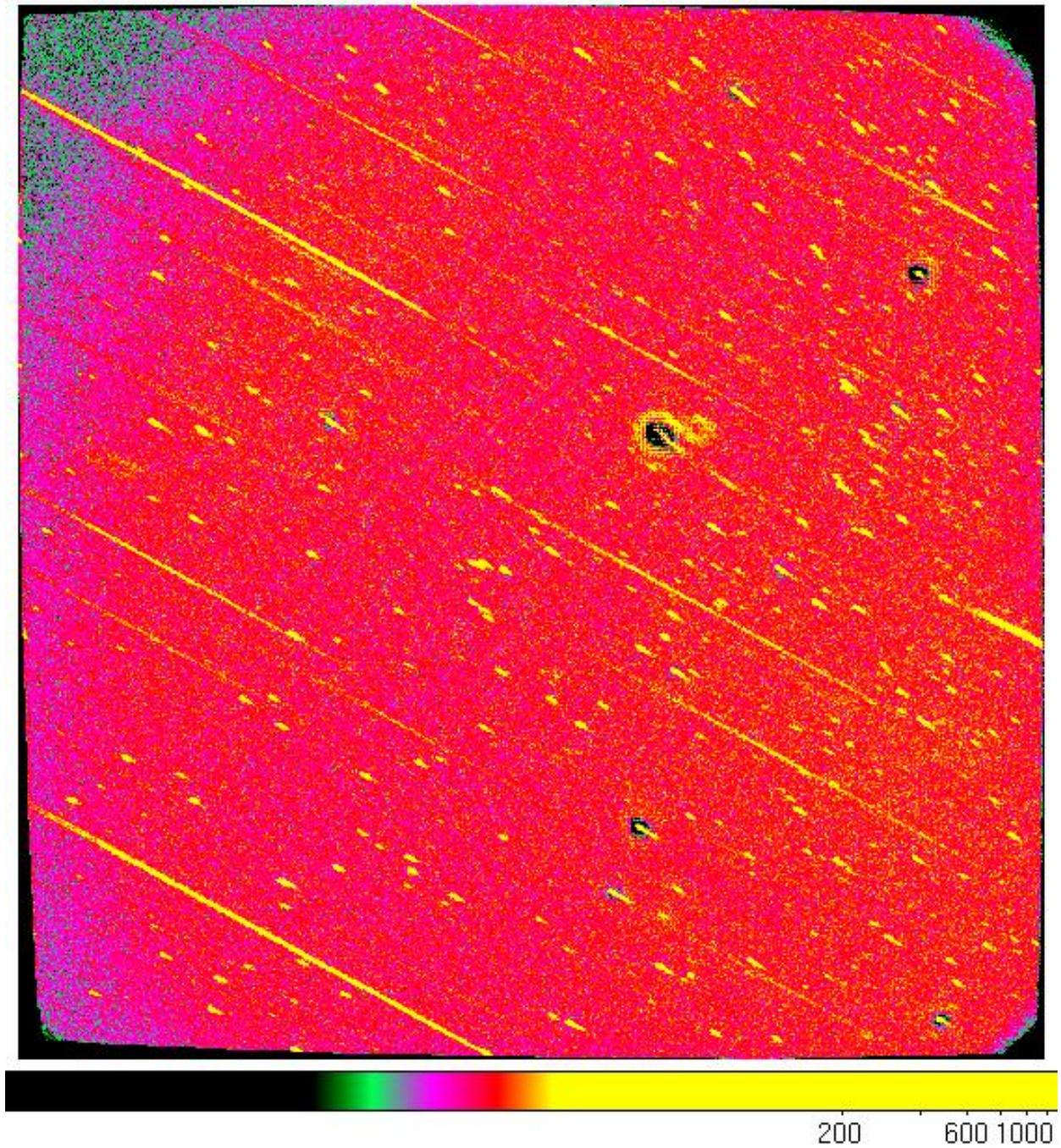


Figure 1.4: This is the grism in the nominal mode.

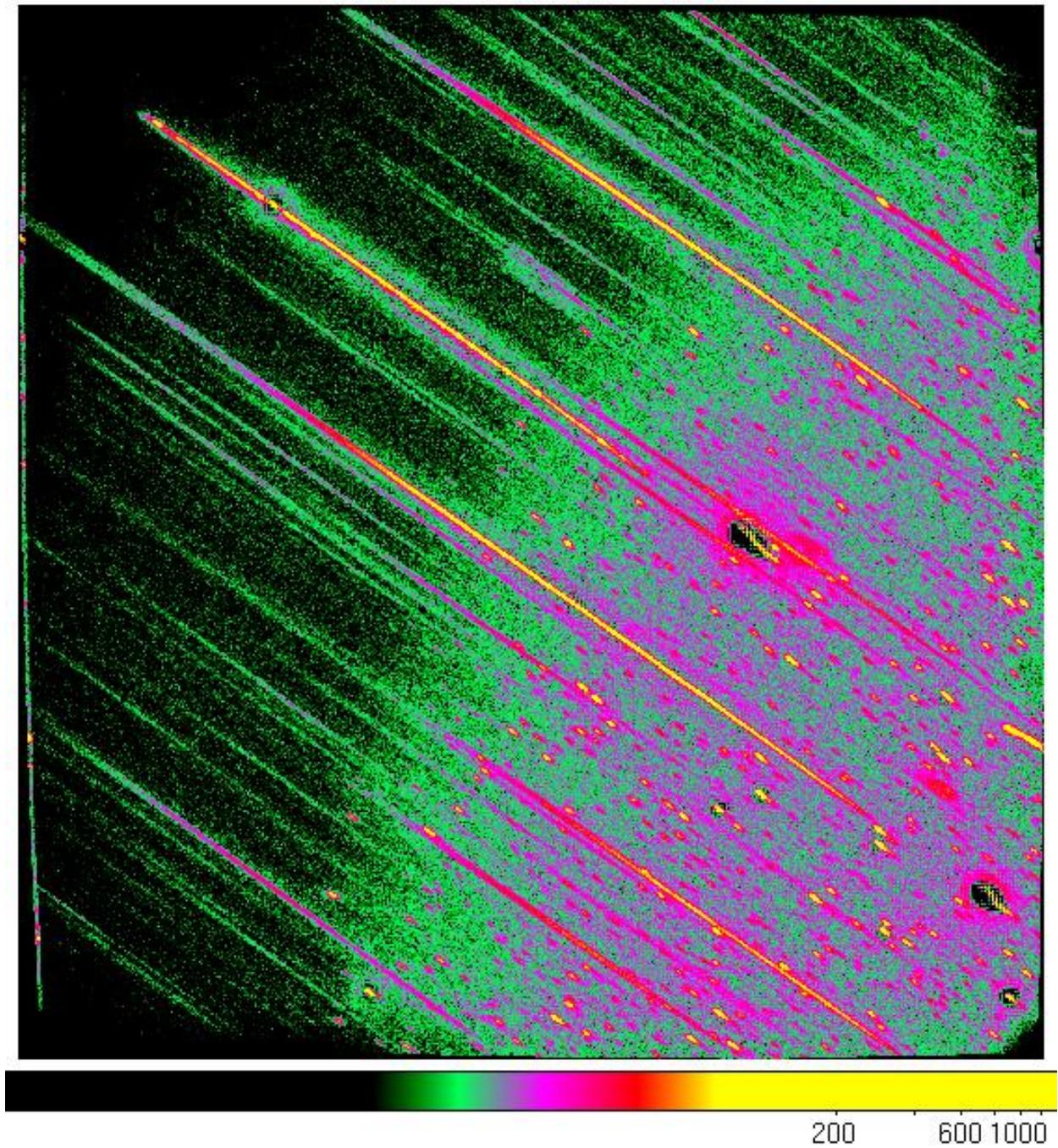


Figure 1.5: This is the grism in the clocked mode.

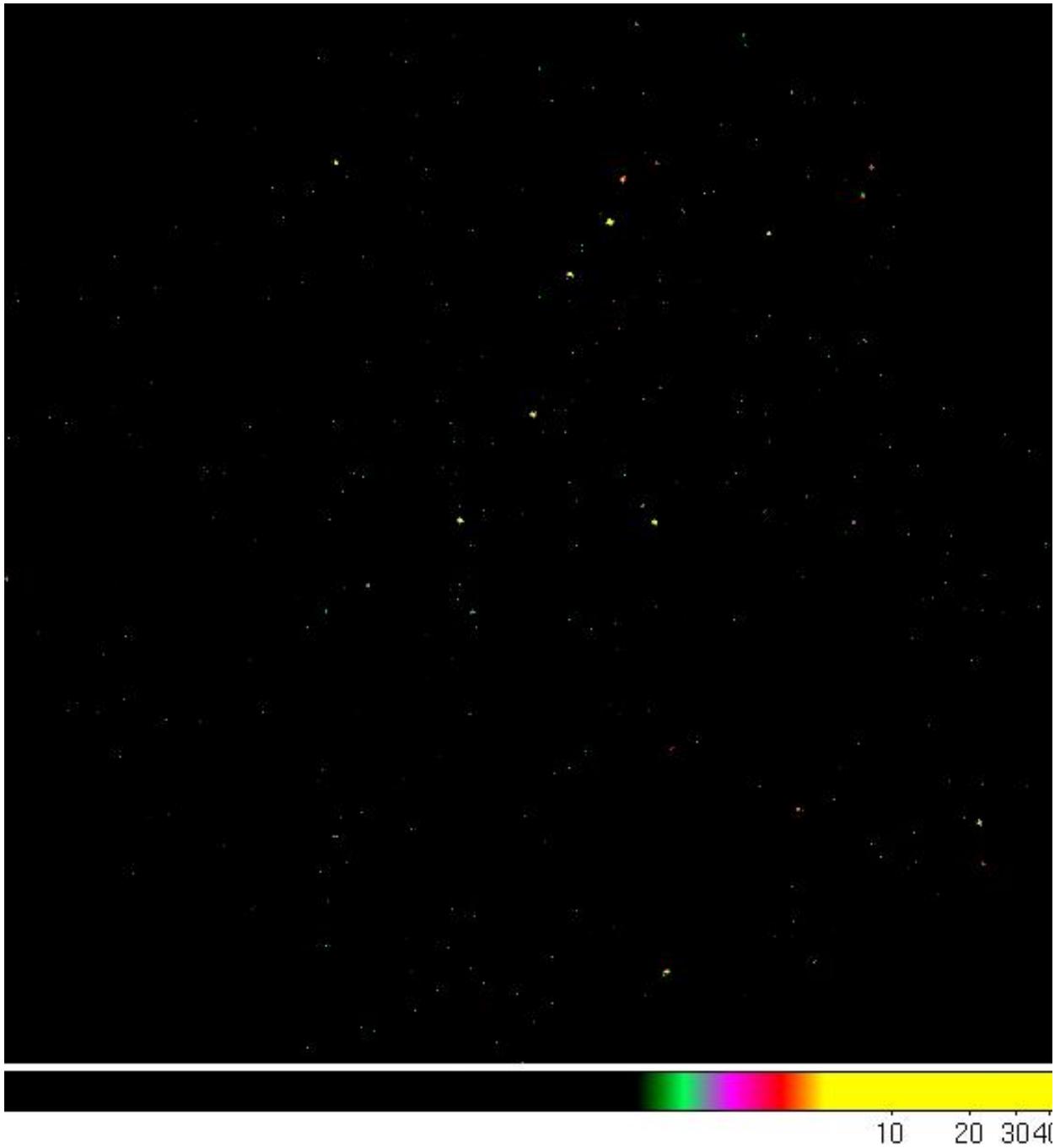
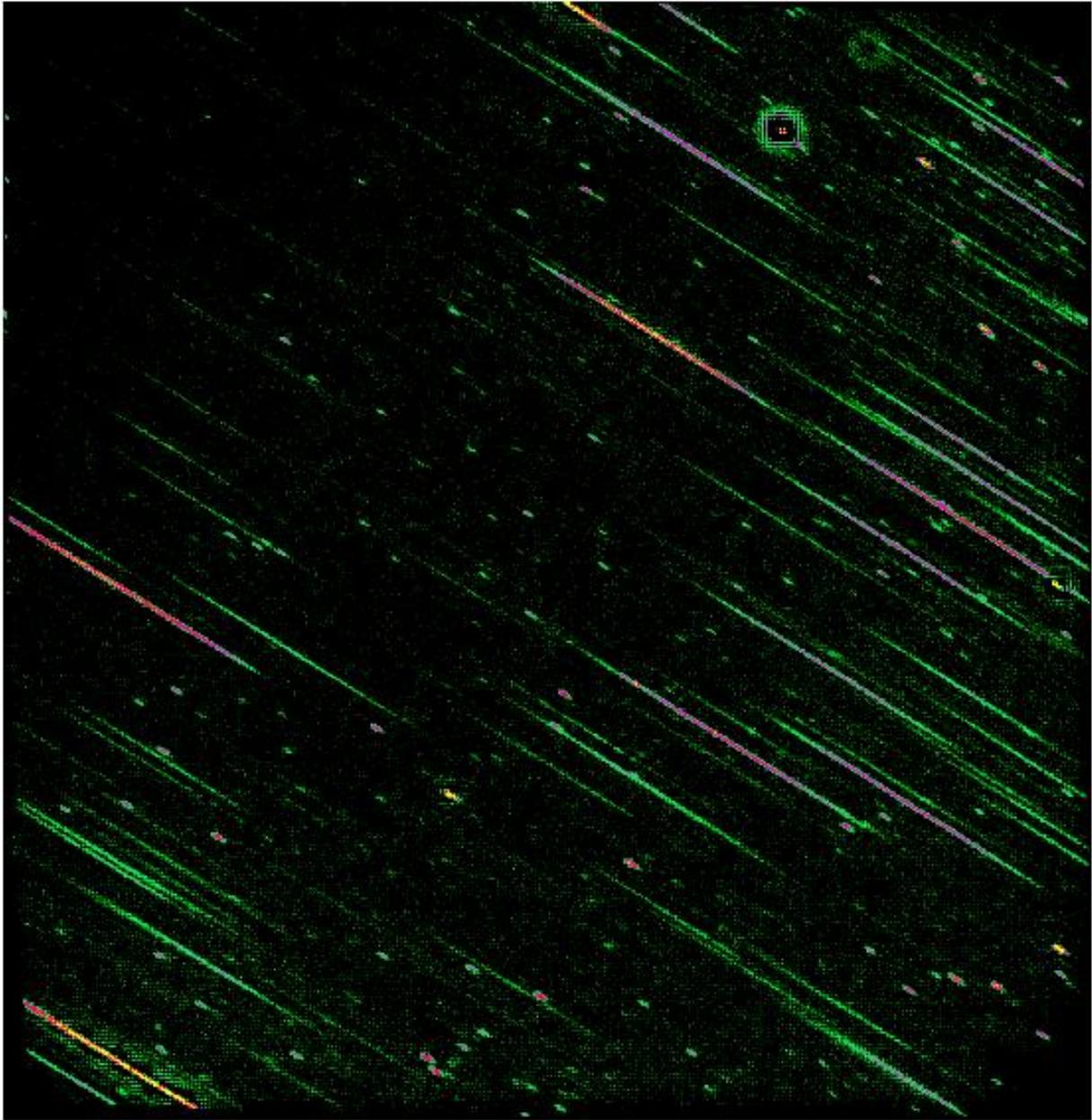


Figure 1.6: This is the field of which the visual grism image was taken.



2000 4000 8000

Figure 1.7: This is the visual grism image in the nominal mode.

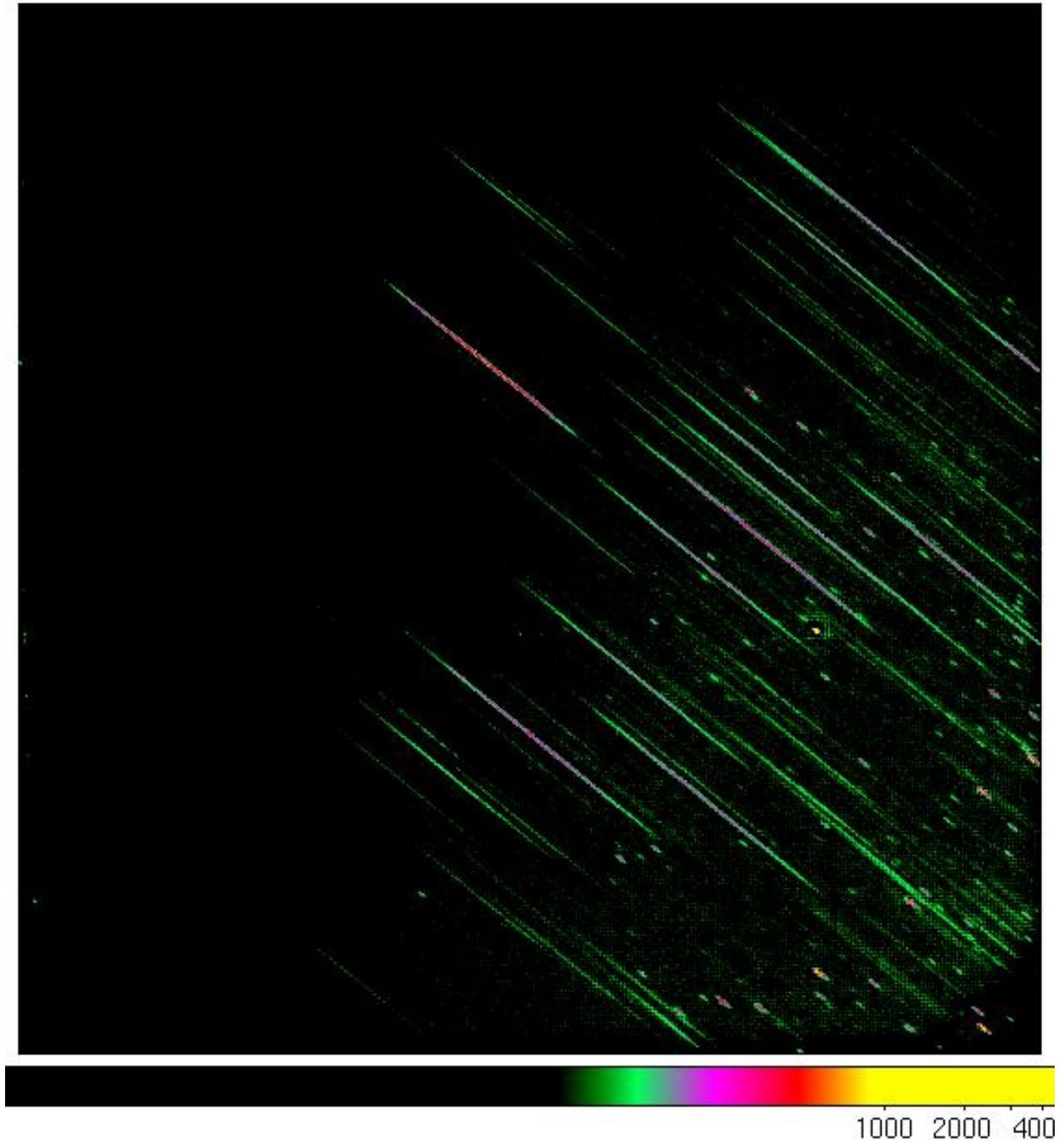


Figure 1.8: This is the visual grism image in the clocked mode.

1.4 order layout

The spectrum of any object in the field of view of the telescope will be dispersed. For each *order* of dispersion, a separate spectrum appears on the detector. Each order also has a unique sensitivity which depends on the wavelength. There is also some variation of all that over the face of the detector, meaning that a spectrum in the centre will be slightly differently dispersed and have a different response than a spectrum near the bottom or the top, etc., of the detector.

The response for the uv grism, which was not blazed, is different than that for the visual grism, which was. Read for example about blazed gratings in [Appenzellers book](#).

1.4.1 The uv-grism

The design goal for the uv-grism was to optimize the response in the 2000-3400Å region. The design was made for the centre of the detector, since that is where the main target would be located during operations.

The zeroth order in the final design is not a point, but dispersed. While most of the optical light falls on about three pixels, the zeroth order of the uv-grism has an extended tail with a dispersion proportional to the inverse of the wavelength. For most sources the uv tail is hidden by the background noise. However, uv-bright sources, like White Dwarfs, Gamma Ray Bursts, and such show an extended zeroth order tail, often to appear disconnected from the visual part due to the reduced flux around the 2200Å region due to the interstellar absorption.

The zeroth order main peak is separated from both the first and minus-first orders by about 300 pixels in the uv-grism. The minus-first order has a terrible point spread function, and therefore is not useful for science.

The second and third order wholly or partially overlay the first order. The part of the first order completely free from second order contamination varies over the detector. In the bottom half of the detector it is fair to say that the overlay starts in the first order frame of reference around 2750Å, but if the source is not very blue the limit is more like 2950Å. For a red source, it the wavelength at which the second order contamination rises above the background noise can be a larger wavelength, of course.

In the upper half of the detector the uv-part of the second order curves away from the first order, and also shows some lateral displacement. There, even for blue sources the second order will not contaminate the first order until higher wavelengths are reached. The best result is for an offset of about +3.5 arcmin in X, and Y on the detector, and the region up to about 4500Å shows second and first order emission not overlapping.

The second order sensitivity peaks further in the uv than that of the first order which peaks around 2700Å. With also the width of the spectral track of the first order being larger than the first, i.e., 4.5 versus 3.1 pixels, the signal per pixel for the second order is generally much lower, with the exception of bright spectral lines.

The third order overlap starts close to 4900Å of the first order spectrum. If the source has a blue spectrum, there is sometimes excess emission causing a jump in the flux. However, for most sources the third order flux is small being spread over about 6 pixels wide, and in the top part of the detector the offset is usually large enough to make it irrelevant for the first order spectrum.

1.4.2 The visual grism

In the visual grism, the second order is much fainter than the first since the grism was blazed. The first second, third, etc. orders all fall on one line, thus partially overlap. Like the uv grism, the zeroth order lies apart. The dispersion is about half that of the uv-grism, 6Å/pixel in the visible grism versus 3Å/pixel in the uv-grism. The response below 3000Å (3000-2700Å) is not very good, so it covers a slightly bluer region than ground based spectroscopy achieves.

- The grism point Spread Function
- Attitude drift during observations

1.5 Calibration

1.5.1 The wavelength calibration of the UVOT Grisms

History of the wavelength calibration

From the grism design and ground calibration the dispersion was determined for the centre of the detector. This was further refined post launch and implemented in the Swift UVOT “ftool” “uvotimgrism” and related calibration files. The anchor for this calibration is the position of the zeroth order peak (which after April 2008 has been derived by an aspect solution using the weak zeroth orders).

The 2009 wavelength calibration, which instead is using an anchor defined in the first order, is valid for the *whole* detector. The current calibration is that one with some minor improvements. In the following pages some detailed information dates back to 2009. Any updates mostly concern the determination of the accuracy of the anchor which is indicated in the relevant text.

Calibration approach

The wavelength calibration relies on two parts. One is the mapping of the sky position of the target to the grism detector image which is used to define an *anchor* position at a default wavelength, the other is determining the dispersion in relation to the anchor position.

The anchor position selected in the initial calibration was the peak emission of the zeroth order. The current calibration uses the 2600Å position for the uv grism, and the 4200Å position for the visible grism in the first order. A mapping from the source position in a lenticular filter (i.e., the *uv*, *uvb*, *uvu*, *uvw1*, *uvm2*, or *uvw2* filters) relative to its bore-sight to that in the grism image has been determined. Recent observations with the *Swift* UVOT grism have by default an observation right before or after the grism exposure in one of the lenticular filters to assist in that mapping. Older grism observations can be aspect corrected using the distortion-corrected zeroth orders. Using the corrected World Coordinate System (S) then allows also a determination of the first order anchor position.

The dispersion varies over the detector. Remarkably, in the uv grism the cross-dispersion location of the orders relative to each other depends also on the location of the spectrum on the detector. In the default observing position at the centre of the detector, the second and third order partially overlap with the first order. Unfortunately, in the uv grism the second order is still quite strong compared to the first order, especially in the blue which for spectra from uv-bright sources can cause overlap from 2750Å in first order. Away from the centre, near the top of the detector the second order uv section curves away from the first order, and also has a small displacement. This means that for that part of the detector, for sources with known bright emission line spectra, a second order wavelength scale could be determined.

The anchor determination and dispersion calibration accuracy is limited since we are not able to have a dense coverage of the detector with calibration observations. Our approach has been to use the optical model used for the instrument design to aid us in extending the calibration over the whole detector. This “Zemax” model was originally developed for the design of the *XMM* “OM” instrument, it was used again for the *Swift* UVOT with only very minor changes to account for the UVOT setup. Since the UVOT operations use each of the two grism also in a so-called “clocked” mode, where the filter wheel positions the grism optics slightly offset from the main optical axis, these were modeled separately.

Determination of the anchor positions

Each grism mode was calibrated in turn. More detailed results can be found in the HEASARC CALDB document, or in separate pages with links given below.

Considering the spectral resolution ($R=75$ uv grism; $R=100$ visible grism), and relatively faint calibration sources needed, the emission lines from Wolf-Rayet spectra were used. The UV grism, nominal mode calibration used mostly spectra taken from WR86, a known long period binary, with good lines going down to 1909Å. The UV clocked grism

used mostly WR52 which has lines going down even further in the UV. The visible grism calibration used for both modes mainly spectra from the fainter WR121, since the visible grism is too sensitive to use WR52 or WR86.

Initially, the ground calibration of the UV grism was also used as further verification of the in-orbit results. There were no unexpected discrepancies found.

The following steps were found to be needed to determine anchor positions:

- the images of the calibration line spectra were used to measure the position of the lines.
- the anchor position on each image was determined from the position of spectral lines.
- the position of the source in aspect corrected lenticular filter images was determined with respect to the boresight location at the time of observation.
- the boresight location of the anchor on the grism image was determined as well as the mapping from the lenticular filter to the grism image.
- the anchor positions predicted by the zemax model were shifted to align the model to the observations of the boresight.
- first the model was scaled with a simple scale factor to align all anchors to within 20Å.
- As a further correction a bispline was determined. The order of the bispline varies with the grism mode.

In a given observation, the actual anchor positions may still show an offset from the calibrated position. We find that the accuracy is about 15Å unless there is drift during the exposure of grism and associated lenticular filter. In a few cases the offsets found are as large as 40Å, so be warned.

If no lenticular filter observation is available and the UVOTGRASPCORR program has been used for finding the world coordinates of the grism image, the error in the anchor position may be much larger. This larger error seems to be due to some correctible problem. Currently (December 2014) the source of the problem has not been found.

Some sky locations are worse than others for attitude lock, and it is difficult to predict where and when the results will be off. *These anchor errors/offsets imply that the whole wavelength scale is shifted.*

Determination of the dispersion

How the dispersion changes by position of the spectrum on the detector has been determined by off-setting the calibration spectra so they covered all areas of the detector. The observed dispersion has been used to determine how to scale the “Zemax” model. To determine the dispersion scale factors, the line positions were determined in pixel distances from the anchor, and compared to the model. The scale factor was allowed to slowly vary with detector position.

In the 2009 calibration the model was scaled with a single factor, although the factor was allowed to vary slowly over the detector. That works well for most of the spectrum, but not for the shortest nor for the longest wavelengths. For the UV grism the model is lacking the details below 2100Å of the filter characteristics, while above 4000Å the PSF is assuming a very asymmetric and extended shape, calling into question the definition of the PSF peak. The 2013 version of the uv clocked grism wavecal is using a linear fit in wavelength rather than a fixed scale factor which gives a better dispersion result overall. For the other grism modes a single scale factor was sufficient.

The accuracy of the dispersion, neglecting wholesale offsets due to anchor problems, is about 15Å in the uv grism. It is worst at the low and high wavelengths (< 1900Å, > 4000Å).

Details for each grism mode

Pre-launch ground calibration of the UV-Grism

Relating the wavelength scale and sensitivity of the spectral orders

Report by Paul Kuin

Date:2008/07/21

Version: 1.1, converted to Sphinx 2013-06-11

Introduction In November 2002, the Swift UVOT underwent a ground calibration at the NASA GSFC. During this calibration, images and spectra were taken. The UV grism data discussed in this document are those taken for the redundant system which is the current operational system.

Observations and processing The instrument was mounted on the optical bench of the test system with the artificial source at, or very near, the boresight (unless the telescope was rotated on the gimbal for special observations). The light source was a platinum/Neon lamp. A graph paper with the spectrum of the lamp with a resolution of $100\text{\AA}/2\text{cm}$ was provided. Tests were taken with either a blank filter, or one of three narrow-band filters in the path of the beam that fed the telescope. The narrow band filters are known as NB193, NB214, and NB260. The document with the specifications of the band passes of the filters is not available to us. The effective bandpass of each band was reconstructed, using a combination of the observer notes provided, the zemax solution for the boresight, and tables of the Pt/Cr/Ne spectrum as provided by NIST on the WWW. The observations are available as event data in FITS files. The files were read in using an IDL routine to construct images, which are 2048x2048 pixels. The image X and Y were transposed in order to obtain the same geometry as the in-orbit data. The raw images were converted into detector coordinate images by appending headers to the raw images, and processing with the UVOT/ftools software (swiftxform).

In table 1, the observations of the current UV grism (redundant prism 1) that were used have been listed. Open items were not recorded.

Table 1. Observations used

obs #	Path	file-name	target	metallic neutral density NDm#	Narrow band filter λ	lamp mA	notes
0128	20021116	g_1637448536_0	pin-hole		NB214		on-axis
0129	20021116	g_16374410745_0	pin-hole		NB193		short time exposed with clear filter in addition to long exposure in NB193
0130	20021116	g_1637443724_0	pin-hole	clear	NB260	14	observer mistakenly notes a separation of the peaks $\lambda 2550$ and $\lambda 2559$ (These appear to be the $\lambda 2559+2550$ blend and $\lambda 2531$ instead)
0228	20021118	g_1637612492_0	pin-hole	clear	NB193	10	wheel position 200
0232	20021118	g_1637616576_0	pin-hole	clear	NB214	12	wheel position 200
0236	20021118	g_16376110591_0	pin-hole	clear	NB260		wheel position 200

Table 2: line identifications

Narrow band filter	observed wavelength		line identifications (blends)
NB193	1883	(weak peak)	1883.0
NB193	1916	(main peak)	1911.7,1916.1
NB193	1930	(strong secondary peak)	1929.2,1930.0,1937.4
NB193	1971	(weak peak)	1969.7,1971.5,1978.8
NB193	2036	(weak peak)	2030.5,2032.4,2035.8,2036.5,2041.6
NB214	2175	(weak peak, blends with 2203)	2174.7
NB214	2203	(secondary peak)	2202.3, 2209.5
NB214	2246	(main peak)	2245
NB260	2631	(secondary peak)	2631
NB260	2660	(main peak)	2559.45,2650.8

In the following plot, the total counts observed in three observations (128,129,130) with three different filters have been combined and plotted as function of wavelength. Note that the spectral range probed is from about 1850 to 2800 Angstroms.

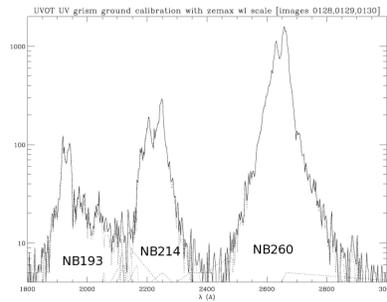


Figure 1.9: Figure 1. Combined in one figure are three observations with the measurements from each filter along the dispersion direction in the first order. The observations were taken in sequence with the source on-axis. The dotted lines show the individual filter contributions.

A contour plot shows the location of the flux on the image. Once again observations in three filters were combined in one image.

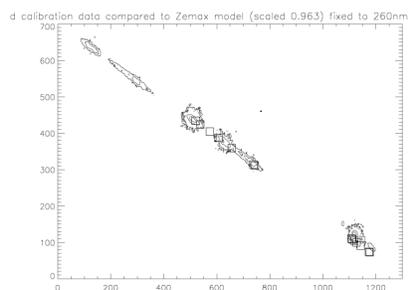


Figure 1.10: Figure 2 shows the relative positions (from right to left) of the zero, first, and part of the second order, in the calibration observations. The zemax model calculations, with pixels scaled by 0.963, have been plotted as squares at wavelengths ranging from 1900-2600 Å. The zemax model was shifted to fit the first order. As can be seen, the zero order shows a slight offset from the model position. This zemax model was calculated with a 3.8 degree correction.

Table 3 lists the (detector) image coordinates of the main peaks derived from examining the data in “DS9”. These have FITS coordinates defined by:

```
CRPIX1=996.5
CRPIX2=1022.5
CRVAL1=0.
CRVAL2=0.
CDELTA1=0.009075
CDELTA2=0.009075
CTYPE1= 'DETX' /X coordinate type
CTYPE2= 'DETY' /Y coordinate type
CUNIT1 = 'mm' / X coordinate units
CUNIT2 = 'mm' / Y coordinate units
```

Table 3: Coordinates of peaks (DET image)

Filter	obs	Zeroth order	First order	Second order	Third order
NB193	129	(1650.1,562.8)	(1207.3,807.0)	(781.3,1039.5)	(351.0,1272.4)
		(1641.4,568.3)	(1200.0,811.0)	(768.5,1046.0)	(339.0,1277.5)
NB214	128	(1605.9,592.9)	(1108.8,864.0)	(619.5,1129.0)	(132.3,1394.0)
		(1603.0,594.0)	(1095.3,871.8)	(596.8,1142.0)	(102.5,1413.3)
NB260	130	(1579.1,610.5)	(987.4,933.0)	(395,1255)	off detector
NB193	228	(1638.5,576.8)	(1204.5,814.5)	(796.5,1035.0)	(360.5,1275.0)
			(1198,817.5)	(780.0,1045.5)	(345.0,1280.0)
NB214	232	(1604.5,600.0)	(1105.5,871.5)	(618.0,1136.5)	(108,1411) (90,1421)
		(1598.9,602.1)	(1093.0,878.5)	(593.5,1150.0)	
NB260	236	(1578.5,615.9)	(984.1,940.5)	(395.4,1260.3)	off detector
		(1576,4,618.5)	(977.3,944.5)	(388.8,1266.1)	

The table 4 lists the distances (along the dispersion) of the main lines of each narrow-band filter observation between the first and zeroth orders.

Table 4: measurements of order distances along dispersion direction (DET image)

filter	obs	line	zero-first	first-second	first-third
NB193	0129	1916	505.7	485.3	975
NB193	0129	1930	503.7	419.4	979
NB214	0128	2203	566.2	556.4	1112
NB214	0128	2246	578.7	567.0	1131
NB193	0228	1916	502.0	463.8	961.5
NB193	0228	1930		476.1	970.3
NB214	0232	2203	568.1	554.9	1134.5
NB214	0232	2246	576.5	568.5	1140.3
NB260	0236	2631	677.3	670.0	
NB260	0236	2660	682.1	674.3	
NB260	0130	2631		674.3	
NB260	0130	2660		670.3	

A second order polynomial fit ($D(\text{pix}) = \sum_i C_i \lambda^i$, λ in Å) to the distance from the first to the zeroth, second, and third orders as a function of wavelength was made. The coefficients are:

Table: Polynomial coefficients

.	D(first-zero)	D(first-second)	D(first-third)
c0	241.6 +/- 52.2	-155.5	-2733
c1	0.0595 +/- 0.0463	0.375	3.136
c2	(4.00 +/- 1.01) 10 ⁻⁵	-2.374 10 ⁻⁵	-6.3 10 ⁻⁴

The result can be seen in Figure 3.

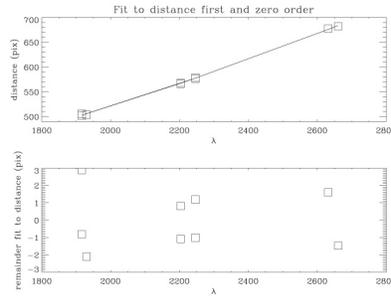


Figure 1.11: Figure 3: the fit to the measured distances from the ground calibration.

The zemax model dispersion relative to 260nm in first order was plotted against the measured dispersion from the ground calibration. The corresponding zemax numbers, are presented in Table 5 for different scale factors. The scale factor of 0.963 ± 0.003 was derived from fitting the wavelength scale to the first order spectra of WR52 and WR86. The distances between zero and first order for the 0.963 model are compared to those from the ground calibration in Figure 4. At wavelengths below 2000\AA the zemax model is not reliable. It can be seen that the measurements for wavelengths larger than 2000\AA are on the average four pixels less distant in the ground calibration. For comparison the zemax results, scaled by a factor of 0.957 have been included, which show a much better fit to the ground calibrated distances above 2000\AA . A scale factor of 0.957 is two sigma from the result from fitting the first order wavelength scale, and is thus not inconsistent therewith. It must be kept in mind, that to derive the distance, the measured coordinates of the peaks were used. Since there is a slight offset of the actual zero order from the Zemax model, while the model distances include the offset, there is no reason to adjust the pixel scale factor from the 0.963 value.

Table 5. Zemax model distances along the dispersion for varying scale factors compared to the measured ground calibration

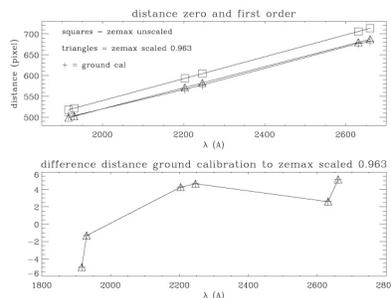


Figure 1.12: Figure 4: comparison ground calibration zero-first order distances to zemax with scale factor = 0.963.

The wavelength scale of observations might now be extended to the zero order, allowing to determine the wavelength of the uvotgraspcorr anchor point, and a photometric calibration for the zero order.

Distance of the first to second order as derived above from the ground calibration, was also compared to the zemax model results. Like for the zero-first order distances, the wavelengths below 2000\AA do not fit very well with the zemax predictions, and the model with the scale factor of 0.963 fits quite well. It may be that the 1916\AA is a misidentification, or that it is not very reliable since it is very noisy.

Table 6. Zemax model distances along the dispersion for varying scale factors compared to the measured ground calibration

The total counts in each filter give a good measurement of the relative sensitivity in the different orders, provided the coincidence loss is negligible. Inspecting the brightest features, no evidence for the kind of pattern we see in sky observations is seen. In fact, the source seems more extended. The source aperture may have been larger than a source on the sky depending on the distance of the pinhole to the instrument. It is therefore possible that some photons

have been lost through the auto-coincidence code, which rejects detections that are within one frame in neighboring pixels. The 0130 observation appears to be approximately 2000 seconds long (no record was kept), and the highest peak rate is about 27000 in the 260NB filter, spread out over 600-800 pixels. The maximum rate is therefore about $24000/2000/600 = 0.02$ c/s/pixel. This is small enough that coincidence losses are negligible.

Table 7a. Measured relative sensitivities between orders: net counts by filter and observation

.	NB193/0129	NB193/0228	NB214/0128	NB214/0232	NB260/0130	NB260/0236
zeroth order	240	319	367	183.4	10545	1900
first order	2572	4103	5010	2630	24232	4274
second order	2063.5	3250	2025	1158	5434	935
order ratio						
Zeroth/first (percent)	9.3+/-1.5	7.8+/-0.5	7.3+/-0.5	6.9+/-0.5	43.5+/-0.4	44 +/- 1.5
Second/first	80+/-2	79.2+/-1	40.4+/-1	44+/-2	22.4+/-0.5	22 +/- 1.5

Table 7b. Measured relative sensitivities between orders: net counts by filter and derived zeroth and second order effective areas (in cm²)

.	NB193	NB214	NB260
ratio			
Zeroth/first	8.6 +/- 1	7.1 +/- 0.4	44 +/- 0.5
Second/first	80 +/- 1	42 +/- 1	22 +/- 0.5
zero order effective area	0.51	0.76	5.6
first order effective area			
(cm ²) (adopted from CALDB)	6.0	10.7	12.7
second order effective area	4.8	4.5	2.8

The extraction of the counts was done with a width of 25 pixels, which included most of the signal. The reason for such a wide extraction width is that the source is slightly extended due to the finite aperture. The effective areas were derived from the ratios and the effective area in the CALDB with latest update dated 2007-07-11, and are mainly included to give an idea of their size.

Point Spread Function The point spread function in the zero and second orders was calculated with Zemax. A summary is given in table 8. The shape of the PSF is a peak with a horse-shoe like extension. It has been fitted with a 2-D gaussian, but I only list the radial widths. Listed are the full width at half maximum, as well as the maximum radius to which the PSF extends. The PSF needs to be studied further, in particular the extend along and across the dispersion, and the energy contained within certain limits which have to be defined intelligently. As such, this table should only be taken as an indication of the character of the PSF variations.

Table 8. Summary of Zemax calculations of the PSF wavelength (Å)

wave-length	zeroth order FWHM (pix)	zeroth order maximum radius (pix)	second order FWHM (pix)	second order maximum radius (pix)
1900	9.0	19	5.9	12
2000	8.4	18	6.5	14
2100	7.7	17	7.7	16
2200	7.4	16.3	8.7	19
2300	7.1	15.7	10	22
2530	6.5	14.6	14	29
2900	5.9	13.6	22	46
4000	5.3	12.1	63	127
5500	5.0	11.5	—	—

Although below 1900Å the zero order dispersion is approaching one Å/pixel, the PSF of the spectrum is increasingly broadened. This is consistent with observations of WR52, which showed very broad bumps. A detailed comparison,

using the wavelength scale from the data above, must still be made.

In the second order, the PSF becomes very large above 2500Å. A similar occurrence may happen in the first order above ~ 5000 Å.

Summary

- The wavelength scale in the zero and second order has been determined relative to the first order, using the on-axis observations taken during the ground calibration in the 1900-2700Å range.
- The relative sensitivity of the zeroth and second order with respect to the first order has been determined from the ground calibration measurements.
- The point spread function in the zeroth and second order was determined from the Zemax optical model, and shows that at short wavelengths the lines in the zero order are the broadest, and the lines in second order are the broadest for the longest wavelengths.

References

- “Atlas of the spectrum of a Platinum/Neon Hollow Cathode Lamp in the region 1130-4330Å,” J.E.Sansonetti, C.J. Sansonetti, J.Reader, N. Aquista, A.M. Sansonetti, and R.Dragoset, J.Res.Natl.Inst.Stand.Tech. 97,1-212 (1992).
- The ground test document: “Grism Resolution; Swift optical test procedure”. SWIFT-UVOT-145-R01, 10 Nov. 2002.
 - accompanying Spreadsheet 02_11_15_psf_data_sheet_final.xls.
 - accompanying CD-ROMs with data from the ground calibration.
 - accompanying chart with spectral output of Pt/Cr/Ne Lamp, dd. 3-11-2002, Lamp S/N NY10258A.

Swift UVOT UV-grism wavelength accuracy in the nominal mode

A Statistical overview of the wavelength calibration accuracy The errors in the wavelength calibration are summarised in the figures on this page, while details for the individual calibration observations are on subsequent pages. The error analysis is based on grism plus lenticular filter observations. If further updates were made they are mentioned at the bottom of this page.

The inset in the top left corner shows that at most times the anchor point that was predicted using the lenticular filters taken just before and after the observation, for the first version of this calibration has a mean offset of -8.0 Å averaged over the calibration observations. One outlier shows that occasionally an even larger offset occurs. Ignoring this offset, the anchor is found to cause an error which can be as large as 33Å.

The contours give a better picture of where the calibration for the anchor position results in an offset in the wavelengths. In the corners and near the edges, only a few observations were available for the calibration, so the values should be taken as an indication of the size of the error, rather than an accurate value that can be used to improve the wavelength scale knowledge.

Looking at the second inset “accuracy”, and disregarding the systematic wavelength scale offset from the top left inset, wavelengths from the dispersion relation are found to typically fall within 14 Å (about 4 pixels) of the correct value. The “accuracy” is the RMS dispersion of the dispersion errors over the whole range as shown in the accuracy plots in more detail, and not the error themselves.

The variation of the accuracy of the dispersion relation over the detector can be seen from the contour plot below. The contours are the RMS values of the wavelength errors of known lines in the calibration spectra. There is only a small variation over the face of the detector present, with the largest discrepancies in the top left and bottom right corners. This can possibly be improved by applying a bilinear correction as a function of the detector position to the Zemax

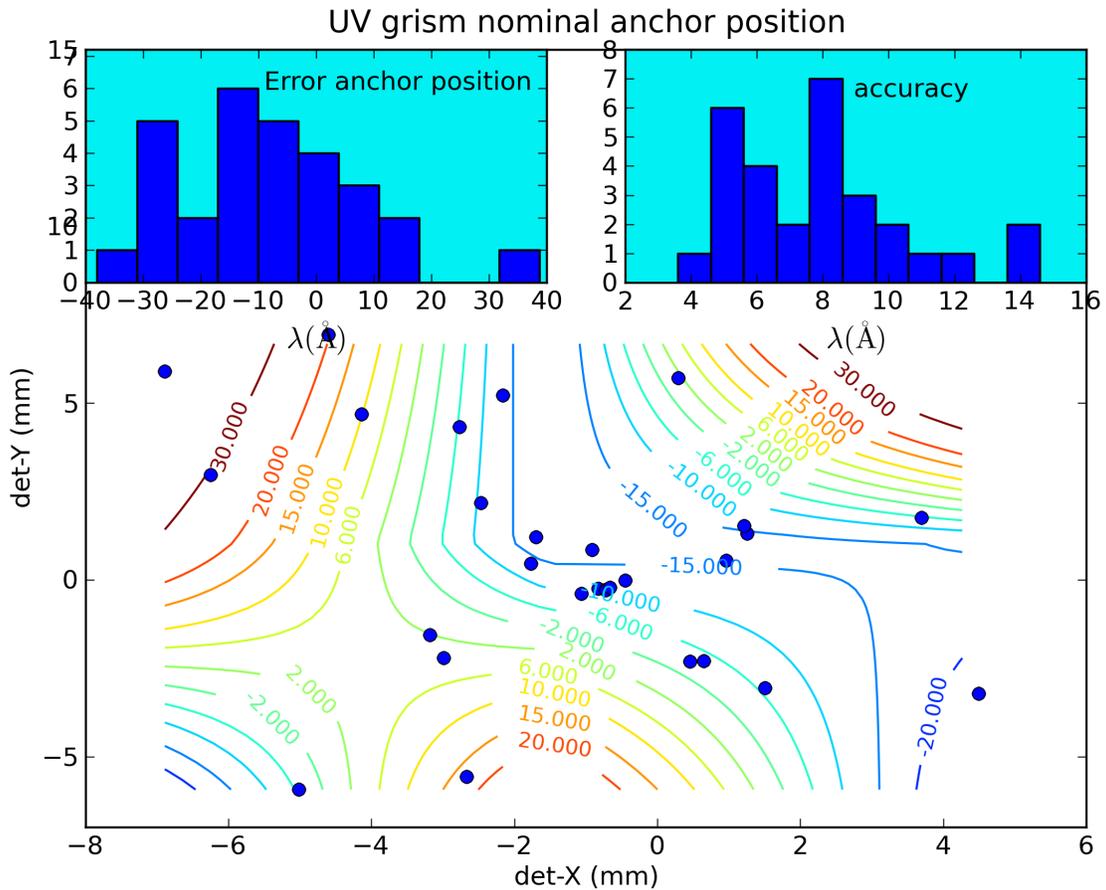


Figure 1.13: Figure 1: The wavelength scale errors have been split in two parts. The first is the offset of the whole scale due to errors in the anchor position and is shown as a histogram in the inset top left. The second is due to errors in the dispersion and is shown as a histogram in the top right. The data are based on the calibration spectra taken at locations all over the detector. The locations are shown in the main graphs by their anchor positions as blue dots. The error in anchor position is smallest near the default location for placing targets and largest near the edges and are shown as contours labeled with the offsets in Å.

model dispersion constants. In the UV nominal grism, a constant was used. In the other grism modes which were done later, a bilinear correction was used. The overall accuracy is generally sufficient to avoid making misidentifications of the larger spectral features which have a FWHM which is comparable in magnitude, although wavelength scale errors of up to 40 Å were seen for the calibration.

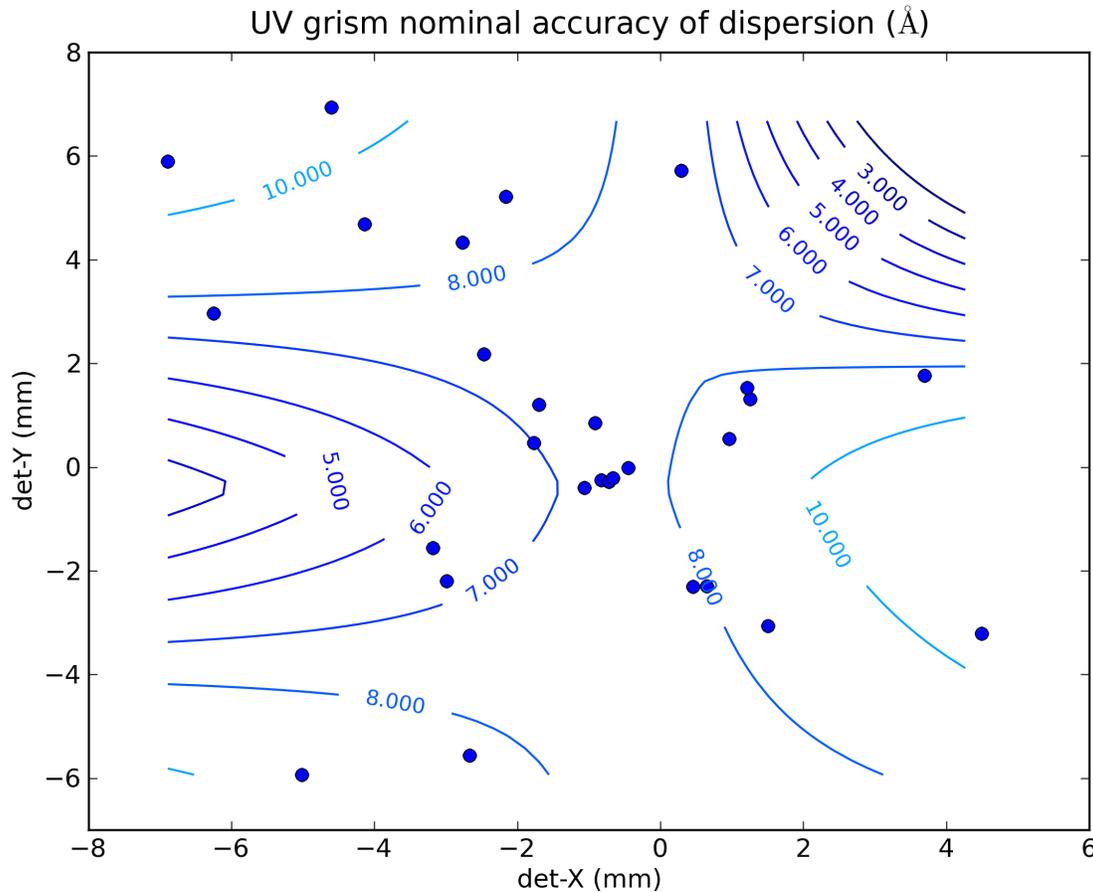


Figure 1.14: Figure 2: contour map showing the variation over the detector of the RMS error due to dispersion. Positions were based on the anchor position of each calibration spectrum.

Detailed plots of the wavelength accuracy and of the spectra used In order to allow better understanding of the capabilities, the data obtained by verification of the calibration spectra with the calibration are provided, as well as plots of the accuracy of fit for the individual observations. See the detailed spectra and accuracy plots.

Updates to the calibration A small correction was made to the anchor positions prior to the release of the wavelength calibration file to compensate for the mean shift seen in the histogram of the anchor position offsets.

Technical Documents

- Swift UVOT Calibration report. Zemax optical models for the UV-Grism study: Zemax pixel scale factor and positioning. MSSL 2008.
- Swift UVOT calibration report: Zemax optical models for the uv grism study. Second order wavelength scale for sources near the axis.

Swift UVOT UV-grism wavelength accuracy in the clocked mode

A Statistical overview of the wavelength calibration accuracy The errors in the wavelength calibration are summarised on this page, while the verification details using the individual calibration observations are on subsequent pages. The error analysis is based on grism plus lenticular filter observations.

For version 1 a detailed analysis was done taking the new calibration file and reanalysing the data, measuring the line positions, and comparing to the correct positions. For the version 2 (June 2013) update, the anchor positions and dispersion relations from the calibration file were compared to those measured directly from the observations.

If further updates were made they will be mentioned at the bottom of this page.

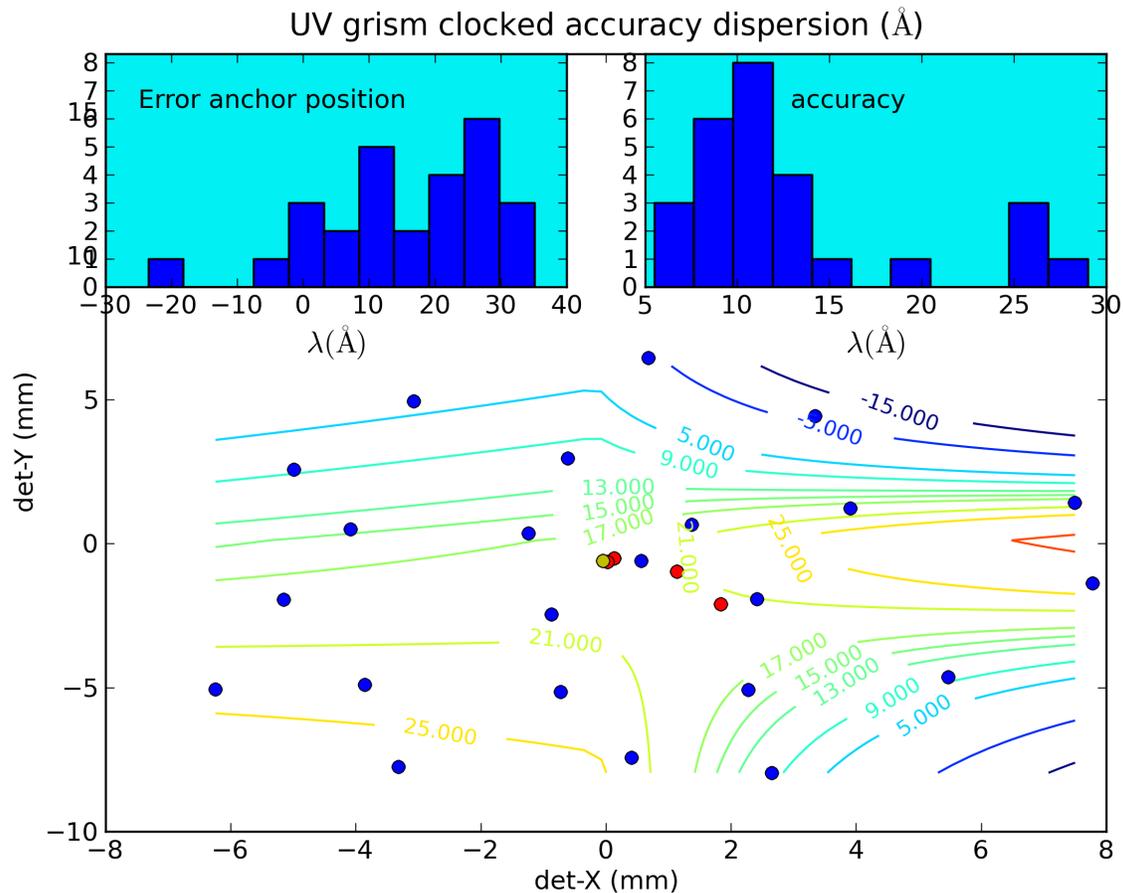


Figure 1.15: Figure 1: The wavelength scale errors have been split in two parts. The first is the offset of the whole scale due to errors in the anchor position and is shown as a histogram in the inset top left. The second is due to errors in the dispersion and is shown as a histogram in the top right. The data are based on the calibration spectra taken at locations all over the detector. The locations are shown in the main graphs by their anchor positions as blue dots. The error in anchor position is smallest near the default location for placing targets and largest near the edges and are shown as contours labeled with the offsets in Å.

Version 1 wavelength calibration (2009) The inset in the top left corner shows that at most times the anchor point that was predicted using the lenticular filters taken just before and after the observation, for the first version of this calibration has a mean offset of 15.9 Å averaged over the calibration observations. One outlier shows that occasionally an even larger offset occurs. Ignoring this offset, the anchor is found to cause an error which can be as large as 33Å.

The contours give a better picture of where the calibration for the anchor position results in an offset in the wavelengths. In the corners and near the edges, only a few observations were available for the calibration, so the values should be taken as an indication of the size of the error, rather than an accurate value that can be used to improve the wavelength scale knowledge.

Looking at the second inset “accuracy”, and disregarding the systematic wavelength scale offset from the top left inset, wavelengths from the dispersion relation are found to typically fall within 14 Å (about 4 pixels) of the correct value. The “accuracy” is the RMS dispersion of the dispersion errors over the whole range as shown in the accuracy plots in more detail, and not the error themselves.

This figure shows the errors in the wavelength calibration. The blue dots in the main figure are the anchor positions (at 260nm in first order) of the calibration spectra on the detector. The contours are a fit to the accuracy of the dispersion expressed as the RMS error between measured and predicted wavelength for known lines in the spectrum.

The inset in the top left corner shows that at most times the anchor point that was predicted using the source position in the lenticular filters taken just before and after the observation has a systematic offset of 15.9 Å. (This offset was corrected for in a final revision of the wavelength calibration file). Disregarding the systematic offset, the anchor position typically falls within 23 Å (about 7 pixels) of the mean position. This is sufficient to avoid making misidentifications of the larger spectral features which have a FWHM which is comparable in magnitude.

The inset in the top right corner shows the accuracy of the adopted dispersion from the scaled zemax model after removal of the anchor point offsets. The histogram shows the root mean squared measure. The contours in the main figure show the overall variation of dispersion accuracy over the detector. The main result is, that the RMS error in the wavelengths in a spectrum is typically less than 16 Å, which is about 5 pixels. The variation of the accuracy of the dispersion relation over the detector can be seen from the contour plot below. The contours are the RMS values of the wavelength errors of known lines in the calibration spectra. There is only a small variation over the face of the detector present, with the largest discrepancies in the top left and bottom right corners.

Detailed plots of the wavelength accuracy and of the spectra used In order to allow better understanding of the capabilities, the data obtained by verification of the calibration spectra with the calibration are provided, as well as plots of the accuracy of fit for the individual observations. See the detailed spectra and accuracy plots.

Version 2 (June 2013) The correct Zemax model parameters were used. The observations were reprocessed since the software had much improved since the last version was made, and stored in a Python dictionary. This allowed a much more straightforward check of the resulting calibration file and intermediate products. Since there was a remaining mismatch to the scaled Zemax models at long wavelengths, and to some extent at the shortest, a second term was added to the scaling of the zemax model. That term was held constant over the whole detector, being small and difficult to determine to a larger accuracy with the currently available data. The zemax model was scaled in two steps, first a large scale factor to all model positions that brought the model very close to the observed dispersion at the default (centre) position, and with a small positional shift to line up the observed drop in sensitivity with the model (as discussed in the flux calibration). The differences between observed anchor position and intermediary model were next fitted to a bispline of order 1(in x)+2(in y), and used to derive corrected anchor positions for the zemax model. The dispersion scaling relative to the anchor was next addressed. The initial scaling of all positions had already given a better dispersion in the intermediate model. A further (dispersion) scale factor was derived which varies slowly over the detector and is modeled by a bispline function. Over most of the detector that correction is within 2%. For each model anchor point one scale factor plus a small higher order term were then used to scale the dispersion. The number of adjustable parameters is much smaller than the observations used. Therefore the new calibration can be used with the calibration observations to get an estimate of the accuracy of anchor position and dispersion.

Although the range of offsets is similar as in version 1, more observations lie very close to the correct anchor point.

Within the range up to 4000Å, we find that the mean error of the offset (due to anchor misalignment) is 4.5 Å, while the standard deviation of wavelength errors is 16.7 Å, which are the numbers to compare to the version 1 calibration. The errors are not seen to be larger than 22 Å.

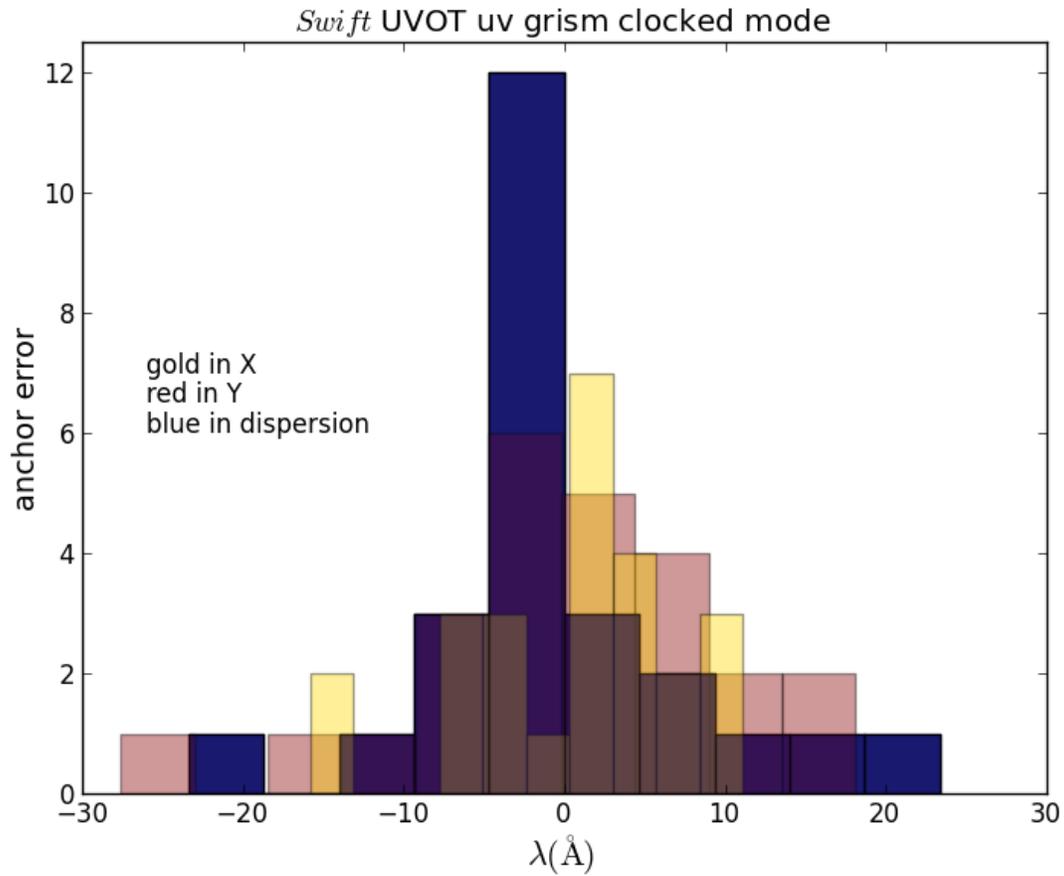


Figure 1.16: Figure 2: The X, and Y are the measured offsets between the correct anchor position and the position extracted from the calibration file. The offset in pixels was multiplied by 3.2 to get a wavelength offset estimate for this plot. Using the typical angle of the spectrum on the detector image, the wavelength offset in the dispersion direction due to the error in anchor position was determined (dark blue).

Updates to the calibration Due to an error in the Zemax model generation used for the first version of the wavelength calibration, the dispersion was not very good at long wavelengths. This has been corrected in Version 2 (June 2013) of the wavecal file. The error analysis of Version 1 above is still valid.

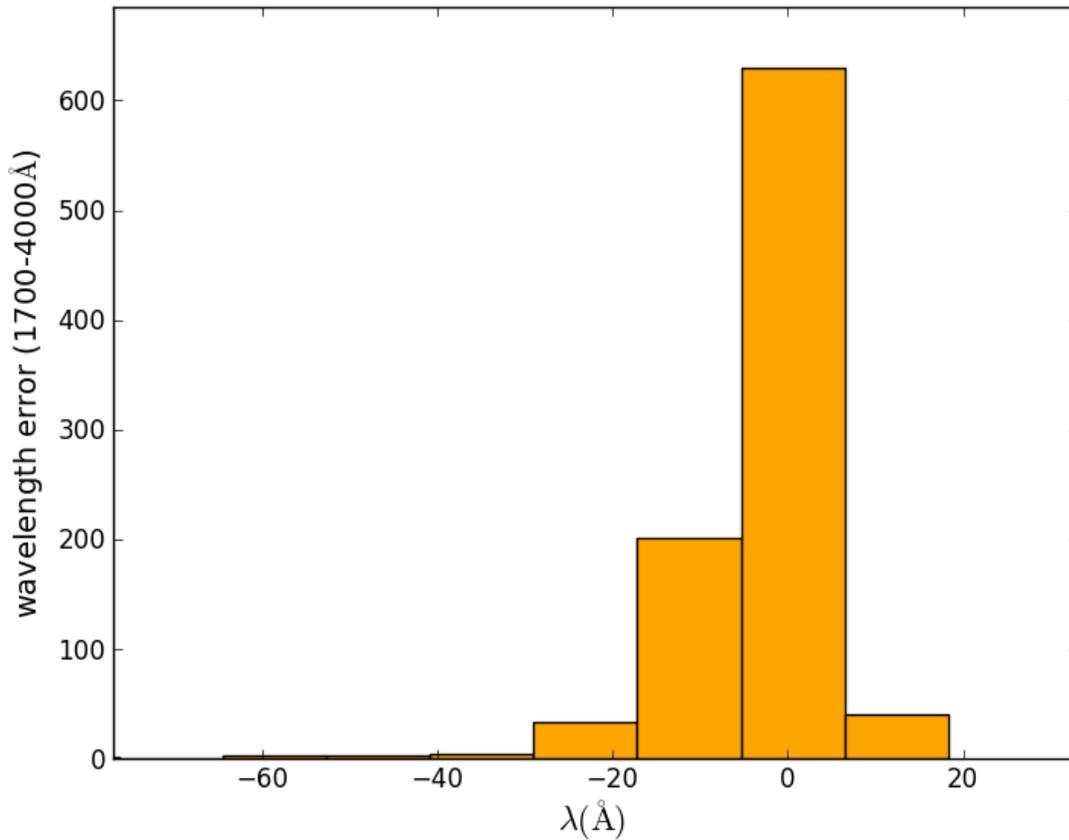


Figure 1.17: The error in the wavelength (due to dispersion) sampled over all calibration observations covering the detector. One outlier has been clipped.

For each calibration observation, the difference in wavelength was derived using the correct dispersion for the observation, and the one derived using the calibration file. One outlier was removed (with errors up to 200\AA). The errors are dominated by those at large wavelengths,

The total error is a linear combination of that from the anchor and dispersion, so lines may in some cases be displaced by $\sim 40\text{\AA}$.

Technical Documents

- Swift UVOT Calibration report. Zemax optical models for the UV-Grism study: Zemax pixel scale factor and positioning. MSSL 2008.
- Swift UVOT calibration report: Zemax optical models for the uv grism study. Second order wavelength scale for sources near the axis.

Swift UVOT visual-grism wavelength accuracy in the nominal mode

A Statistical overview of the wavelength calibration accuracy The errors in the wavelength calibration are summarised on this page, while the verification details using the individual calibration observations are on subsequent pages. The error analysis is based on grism plus lenticular filter observations.

For version 1 a detailed analysis was done taking the new calibration file and reanalysing the data, measuring the line positions, and comparing to the correct positions. For the version 2 (June 2013) update, the anchor positions and dispersion relations from the calibration file were compared to those measured directly from the observations. In 2014 independent sources were used to verify the error in wavelength caused by errors in the anchor position.

The analysis of the 2009 calibration, and the 2013 update, are still valid, as shown by the 2014 verification.

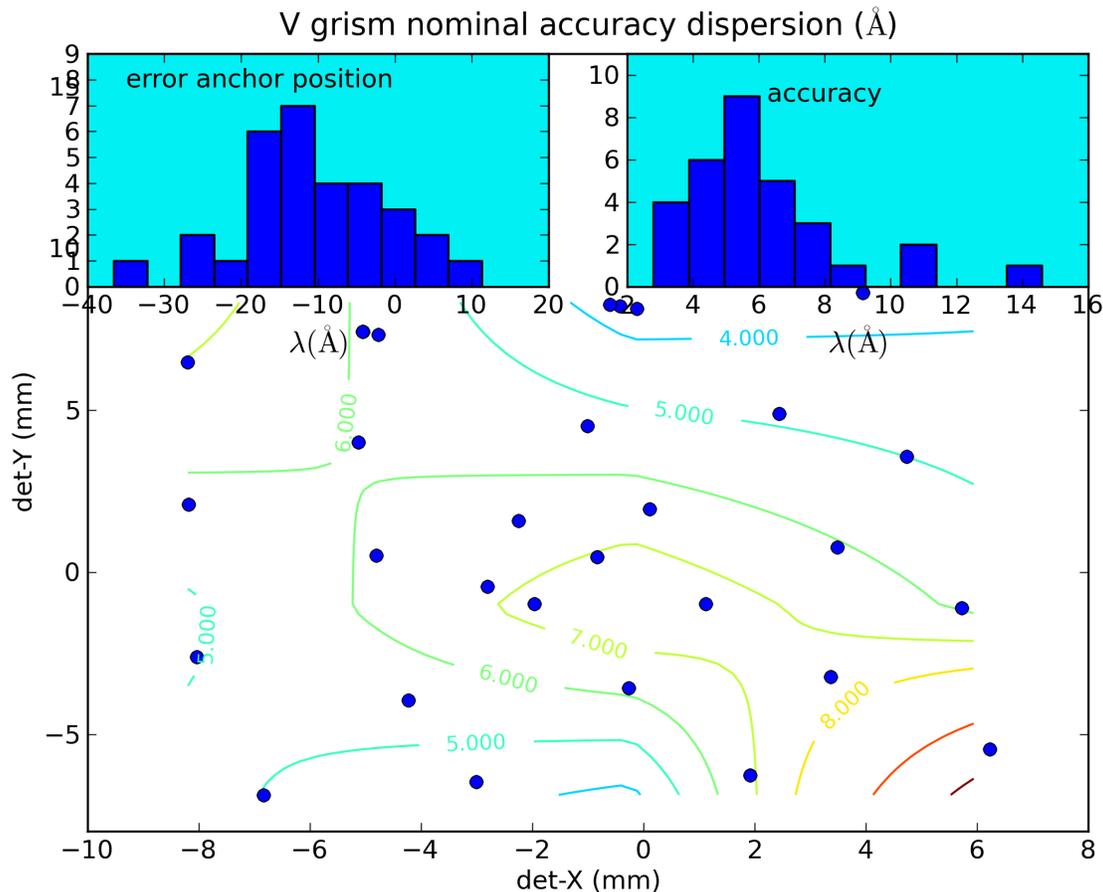


Figure 1.18: *Figure 1: The wavelength scale errors have been split in two parts. The first is the offset of the whole scale due to errors in the anchor position and is shown as a histogram in the inset top left. The second is due to errors in the dispersion and is shown as a histogram in the top right. The data are based on the calibration spectra taken at locations all over the detector. The locations are shown in the main graphs by their anchor positions as blue dots. The error in anchor position is smallest near the default location for placing targets and largest near the edges and are shown as contours labeled with the offsets in Å. The contours are a fit to the accuracy of the dispersion expressed as the RMS error between measured and predicted wavelength for known lines in the spectrum.*

The 2009 wavelength calibration The inset in the top left corner shows that at most times the anchor point that was predicted using the source position in the lenticular filters taken just before and after the observation has a systematic

offset of 15.9 Å. (This offset was corrected for in a final revision of the wavelength calibration file). Disregarding the systematic offset, the anchor position typically falls within 23 Å (about 7 pixels) of the mean position. This is sufficient to avoid making misidentifications of the larger spectral features which have a FWHM which is comparable in magnitude. (But see the 2014 verification using independent sources below for the definite number).

Looking at the second inset “accuracy”, and disregarding the systematic wavelength scale offset from the top left inset, wavelengths from the dispersion relation are found to typically fall within 14 Å (about 4 pixels) of the correct value. The “accuracy” is the RMS error of the wavelength as shown in the accuracy plots in more detail, and not the error themselves. The contours in the main figure show the overall variation of dispersion accuracy over the detector.

To summarize, the main result is, that the RMS error in the wavelengths within a spectrum is typically less than 16 Å, which is about 5 pixels, much smaller than the error from inaccurate anchor position of about 33 Å.

Detailed plots of the wavelength accuracy and of the spectra used (2009) In order to allow better understanding of the capabilities, the data obtained by verification of the calibration spectra with the calibration are provided, as well as plots of the accuracy of fit for the individual observations. See the detailed spectra and accuracy plots.

Recalculation in 2013 In 2013 the line positions and anchor were measured again in each spectrum and stored. The calculated positions were then compared to the observed ones and the errors recalculated. This confirmed the earlier results. It also allows comparison by recalculating errors when program updates produce new calculated line positions, for example after an update of the wavecal file.

The 2014 verification using independent sources In the flux calibration a different set of sources were used, with white dwarf and solar type star spectra. Several of these lent themselves for checking the wavelength accuracy. Since the main source of error in the wavelengths is due to the inaccuracies in the anchor, the main focus was there.

It became apparent that the whole wavelength scale is not just shifted due to an inaccurate anchor position, but also that such a shift causes a distortion to the wavelengths due to the non-linear terms in the dispersion (polynomial) relation. However, when a wavelength shift was observed and corrected for, recalculating the wavelengths using the dispersion relation always gave very acceptable fits to the wavelengths of all known lines, consistent with the 2009 analysis.

The anchor accuracy as a wavelength shift The anchor accuracy was determined for the independent sources in two ways. (1) When a lenticular filter exposure was taken during the grism observation. (2) Only a grism exposure was done (i.e., pre-2008 observations), in which case the anchor position relies on a correct aspect solution with the “uvotgraspcorr ftool”.

The anchor accuracies when using an observation with a lenticular filter exposure included are the same as those found in the 2009 wavelength calibration. However, the spectra which did not have a lenticular filter to determine the anchor showed a complicated behavior with nearly twice the error.

It appears that for some target fields consistent anchor point solutions are found with about the same internal error as when the analysis has the benefit, and uses, a lenticular filter to find the anchor. Then there are some targets that show anchor point solutions that are part of two or three bunches, which appear to occur after an offset or change in roll angle has changed the star field. For one source there are many such bunches, and the impression is that this is a systematic problem in finding a good aspect solution for the grism image using the “uvotgraspcorr ftool”.

A further analysis of this issue will be undertaken, and hopefully we find the bug and can bring the anchor accuracy of all exposures to the same level.

Technical Documents

- Swift UVOT Calibration report. Zemax optical models for the UV-Grism study: Zemax pixel scale factor and positioning. MSSSL 2008.

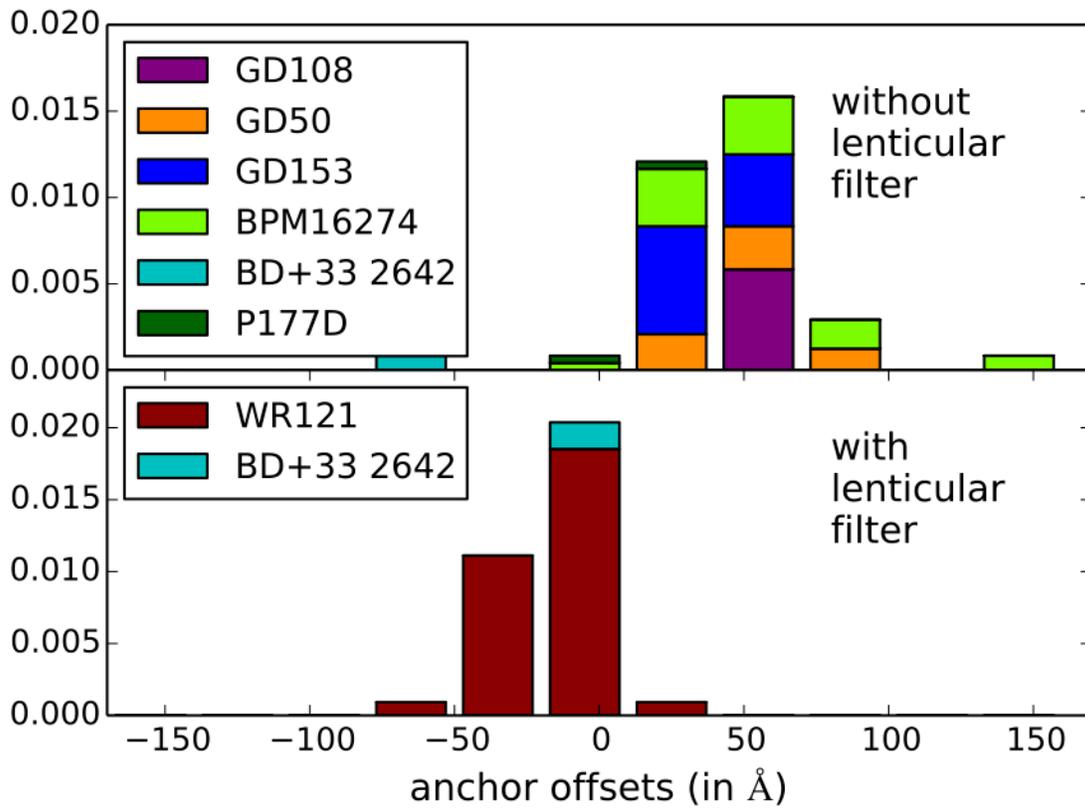


Figure 1.19: *Figure 2: 2014 wavecal verification*: This shows the 2014 verification of the wavelength shift due to anchor inaccuracy. It can be seen that in the absence of a lenticular filter the wavelengths are shifted much more, and that the offset depends on the source.

- Swift UVOT calibration report: Zemax optical models for the UV grism study. Second order wavelength scale for sources near the axis.

Swift UVOT visual-grism wavelength accuracy in the clocked mode

A Statistical overview of the wavelength calibration accuracy The errors in the wavelength calibration are summarised on this page, while the verification details using the individual calibration observations are on subsequent pages. The error analysis is based on grism plus lenticular filter observations.

For version 1 a detailed analysis was done taking the new calibration file and reanalysing the data, measuring the line positions, and comparing to the correct positions. For the version 2 (June 2013) update, the anchor positions and dispersion relations from the calibration file were compared to those measured directly from the observations. In 2014 independent sources were used to verify the error in wavelength caused by errors in the anchor position.

The analysis of the 2009 calibration, and the 2013 update, are still valid, as shown by the 2014 verification.

The 2009 wavelength calibration This figure shows the errors in the wavelength calibration. The small inset in the top left corner shows that at the anchor point that was predicted using the lenticular filters taken just before and after the observation, falls within 22 Å (about 3 pixels) of the correct position. This is sufficient to avoid making misidentifications of the larger spectral features which have a FWHM which is comparable in magnitude.

The inset on the right and the contour lines in the main figure show the RMS of difference between the measured and predicted line positions after removal of the anchor point offsets.

The main result is, that the RMS error in the wavelengths in a spectrum is typically less than 24 Å, which is about 3 pixels. (But see the 2014 verification using independent sources below for the definite number).

Detailed plots of the wavelength accuracy and of the spectra used (2009) In order to allow better understanding of the capabilities, the data obtained by verification of the calibration spectra with the calibration are provided, as well as plots of the accuracy of fit for the individual observations. See the detailed spectra and accuracy plots.

Recalculation in 2013 In 2013 the line positions and anchor were measured again in each spectrum and stored. The calculated positions were then compared to the observed ones and the errors recalculated. This confirmed the earlier results. It also allows for a future comparison using the measured line positions by recalculating errors when program updates produce new calculated line positions, for example after an update of the wavecal file.

The 2014 verification using independent sources In the flux calibration a different set of sources were used, with white dwarf and solar type star spectra. Several of these lent themselves for checking the wavelength accuracy. Since the main source of error in the wavelengths is due to the inaccuracies in the anchor, the main focus has been the anchor position accuracy.

It became apparent that the whole wavelength scale is not just shifted due to an inaccurate anchor position, but also that such a shift causes a distortion to the wavelengths due to the non-linear terms in the dispersion (polynomial) relation. However, when a wavelength shift has been observed and corrected for, recalculating the wavelengths using the dispersion relation always have given very acceptable fits to the wavelengths of all known lines, consistent with the 2009 analysis.

The anchor accuracy as a wavelength shift The anchor accuracy was determined for the independent sources in two ways. (1) When a lenticular filter exposure was taken during the grism observation. (2) Only a grism exposure was done (i.e., pre-2008 observations), in which case the anchor position relies on a correct aspect solution with the “uvotgraspcorr ftool”.

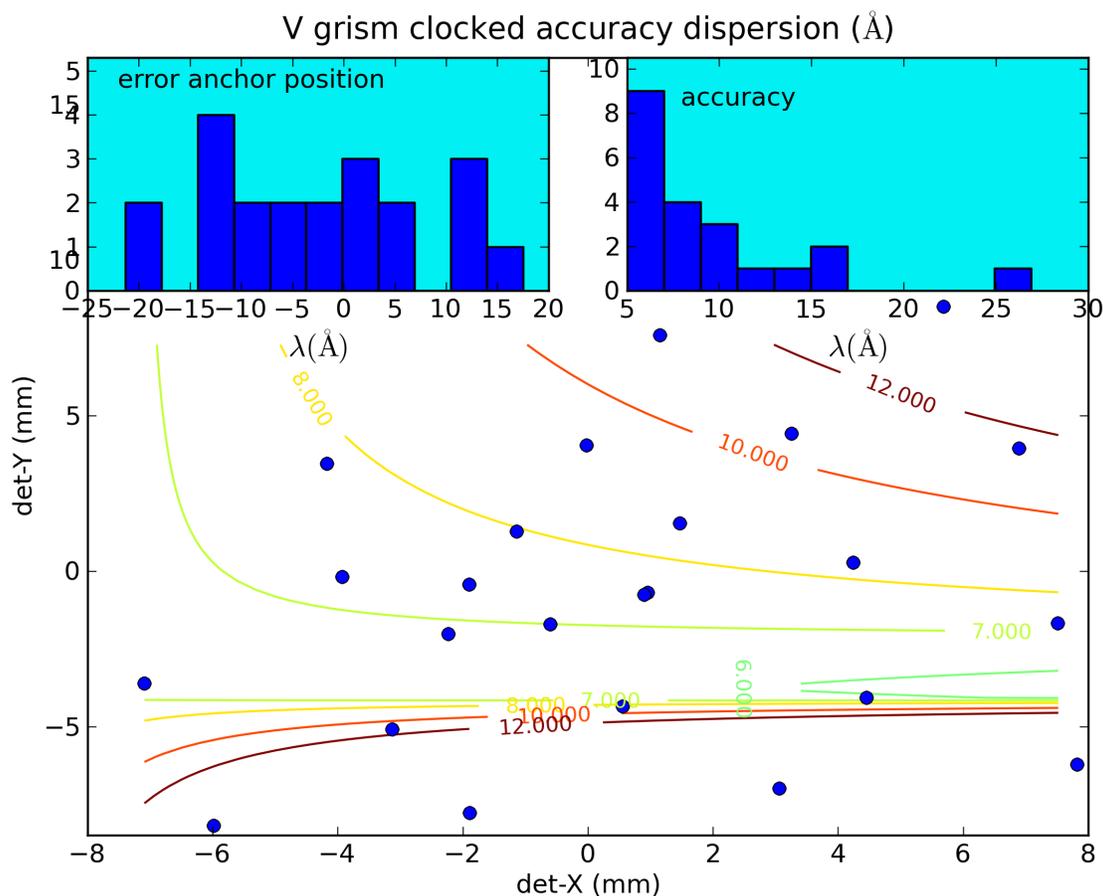


Figure 1.20: *Figure 1: The wavelength scale errors have been split in two parts. The first is the *offset* of the whole scale due to errors in the anchor position and is shown as a histogram in the inset top left. The second is due to *errors in the measured line positions* and is shown as a histogram in the top right. The data are based on the calibration spectra taken at locations all over the detector. The locations are shown in the main graphs by their anchor positions as blue dots. The error in anchor position is smallest near the default location for placing targets and largest near the edges and are shown as contours labeled with the offsets in \AA 's. The contours are a fit to the accuracy of the dispersion expressed as the (1-sigma) RMS error between measured and predicted wavelength for known lines in the spectrum.*

The anchor accuracies when using an observation with a lenticular filter exposure included are the same as those found in the 2009 wavelength calibration. However, the spectra which did not have a lenticular filter to determine the anchor showed a complicated behavior with nearly twice the error.

It appears that for some target fields consistent anchor point solutions are found with about the same internal error as when the analysis has the benefit, and uses, a lenticular filter to find the anchor. Then there are some targets that show anchor point solutions that are part of two or three bunches, which appear to occur after an offset or change in roll angle has changed the star field. For one source there are many such bunches, and the impression is that this is a systematic problem in finding a good aspect solution for the grism image using the “uvotgraspcorr ftool”.

A further analysis of this issue will be undertaken, and hopefully we find the bug and can bring the anchor accuracy of all exposures to the same level.

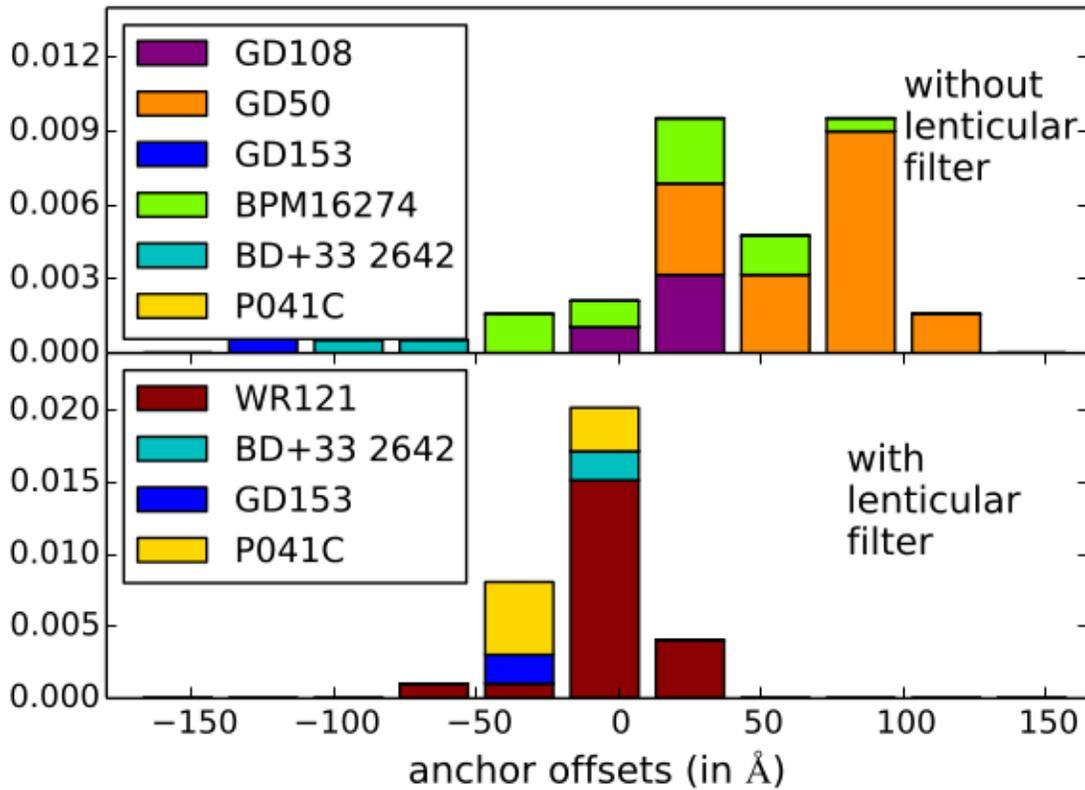


Figure 1.21: *Figure 2: 2014 wavecal verification:* This shows the 2014 verification of the wavelength shift due to anchor inaccuracy. It can be seen that in the absence of a lenticular filter the wavelengths are shifted much more, and that the offset depends on the source.

Technical Documents

- Swift UVOT Calibration report. Zemax optical models for the UV-Grism study: Zemax pixel scale factor and positioning. MSSSL 2008.
- Swift UVOT calibration report: Zemax optical models for the UV grism study. Second order wavelength scale for sources near the axis.

Summary of wavelength accuracy (2014 update)

The accuracy of the anchor error depends on the method used to find the anchor position in the grism. If the position was determined using the grism image only, the accuracy is worse. The best accuracy is obtained when the grism exposure was taken combined with a lenticular filter exposure.

Note: In August 2014 we found that the anchor error is nearly twice as large when no lenticular filter observation was included with the grism mode.

for the combination of grism + lenticular filter the accuracies are as follows:

UV grism nominal mode: For 95% of the spectra (2-sigma error) the anchor position will cause a shift in the wavelength scale of less than 35Å. The 2-sigma error to the wavelengths due to inaccuracies in the dispersion will be less than 18Å (2000-4500 Å).

UV grism clocked mode: A revision of the wave length calibration was released June 2013 with corrections for the shortest and longest wavelengths. The shift due to misalignment is less than 17Å for 95% of the spectra. The error in the wavelengths due to inaccuracies in the dispersion is within 11Å below 4500Å, and within 21Å for wavelengths between 4500-6000 Å.

visible grism nominal mode: The 2-sigma error in anchor positions is 30Å in the centre of the detector, 44Å overall. The 2-sigma error in the dispersion is less than 10Å.

visible grism clocked mode: The for 95% of the spectra the error in the anchor position is 44Å. The error in the dispersion less than 14Å.

for the grism without a lenticular filter the accuracies are larger::

UV grism nominal mode: For 95% of the spectra (2-sigma error) the anchor position will cause a shift in the wavelength scale of less than 53Å. The 2-sigma error to the wavelengths due to inaccuracies in the dispersion will be less than 15Å (2000-4500 Å).

UV grism clocked mode: The shift due to anchor misalignment is less than 47Å for 95% of the spectra. The error in the wavelengths due to inaccuracies in the dispersion is within 11Å below 4500Å, and within 21Å for wavelengths between 4500-6000 Å.

visible grism nominal mode: The 2-sigma error in anchor positions is 88Å. The 2-sigma error in the dispersion is less than 10Å.

visible grism clocked mode: The for 95% of the spectra the error in the anchor position is 118Å. The error in the dispersion less than 16Å.

The Zemax optical model

The Swift UVOT grism calibration uses of the Zemax optical model which was used to design the grisms for the XMM/OM and the Swift/UVOT instruments to supplement calibration observations. We have established that the model is valid and useful in various ways. The model has been compared the pre-launch ground calibration observations, to observations at and away from the boresight of the instrument for the first order and occasionally for the second order or zeroth order data. However, since the Zemax model does not include the fibre taper optics between the MCP and the CCD, small corrections to the model were needed which were derived based on observations of the emission line spectra of WR stars.

The comparison between the original design model and the actual instrument data suggests that the mounting of the grisms was a few degrees rotated away from the design as is evident of the angle of the spectra on the detector. The correction, based on the angles of the grism spectra on the detector in the nominal and clocked modes, is 3.8 ± 0.2 degrees for the UV grism and 2.6 degrees for the V grism. For the computation of the nominal and clocked modes a tilt was applied to the grism assembly model, and an offset for the clocked modes, corresponding to the position angle

of the grism in the filterwheel, and the effect of the clocking action respectively. The tilt angles used were 61.2 deg for UV nominal, 54.665 deg for UV clocked, 57.9 deg for the visible nominal, and 50.6 deg for the visible clocked grism.

Those have been the only corrections made to the original zemax grism optical design model. The original model for the UV grism was optimised for the 260nm wavelength in the first order; that for the V grism for 420nm in first order. These have been used (and referred to) as the anchor point of the spectra in the wavelength calibration described here. The spectra of (telescope) on-axis sources give the *boresight* spectrum where the anchor wavelengths in first order fall near the centre of the detector.

The Zemax model use has limitations, one is caused since the fiber taper part of the detector is not included in the model. The fibre taper results in some image distortion. To some extent this is taken out by using the distortion measured for the lenticular filters. For spectral analysis and its calibration the ‘detector image’ has therefore been used. The detector image is created by applying the distortion correction from the lenticular filter to the raw grism image. However, that process may in turn introduce a small distortion component due to the a possible distortion in the lenticular filters.

Relation to the 2005-2007 calibration

The instrument boresight is known to vary slightly for the different lenticular filters and is not a well-defined location on the detector for a spectrum. The boresight of the grisms in the first calibration was determined as a certain point in the zeroth order, whose location is defined by an algorithm based on the positions zeroth order peak emission for weak sources in the image as implemented in the “UVOTGRASPCORR” “Ftool”. The values for the UVOT grism boresight in the “CALDB” “TELDEF” files are for the zeroth order.

Several limitations in the first wavelength calibration have been resolved in this calibration.

- The first is, that all first order spectra on the grism image can now be calibrated rather than just those near the default position.
- The second improvement is, that the wavelength scale accuracy has improved by providing a solution valid over the whole detector.
- The third improvement is using the optical model.

Anchor point offset errors cause a wholesale shift of the wavelength scale. Anchor point offsets have different error sources in the two methods. Using a lenticular filter taken in combination with the grism gives error sources in the aspect correction of the lenticular filter less than a pixel, and in the transfer to the grism detector image, which is typically one to three pixels. When using the “UVOTGRASPCORR” tool the error sources are the accuracy with which “UVOTGRASPCORR” determined the attitude, which has also a random error of around 3 pixels, but shows also an unexplained systematic error which seems to depend on the particular star field. One in around twenty images tends to have larger offsets. That can be due to satellite drift during the grism plus lenticular filter observation when attitude lock of the satellite is not good, or failure of “UVOTGRASPCORR” to match the correct zeroth orders to the USNO-B1 catalog source positions.

Some colour dependence may be present in the UVOTGRASPCORR aspect solutions: The zeroth order suffers for the brighter sources from a combination of modulo-8 noise and saturation. The dispersion of the zeroth order is very non-linear with most of the red part of the spectrum on a few pixels, and the blue part in a weak extended tail. Although using only weak sources in the image will preferentially select the red peaks, some colour dependence may exist in the aspect correction of the grism image.

By using the optical model, scaled to have a best fit to the calibration observations, we obtain the dispersion and anchor position over the detector. The model also provides data on the zeroth, second and third orders. Only the second order has been calibrated for anchor and dispersion. The multi-order data is essential for understanding the spectra, since the second and third order fall generally on top of the longer wavelengths in the first order. A consistent model prediction of the variation of the angle of the spectral orders on the detector is used, though small corrections for curvature of the spectra were not predicted and have been measured from the calibration spectra. Finally, the point spread function of incoming radiation as function of wavelength, order and position is predicted by the model, and the predicted PSFs are useful for these are not nice gaussians, but rather elongated and horse-shoe or donut shaped for the longer wavelengths.

Relevant Documents

- description UVOT instrument: Roming et al. 2005, see also the XMM-OM description: Mason et al. 2001.
- SWIFT-UVOT-CALDB: Swift UVOT Grism Clocking, Alice Breeveld, 19th October 2005, Revision #01, Swift UVOT Calibration Documents Version 06-Apr-2006
- SWIFT-UVOT-CALDB: Teldef Files, Alice Breeveld, 19th October 2005, Revision #01, Swift UVOT Calibration Documents Version 06-Apr-2006

1.5.2 Coincidence loss in the UVOT

As explained in the description of the UVOT MIC detector, there is a possibility that counts incident on the detector are not counted when multiple photons are coincident on the same area on the detector.

The data are being acquired by reading out the CCD every frame time, which for the grism images is every 0.0110329 seconds. Typical exposure times range from 100-1000 seconds, which means there are a large number of frames in an exposure. Due to the finite number of frames in any exposure the statistics of incident photons, which follow a binomial distribution, can be treated for the limiting case of a Poisson distribution. The position of the incident photons can be determined to 1/8 of a physical pixel on the CCD by the fast on-board centroiding of the amplified photon splash. The coincidence loss affects measurement of photons falling in a common area on the detector, the coincidence area (short coi-area) of certain size and geometry.

Coincidence loss theory

Let's define the counts per frame by the letter z , and the count rate in counts per second by the letters Cr , and the frame time as t_f . We have then,

$z_i = Cr t_f$ <p style="text-align: center;">(1.1)</p>	eq.(1)
---	--------

With z_i the incident counts per frame per coi-area. If we have many frames, with a mean rate in counts per frame z , the probability of having k counts per frame incident on a coincidence area is therefore described by Poisson's equation:

$P(z_i, k) = z^k \exp(-z_i)/k!$ <p style="text-align: center;">(1.2)</p>	eq.(2)
--	--------

where $k!$ is the factorial of k . So there is a probability of $\exp(-z_i)$ of no detection ($k = 0$) in a frame. If we measure:

$z_m = Cm/N$ <p style="text-align: center;">(1.3)</p>	eq.(3)
---	--------

counts counts per frame, where Cm is the measured count rate, and N the number of frames, this measurement equals the probability of any detection, so we get

$z_m = (1 - \exp(-z_i))$ <p>(1.4)</p>	eq.(4)
---------------------------------------	--------

However, the count rate in the UVOT is corrected for the detector dead-time (the time taken to read out the data from the CCD, which means only during a fraction $a = 0.984$ of the time, incoming photons hit the coi-area on the detector.

Now we can derive the relation between measured counts per frame and the incident photons per frame as

$z_i = (-1/a) \ln(1 - z_m)$ <p>(1.5)</p>	eq.(5)
--	--------

where \ln is the natural logarithm.

Determination from observations

The relation of equation 5 can be converted into that for the count rate per second using equation 3.

For astronomical calibration sources with a known flux we can convert that flux into a prediction of the count rate by dividing by the photon energy. Once we measure that source we can relate the measured count rate to the predicted one. The following four figures show that for the four UVOT grism modes. Since one particular source only covers a limited region in count rates in the spectra, calibration sources for a range of brightness have been used.

Figure 1: observed and predicted rates

Figure 1.22: For the UV grism clocked mode, the observed count rate in a coi-area per frame is plotted against the incoming counts per frame. Since the predicted count rate derived from the calibration spectrum does not include the background, the background from the observed image was added after making an appropriate correction for the coincidence loss in the background. The theoretical relation from equation 5 is shown as a black line.

The coincidence loss area

The maximum number of counts per frame per coi-area is one. If the coi-area is too small, the observed count rates are too low; conversely, if the coi-area is too high, the observed count rates are too high. We discuss more in

The coincidence area determination in the Grisms

Early work on the MIC detectors at UCL ([Fordham et al.](#)) showed that the coincidence loss could be corrected for flat illumination by computing the observed rate over a certain area, which we will call here the coincidence area, or in short coi-area. They further showed that the same approach can be used to correct for point sources. They report that the coi-area can be dependent on the detector.

The UVOT has redundancy in its filter wheel and detector assembly. Since launch one unit has been used and calibrated, which is known as ‘uvota’. For point sources the coincidence loss correction is performed using a circular area of 10 pixels centered on the source. [Poole et al.](#) have shown that for this size of the coi-area the photometry was least affected (i.e., had the smallest error) by coincidence loss. It should be mentioned that the choice of the coi-area is also relevant to the determination of the instrumental zero points of the photometry, as a larger coi-area would include a larger part of the source photons which are falling off according to the point spread function. In [Breeveld et al.](#) it was

shown that this also gives a good correction for the background, unless the background is exceptionally high, in which case a slightly larger coi-area works well.

In spectra the detector illumination extends over many pixels, far more than the three used in the centroiding of events. This will tend to increase the coincidence loss. The physical reasoning behind this is that the centroiding of a photon splash in a frame will pick the highest peak. Statistical noise, and the linear spectrum in the grism images implies that in some frames neighbouring pixels will have more chance to cause coincidence loss than would happen in a point source. Based on this simple picture we would expect a larger coincidence loss area for grism spectra than for a point source.

Some new questions arise that were no concern for a point source. One question is how far the effects of coincidence loss reach along and across the spectrum? Another question is coincidence loss in a bright emission line which is seen to affect the spectrum over some distance. That distance may be an indication of the extent of the coincidence loss area along the spectrum.

Figure 1.23: *This figure shows the effect of coincidence loss on a very bright smooth input spectrum. The blue spectrum shows the DA White Dwarf GD153 in the visible grism, the green spectrum in the uv grism. The net count rates are shown where the background has been removed.*

Figure 1 When we consider the spectra of a series of sources with increasing brightness, it becomes clear that spectra exceeding a certain brightness level show instrumental peaks and vales. It is thought that the high brightness affects the on-board centroiding which has been fine tuned for a smoother brightness variation. The photons at high coincidence loss are mis-registered and appear as bright points in a modulo-8 pattern in the grism image. As the spectrum lies at an angle over the pixel grid, the brightness distribution of the extracted spectrum of a very bright source shows this variation, even when the source spectrum is smooth. This effect is most prominent at the bright part of the spectrum where the grism effective area peaks. For example, spectra taken from GD153, a DA white dwarf star, see figure [Figure 1](#) for this. That spectrum is quite typical for the variations in spatial frequency. The pattern is more pronounced in the visible grism since it is more sensitive than the uv grism.

For the determination of the coincidence loss, we need to average over this pattern. Bright spectral lines show that the coincidence loss is quite extended. After some trials, we decided to use a length L_{coi} along the spectrum determined approximately from 24 subpixels divided by the cosine of the typical angle of the spectrum on the detector for each of the grism modes. This averages the count rate over most of the variations.

Table: Coincidence loss area box; length, width, and multiplier

Grism mode	L_{coi} pixels	W_{coi} pixels	coi-multiplier m_{coi}
uv nominal	27	15	1.12 ± 0.05
uv clocked	29	15	1.11 ± 0.05
visible nominal	28	14	1.13 ± 0.03
visible clocked	31	13	1.09 ± 0.03

The coincidence area A_{coi} is most simply represented by a box of length L_{coi} and width W_{coi} , multiplied with a correction factor m_{coi} , named coi-multiplier,

$$A_{coi} = L_{coi} W_{coi} m_{coi} \tag{1.6}$$

The measured count rate for the coincidence loss correction is thus determined for that area, and the corresponding background is computed for an equivalently sized area.

Finally, a view of a bright spectrum on the detector.

Figure 2 This spectrum is on the upper brightness ranges of where the coincidence loss can be corrected for.

The coi-area is then used in correcting for *Coincidence loss in the UVOT*.

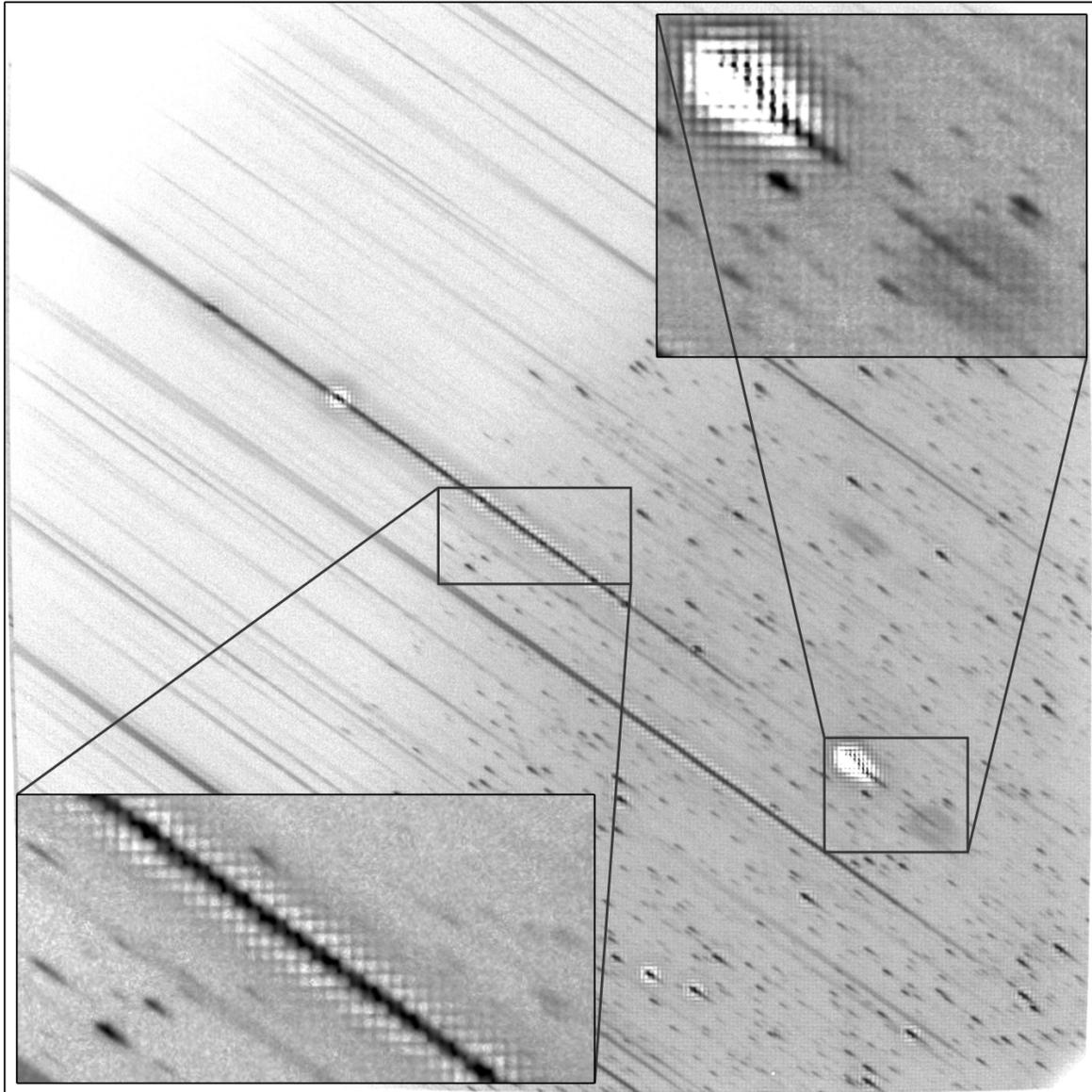


Figure 1.24: *The bottom inset enlarges a part of the first order spectrum of a bright source. Although not as bright as GD153, the modulo-8 pattern is clearly seen. The pattern around the bright emission lines show the extent of the coincidence loss of bright features along the spectrum.*

Figures 2,3,4: observed and predicted rates for the other detector modes

Figure 1.25: Same as figure *Figure 1: observed and predicted rates* for the UV grism nominal mode.

Figure 1.26: Same as figure *Figure 1: observed and predicted rates* for the Visible grism clocked mode.

Figure 1.27: Same as figure *Figure 1: observed and predicted rates* for the Visible grism nominal mode.

Some of the very bright sources deviated from the overall curve. This kind of problem is not completely unexpected because the coincidence loss distorts the spectrum very significantly.

Extremely bright stars can flood the detector so that a significant part of the spectrum saturates the detector in adjacent (physical) pixels. That means that the centroiding fails, since that requires one pixel to be brighter than its neighbors. Once the centroiding fails, no count will be registered. This leads the very brightest sources to register less than the maximum count rate per frame of one. We observe that an incident count rate per frame larger than about 7 starts showing a decreased observed maximum rate.

For studying the coi-effect the spectra were split into adjacent areas extending over a length L_{coi} . A range of width W_{coi} were chosen. For each area the average observed count rate was determined. That count rate differs from the count rate determined within the aperture of the spectrum, as it is used for calculating a correction factor for the coincidence loss only. We refer to this as the “observed coi count rate”.

1.5.3 The effective area of the grisms

The latest update of the effective area has been done after the coincidence loss treatment reached its final form. The changes to the effective area are minor ones, as could be expected, except in the visible grism, where the accuracy increased substantially, and at the end of the range where use of the new coi-correction allowed us to use very bright sources, thus increasing the accuracy of the effective area when smaller than 10% of its peak value.

Therefore the reader is invited to look at the description of the effective area in: *fluxcal_v1*, which was made using the previous version of the coincidence loss correction.

1.5.4 Verification of the Swift UVOT Grism Flux Calibration (2013)

The verification has been split up by grism mode. The plots of the spectra below are showing wavelength offsets consistent with the determination of the anchor accuracy in 2014 which has been discussed elsewhere. The visible grism observations were mostly done without a lenticular filter, while the uv grism observations were mostly done using a lenticular filter.

Note: The 2014 change from using an ad-hoc correction to a theoretically sound correction for the coincidence loss did not produce major changes, so the verification presented here is representative.

Verification of the flux calibration in the UV clocked grism

Overview

The new uv-grism flux calibration is valid over the whole detector, and makes also a correction for coincidence-loss. The first integrated software version is UVOTPY-0.9.7.0, but we will do the verification to the 0.9.7.1 version which has patches for the sensitivity-loss and flux model interpolation.

With the choice of data, we have an independent set from the spectra used to construct the effective area. Of course, spectra from the same targets are used.

Method

Calibration spectra that were not used in the flux calibration are reprocessed using the new flux calibration and then compared to the known stellar flux. The spectra for verification are located on the detector between the areas with the spectra used to determine the effective areas for the flux calibration.

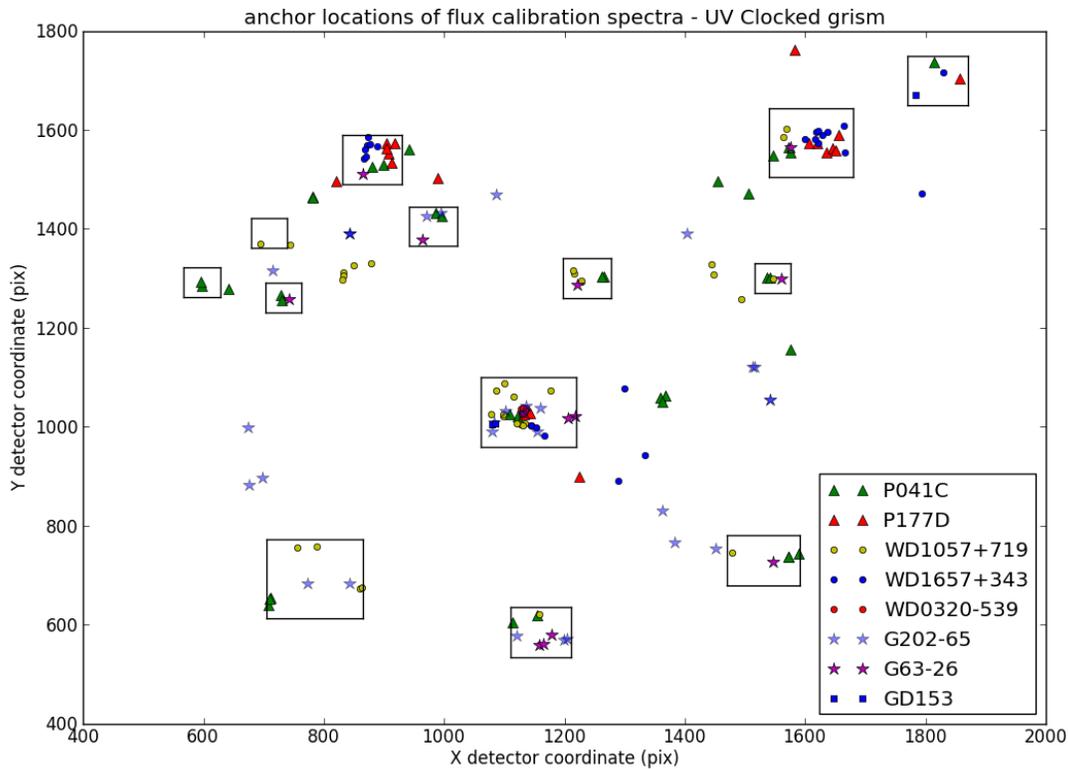


Figure 1.28: Figure 1: The anchor positions of the calibration spectra marked by target. The boxes are enclosing sources used to derive an effective area for the location of the box.

Data used for verification

The approximate location of the anchor of the spectra, their obsid, and number of the fits extension is listed. First the cool stars are listed, followed by the hot white dwarfs.

GSPC-P041C spectra (F0 V)

anchor	obsid	ext	% err	plot	aspect correction
781,1459	00057965002	1,2	24	<i>GSPC-P041C Figure 1</i>	yes,yes
641,1275	00057964002	2	18	<i>GSPC-P041C Figure 2</i>	yes
940,1553	00057972002	1	1	<i>GSPC-P041C Figure 3</i>	yes
1361,1051	00057960002	1,2,3	1.5	<i>GSPC-P041C Figure 4</i>	yes,all
1578,1151	00057969001	3	6	<i>GSPC-P041C Figure 5</i>	yes
1454,1494	00057958001	2	1	<i>GSPC-P041C Figure 6</i>	yes
1506,1464	00057958001	1	4	<i>GSPC-P041C Figure 7</i>	yes

GSPC P177D spectra (F0 V)

anchor	obsid	ext	% err	plot	aspect correction
1586,1756	00056762002	4	17	<i>GSPC-P177D Figure 1</i>	yes
993,1496	00056763002	6	1	<i>GSPC-P177D Figure 2</i>	yes
821,1491	00056763002	2	3	<i>GSPC-P177D Figure 3</i>	yes
1229, 889	00056760002	1	8	<i>GSPC-P177D Figure 4</i>	yes

The error is calculated as the mean of the flux difference over a good region of the spectrum, and is considered to be a measure of the error in the flux level. The flux was calibrated using `uvotpy.uvotio.rate2flux` default option with anchor position given.

WD1657+343 spectra

anchor	obsid	ext	% err	plot	aspect correction
1797,1467	00055903006	1	1	<i>WD1657+343 Figure 1</i>	yes
1338, 941	00055900050	1	4	<i>WD1657+343 Figure 2</i>	yes
1291, 891	00055900050	3	9	<i>WD1657+343 Figure 3</i>	yes
1302,1082	00055900050	2	5	<i>WD1657+343 Figure 4</i>	yes

WD1057+719 spectra

anchor	obsid	ext	% err	plot	aspect correction
833,1307	00055203007	1,2,3	11.3	<i>WD1057+719 Figure 1</i>	yes,all
833,1307	00055203008	1	13	<i>WD1057+719 Figure 1</i>	yes
876,1334	00055203007	4	2	<i>WD1057+719 Figure 2</i>	yes
1447,1324	00055201011	1,2	4	<i>WD1057+719 Figure 3</i>	yes,yes
1496,1255	00055201011	3	14	<i>WD1057+719 Figure 4</i>	yes

Results

Although the number in the table for the error is based on the mean, since the total exposure time was not long enough to beat down the noise enough, it appears that over most of the detector the flux calibration is within 10% or better. As expected, the flux is most uncertain near the sudden drop in response. The current calibration file (version 002) used with the current `uvotpy` version (0.9.7.1) give a fit to within about 15%. The largest deviations seem related to contamination of the spectrum by the spectra from other sources in the field, and of course from the second order.

I should mention that the limit on the errors is to some degree dominated by the noise in the data. The error in the flux calibration as determined from the calibrated areas (see the boxes in the figure above,) is more reliable since the total exposure time in those areas is larger.

The wavelength scale errors are quite apparent. Errors in the anchor position cause wholesale shifts of the scale, while the dispersion at long wavelengths is not correct (> 4000Å) at present. The zemax model which was used for the dispersion turned out to be not correct, and an update of the dispersion is planned. That in turn causes large swings in the comparison of the flux from the extracted spectra (which were not shifted) and the reference spectra. The anchor positions were determined using lenticular files taken before or after the grism exposure in the same *Swift* “mode”, meaning that no slew took place between grism and corresponding exposure in one of the lenticular filters.

However, occasionally drift of the pointing occurs, especially noticeable in the weaker spectra with more than 1000 second exposure times.

Figures for the uv grism : clocked mode

These figures go with the description *Verification of the flux calibration in the UV clocked grism*, and show in the top panel the observed flux and the reference spectrum which is usually taken by HST. The sources are calibration sources with the exception of G202-65.

In the bottom panel is plotted $(\text{observed} - \text{reference})/\text{reference}$ as a percentage. Notice that the bottom plot gets very noisy when the source flux is low. This is due to noise in the data. Also, at longer wavelengths second order contamination raises the observed flux, setting in at a wavelength that depends on the location of the blue rise of the spectrum.

GSPC-P041C spectra (F0 V)

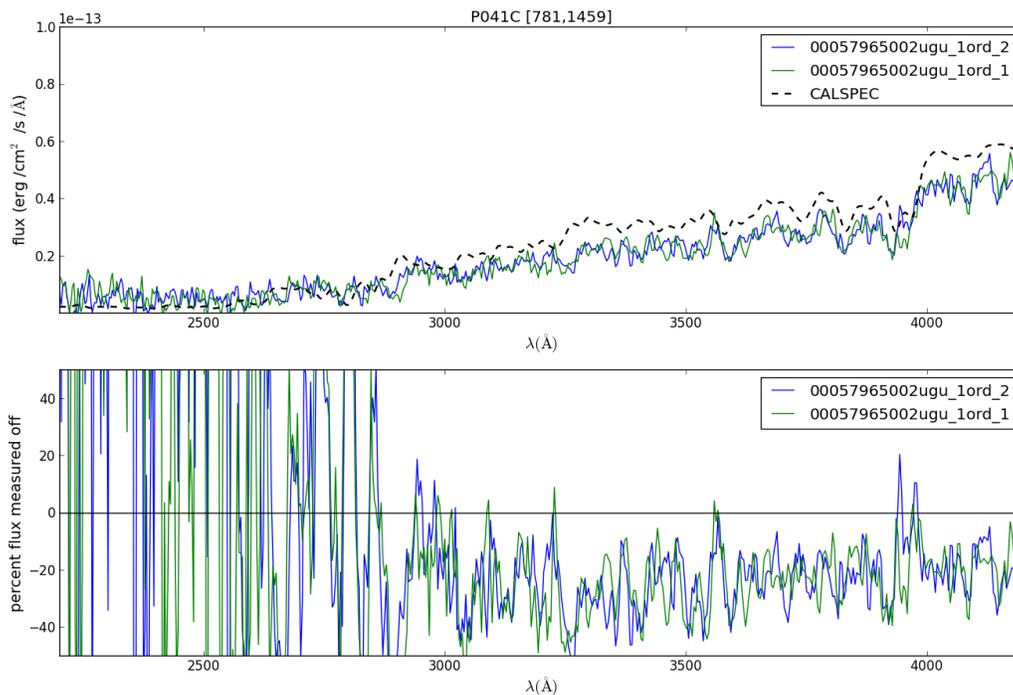


Figure 1.29: This spectrum with anchor at [781,1459] is affected by the decreasing sensitivity in this part of the detector.

GSPC-P041C Figure 1

GSPC-P041C Figure 2

GSPC-P041C Figure 3

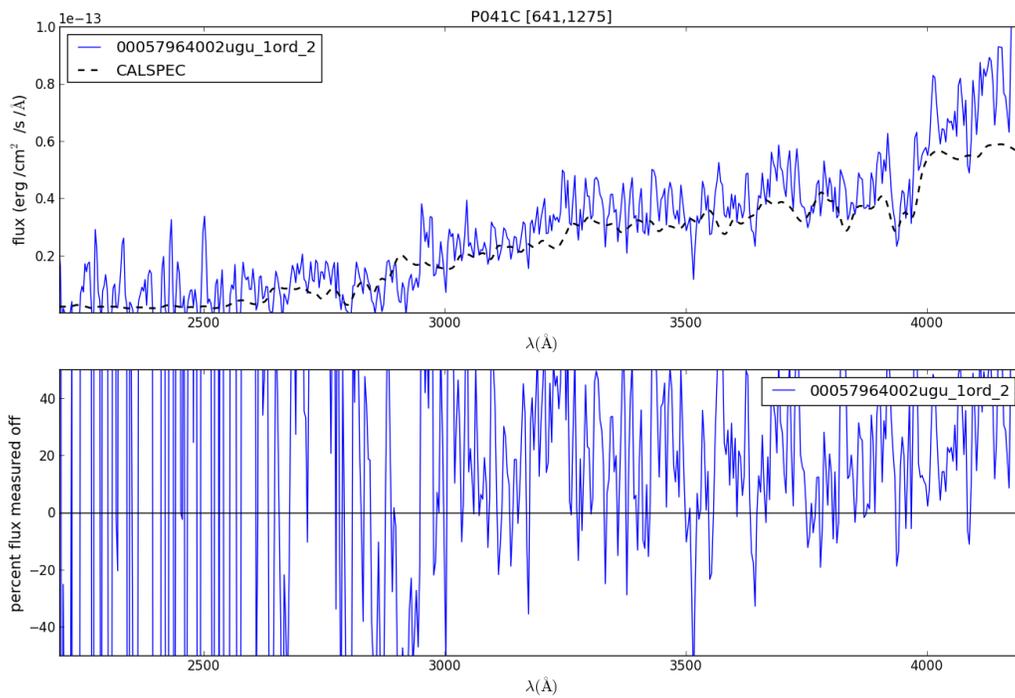


Figure 1.30: This spectrum with anchor at [641,1275] is affected by the decreasing sensitivity in this part of the detector.

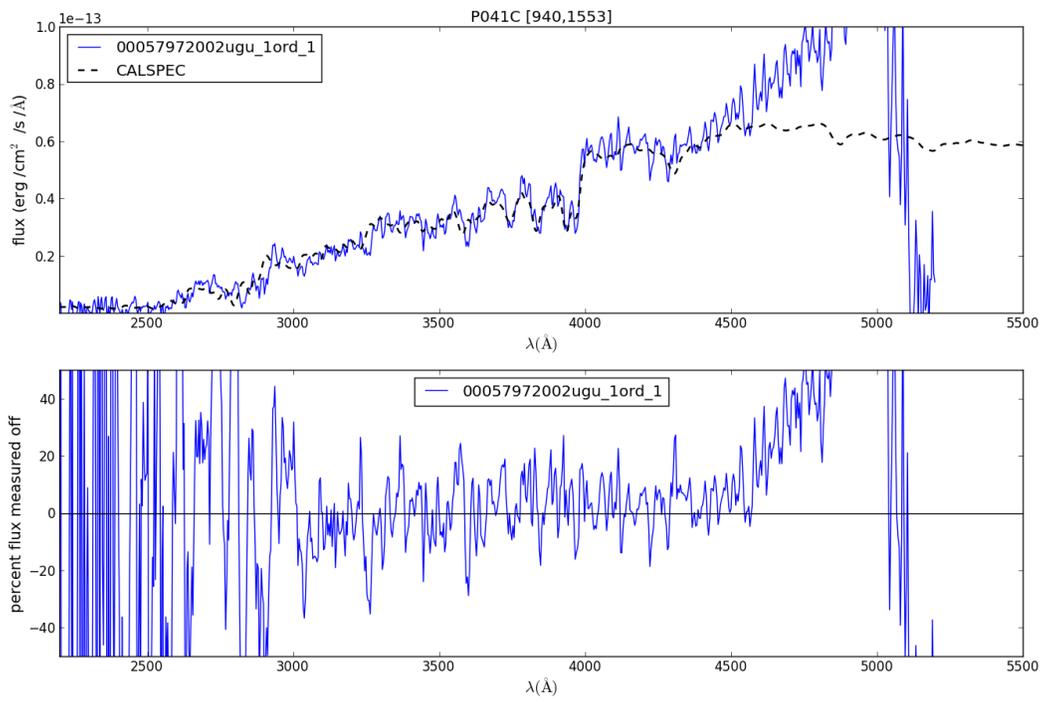


Figure 1.31: This spectrum has its anchor at [940,1553].

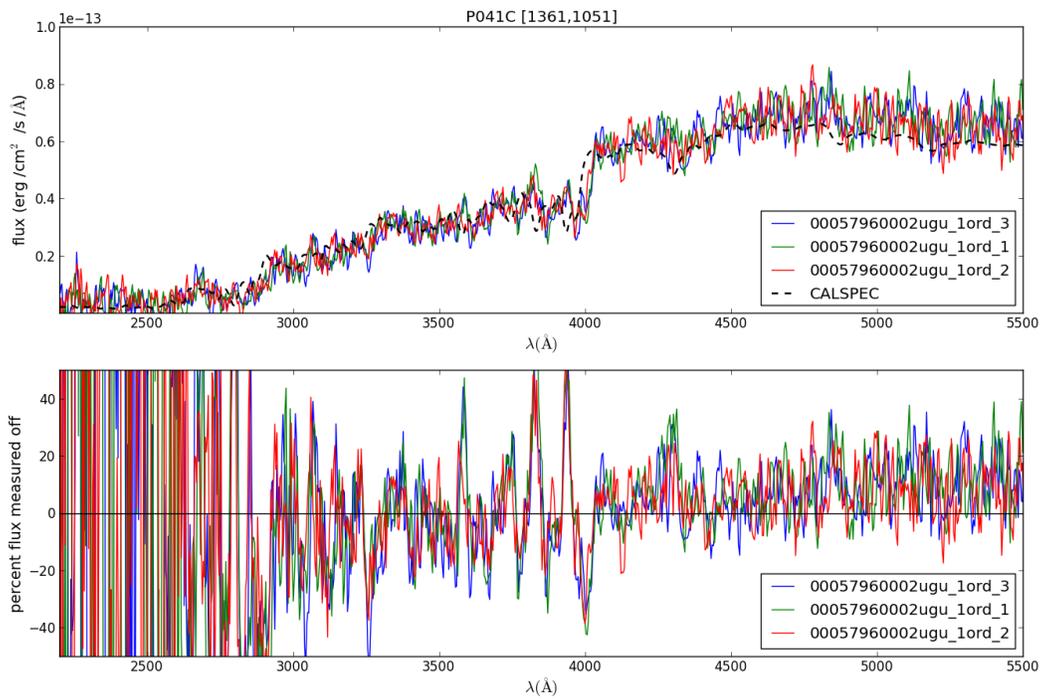


Figure 1.32: This spectrum has its anchor at [1361,1051].

GSPC-P041C Figure 4

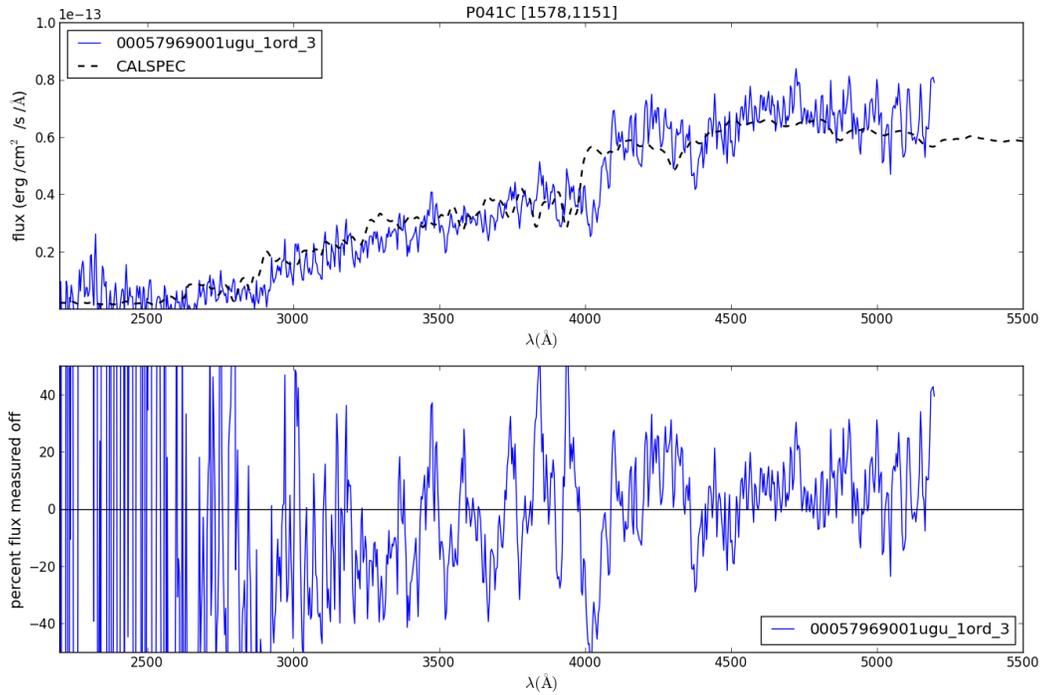


Figure 1.33: This spectrum has its anchor at [1578,1151].

GSPC-P041C Figure 5

GSPC-P041C Figure 6

GSPC-P041C Figure 7

GSPC-P177D spectra (F0 V)

GSPC-P177D Figure 1

GSPC-P177D Figure 2

GSPC-P177D Figure 3

GSPC-P177D Figure 4

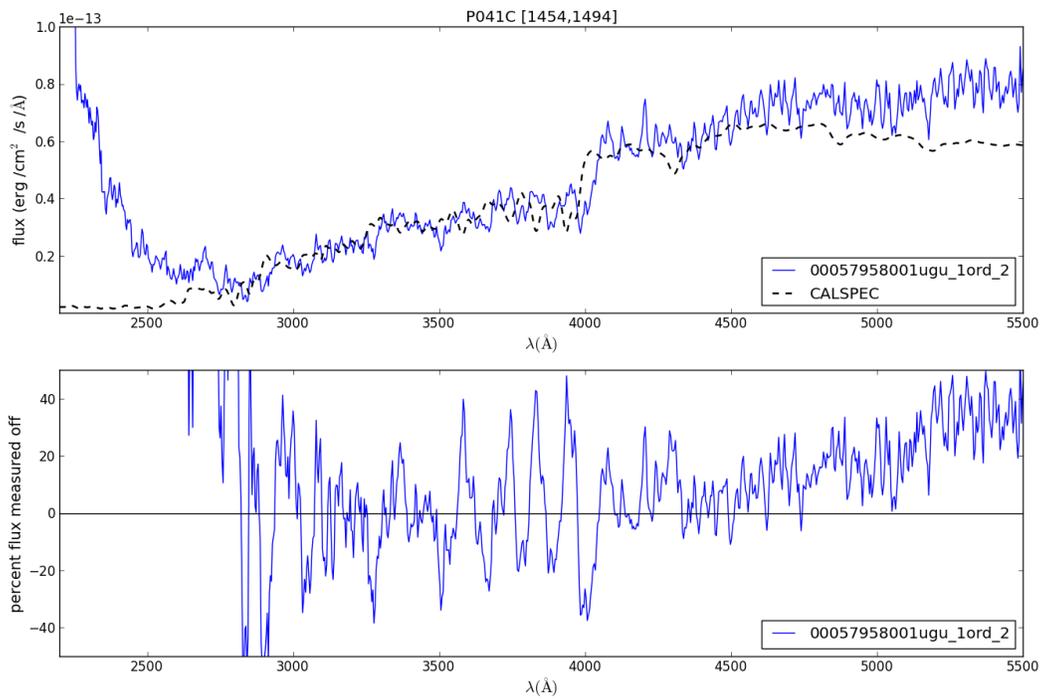


Figure 1.34: This spectrum has its anchor at [1454,1494].

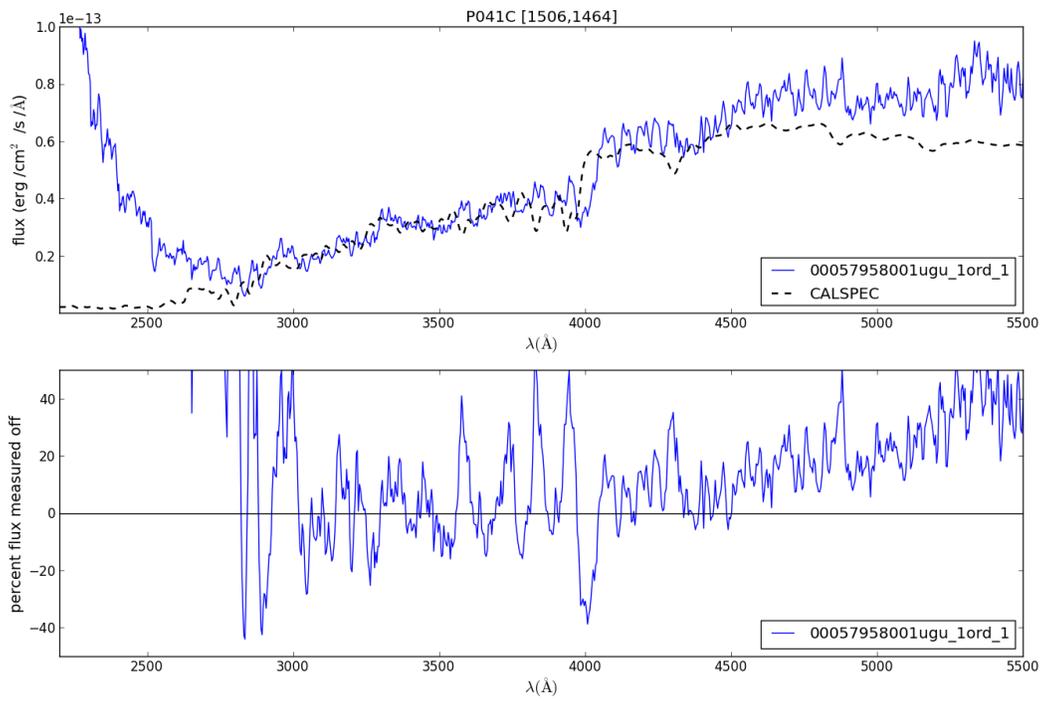


Figure 1.35: This spectrum has its anchor at [1506,1464].

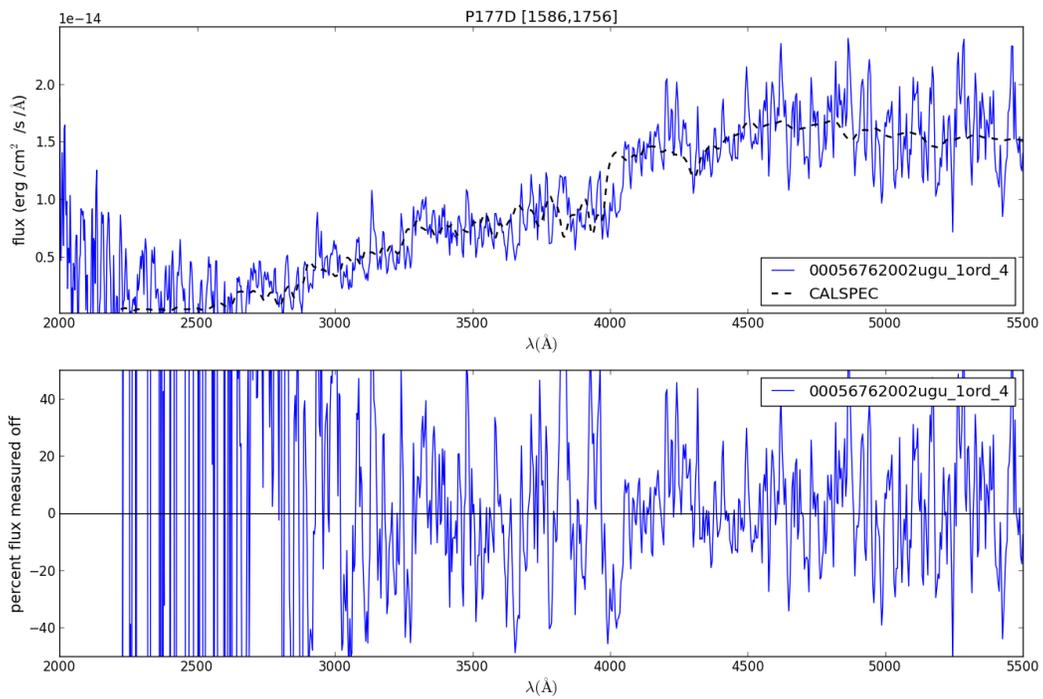


Figure 1.36: This spectrum has its anchor near [1586,1756].

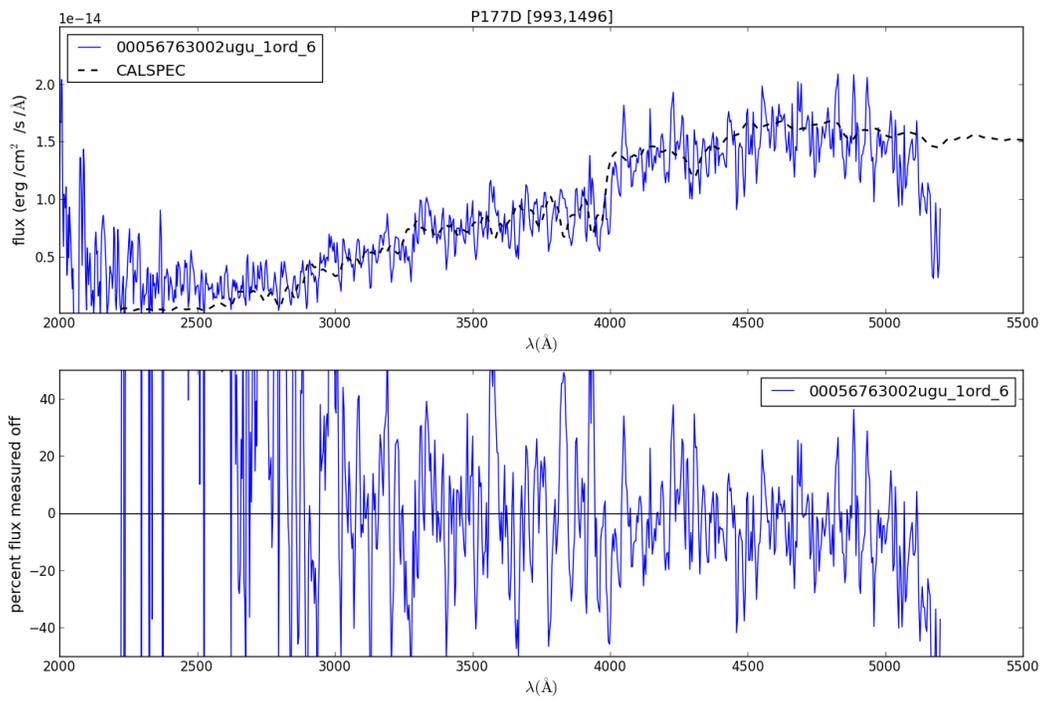


Figure 1.37: This spectrum has it anchor near [993,1496].

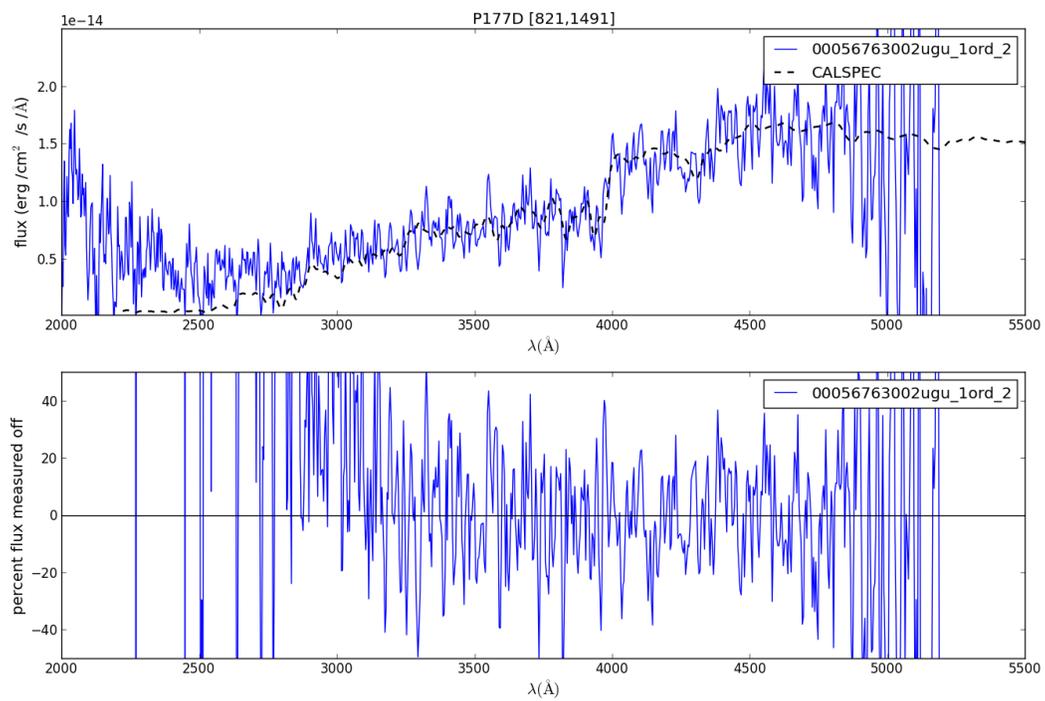


Figure 1.38: This spectrum has its anchor near [821,1491].

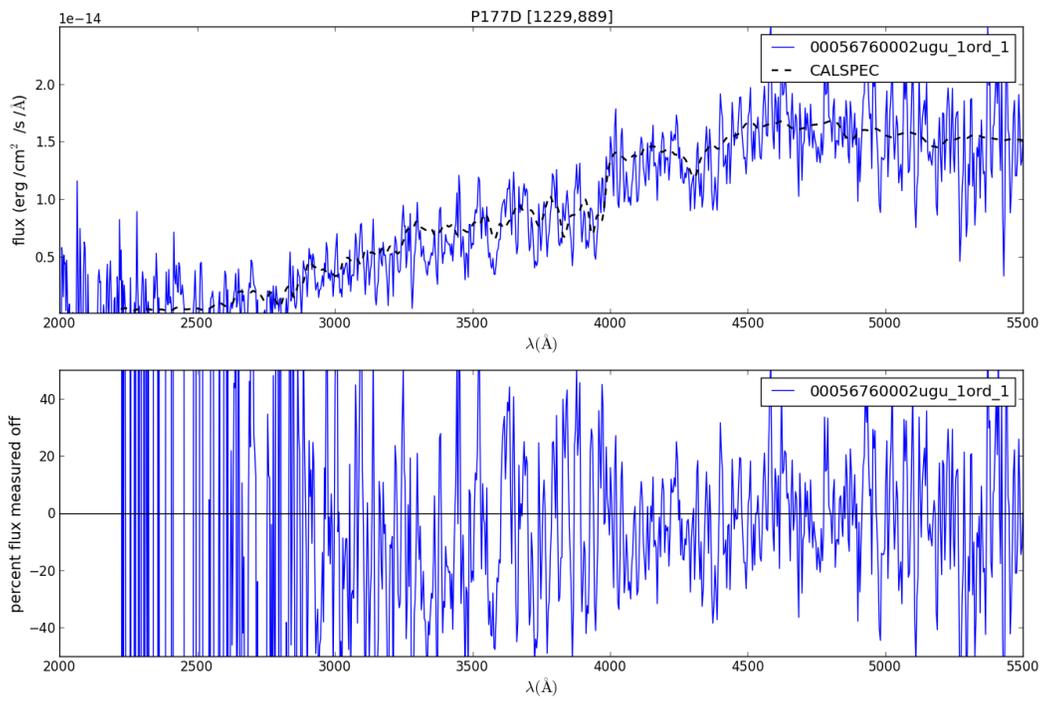


Figure 1.39: This spectrum has its anchor near [1229,889].

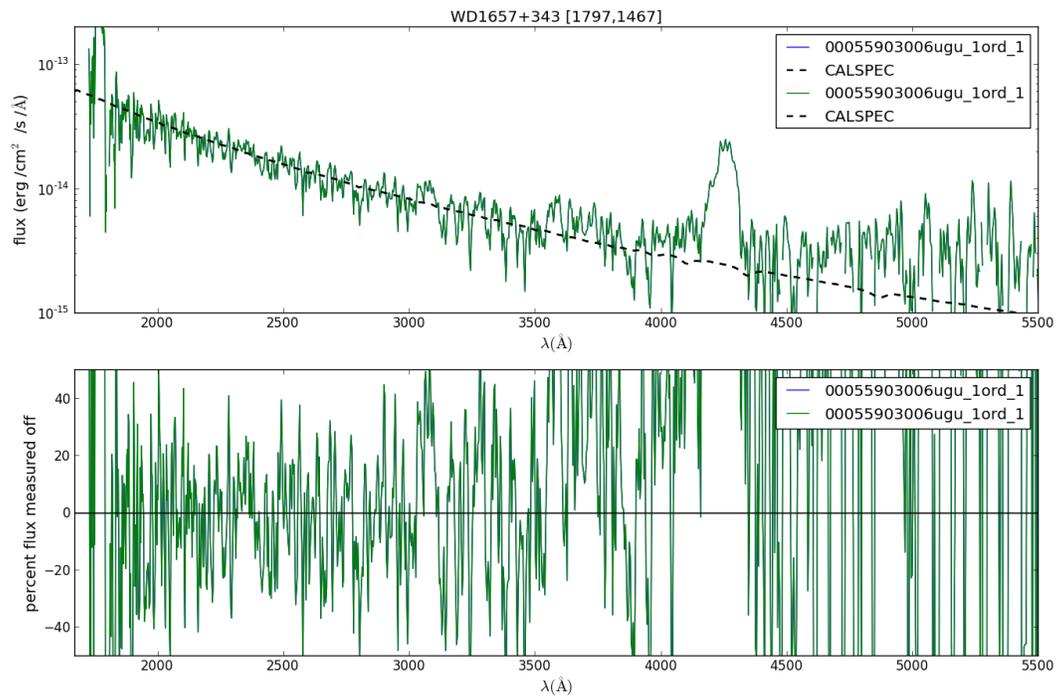


Figure 1.40: This spectrum has its anchor near [1797,1467]. A large zeroth order is present contaminating the 4300 \AA region.

WD1657+343 spectra (DA)

WD1657+343 Figure 1

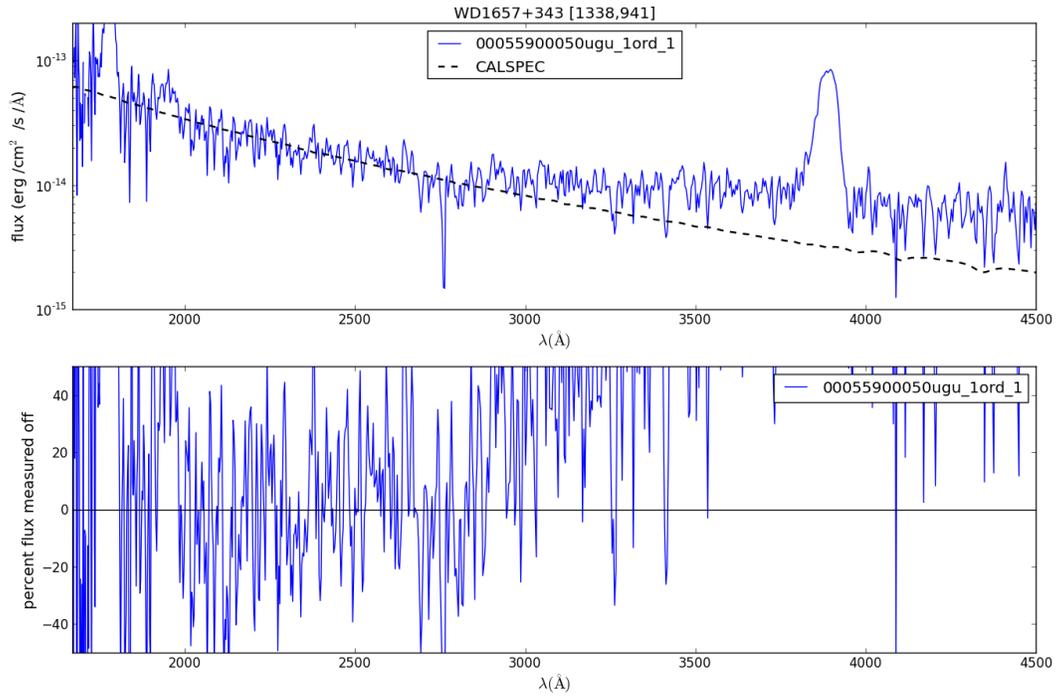


Figure 1.41: This spectrum has its anchor near [1338, 941].

WD1657+343 Figure 2

WD1657+343 Figure 3

WD1657+343 Figure 4

WD1057+719 spectra (DA)

WD1057+719 Figure 1

WD1057+719 Figure 2

WD1057+719 Figure 3

WD1057+719 Figure 4

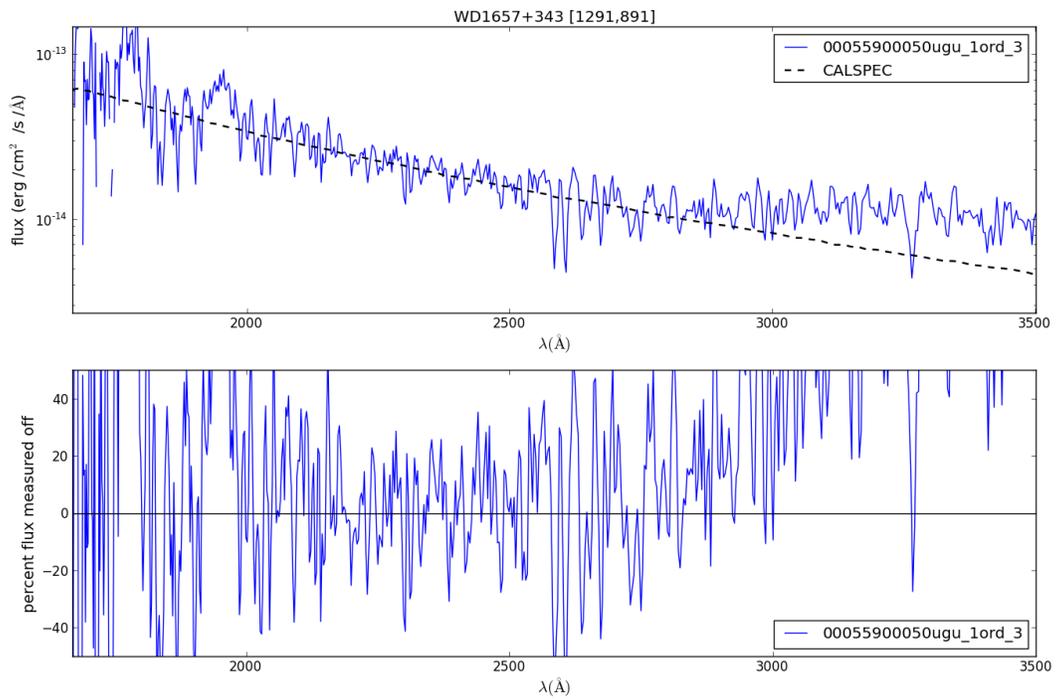


Figure 1.42: This spectrum has its anchor near [1291, 891].

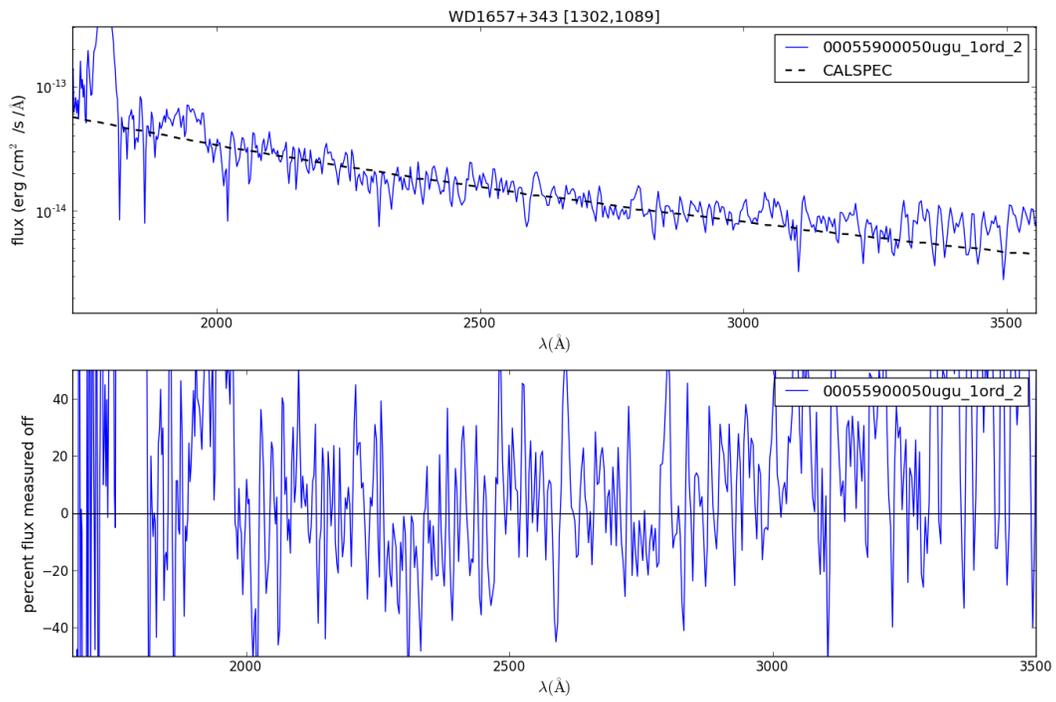


Figure 1.43: This spectrum has its anchor near [1302,1082].

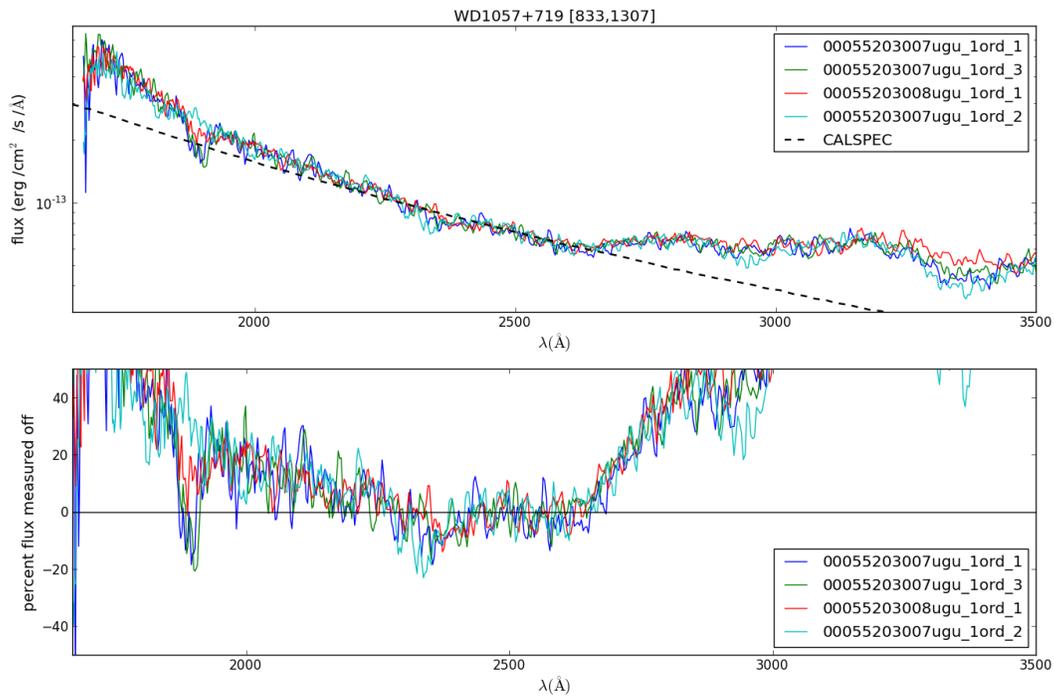


Figure 1.44: This spectrum has its anchor near [833,1307]. This spectrum is located near the region of the detector with reduced response. Notice the dip in response near 1800 \AA happens in two locations on the detector and which appear to have a slightly reduced sensitivity. These are only seen very occasionally, and do not have a photometric signature of more than 1%. Very few spectra have been taken at those locations. In the grism, with coincidence loss amplifying differences and background subtraction doing it as well, this appears to be more significant than for the UVOT photometry, at least for bright sources. The issue is still under investigation.

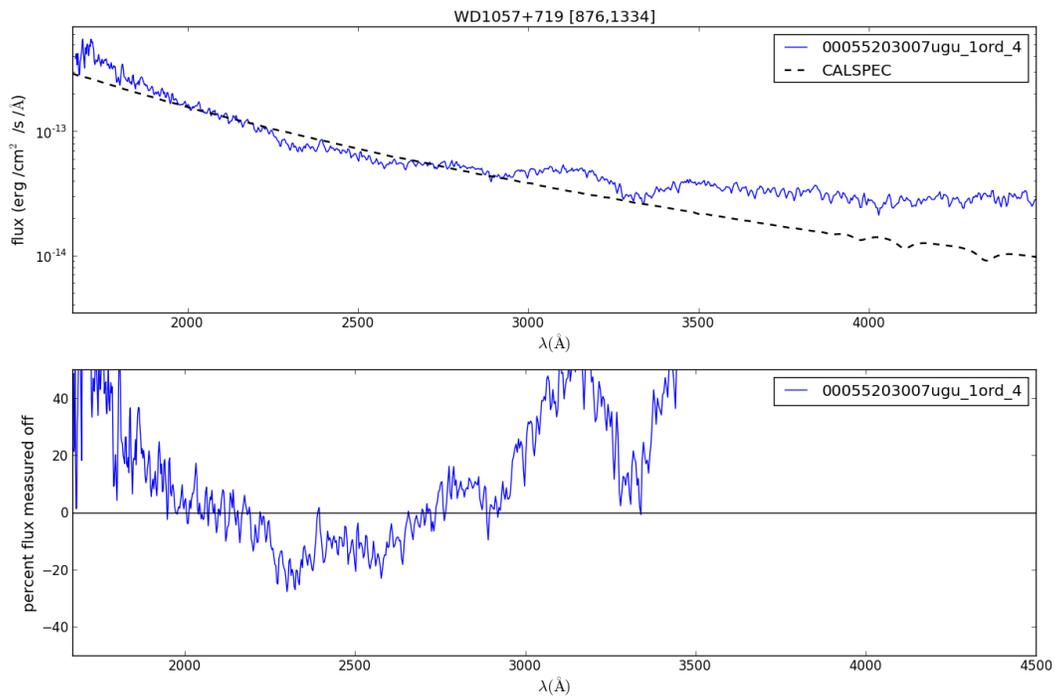


Figure 1.45: This spectrum has its anchor near [876,1334]. This spectrum is located near the region of the detector with reduced response.

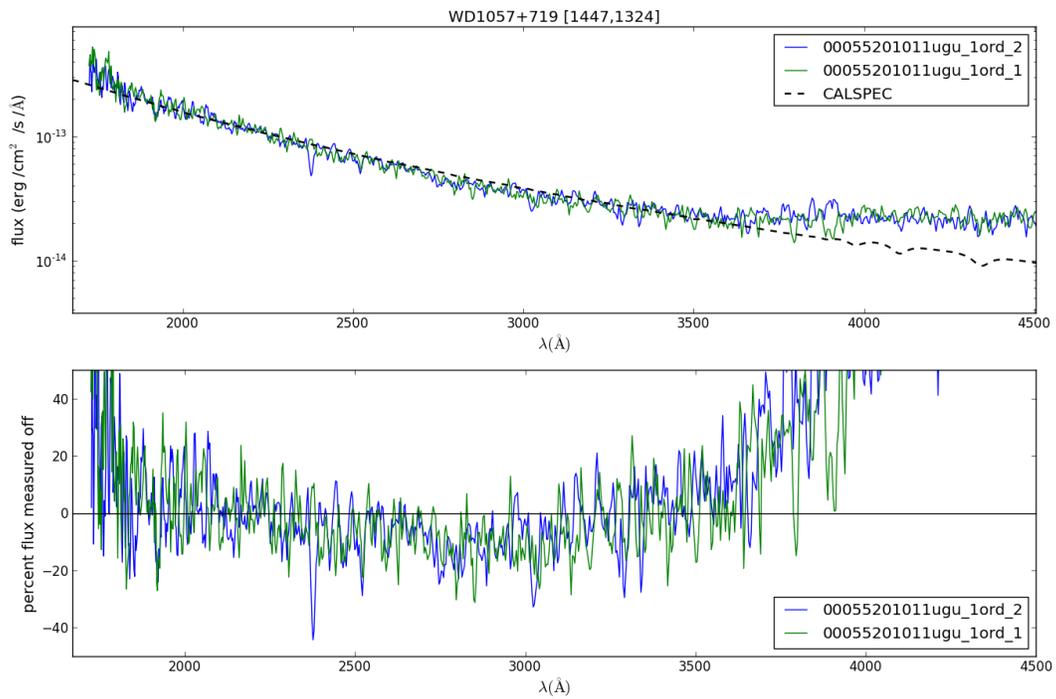


Figure 1.46: This spectrum has its anchor near [1447,1324.]

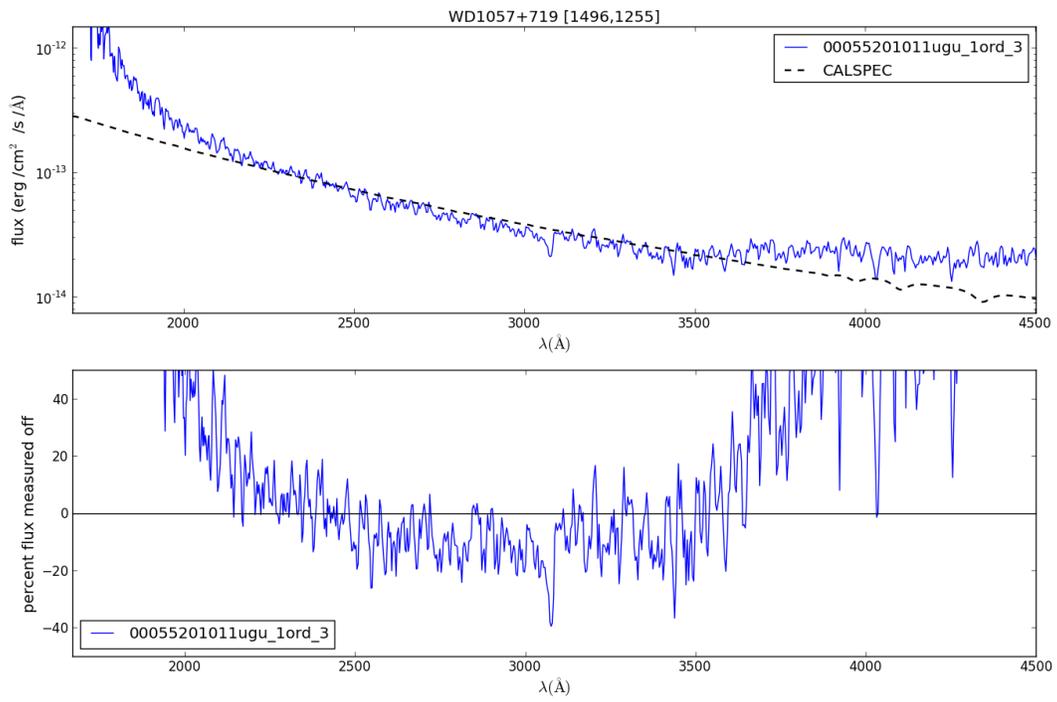


Figure 1.47: This spectrum has its anchor near [1496,1255].

Verification of the flux calibration in the UV nominal grism

Overview

The new uv-grism flux calibration is valid over the whole detector, and makes also a correction for coincidence-loss. The first integrated software version is `UVOTPY-0.9.7.0`, but we will do the verification to the 0.9.7.1 version which has patches for the sensitivity-loss and flux model interpolation.

With the choice of data, we have an independent set from the spectra used to construct the effective area. Of course, spectra from the same targets are used.

Method

Calibration spectra that were not used in the flux calibration are reprocessed using the new flux calibration and then compared to the known stellar flux. The spectra for verification are located on the detector between the areas with the spectra used to determine the effective areas for the flux calibration.

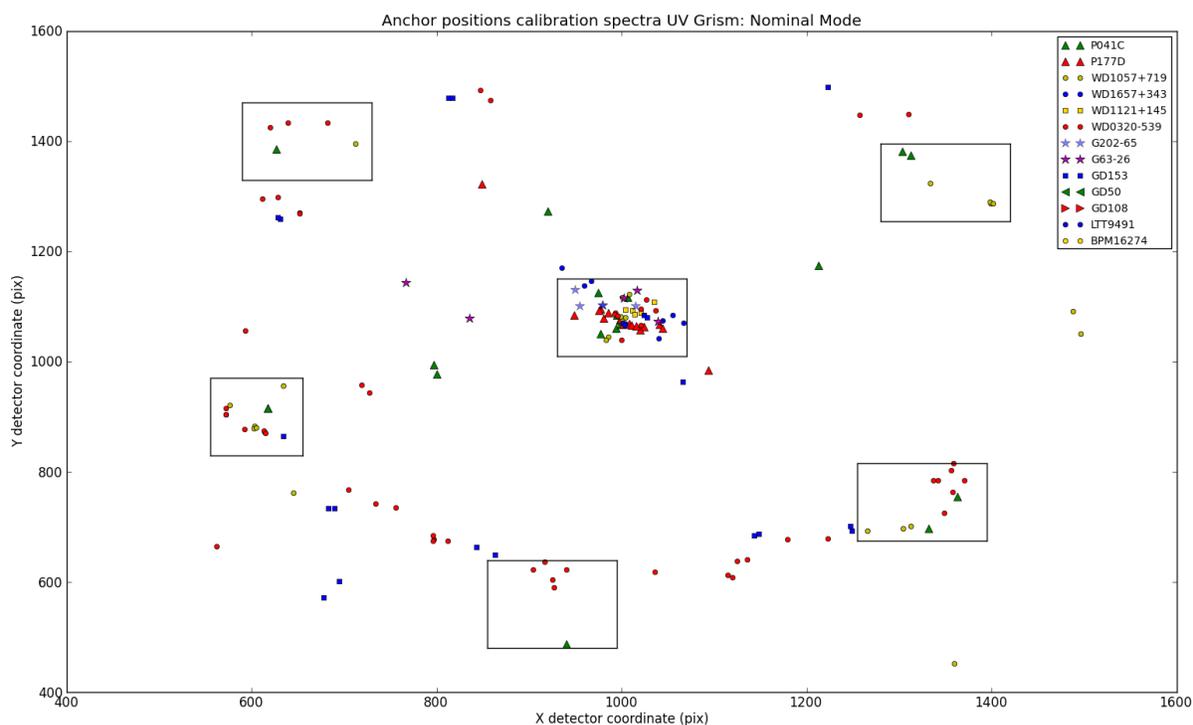


Figure 1.48: Figure 2: The anchor positions of the calibration spectra marked by target. The boxes are enclosing sources used to derive an effective area for the location of the box.

Data used for verification

The approximate location of the anchor of the spectra, their `obsid`, and number of the `fits` extension is listed. First the cool stars are listed, followed by the hot white dwarfs.

GSPC-P041C spectra (F0 V)

anchor	obsid	ext	% err	plot
800, 980	00057953006	1	12	<i>GSPC-P041C Figure 8</i>
800, 980	00057953003	1	14	<i>GSPC-P041C Figure 8</i>
920,1273	00057951006	1	12	<i>GSPC-P041C Figure 9</i>
1213,1175	00057954004	1	6	<i>GSPC-P041C Figure 10</i>

GSPC P177D spectra (F0 V)

anchor	obsid	ext	% err	plot
850,1320	00056760022	5	11	<i>GSPC-P177D Figure 5</i>

WD1057+719 spectra

anchor	obsid	ext	% err	plot
645, 762	00055201005	1	7.4	<i>WD1057+719 Figure 5</i>
1490,1066	00055201009	1,2	30	<i>WD1057+719 Figure 6</i>
1360, 462	00055203003	2	9	<i>WD1057+719 Figure 7</i>

WD0320-539 spectra

anchor	obsid	ext	% err	plot
562, 665	00054254008	2	16	<i>WD0320-539 Figure 1</i>
1310,1449	00054251047	1	5	<i>WD0320-539 Figure 2</i>
850,1480	00054252003	1,2	7	<i>WD0320-539 Figure 3</i>

GD153 spectra

anchor	obsid	ext	% err	plot
685, 590	00055501008	1	15	<i>GD153 Figure 1</i>
685, 590	00055501010	1	15	<i>GD153 Figure 1</i>
610,1240	00055503026	1,2	15	<i>GD153 Figure 2</i>
1145, 685	00055503010	1	7	<i>GD153 Figure 3</i>
1145, 685	00055503012	1	7	<i>GD153 Figure 3</i>

Result of the verification

The errors are typically 10% or less. The response below 1750Å shows some inconsistencies which is partly due to the low sensitivity, partly to the coincidence-loss correction as can be seen in GD153, which has a correction of about 50% due to coincidence-loss in the bright part of the spectrum.

Figures for the uv grism : nominal mode

These figures go with the description *Verification of the flux calibration in the UV nominal grism*, and show in the top panel the observed flux and the reference spectrum which is usually taken by HST.

In the bottom panel is plotted (observed - reference)/reference as a percentage. Notice that the bottom plot gets very noisy when the source flux is low. This is due to noise in the data. Also, at longer wavelengths second order contamination raises the observed flux, setting in at a wavelength that depends on the location of the blue rise of the spectrum.

GSPC-P041C spectra (F0 V)

There are some zeroth orders from field sources present in some of the spectra; easily identified since they show up as a large peak being different from the reference spectrum.

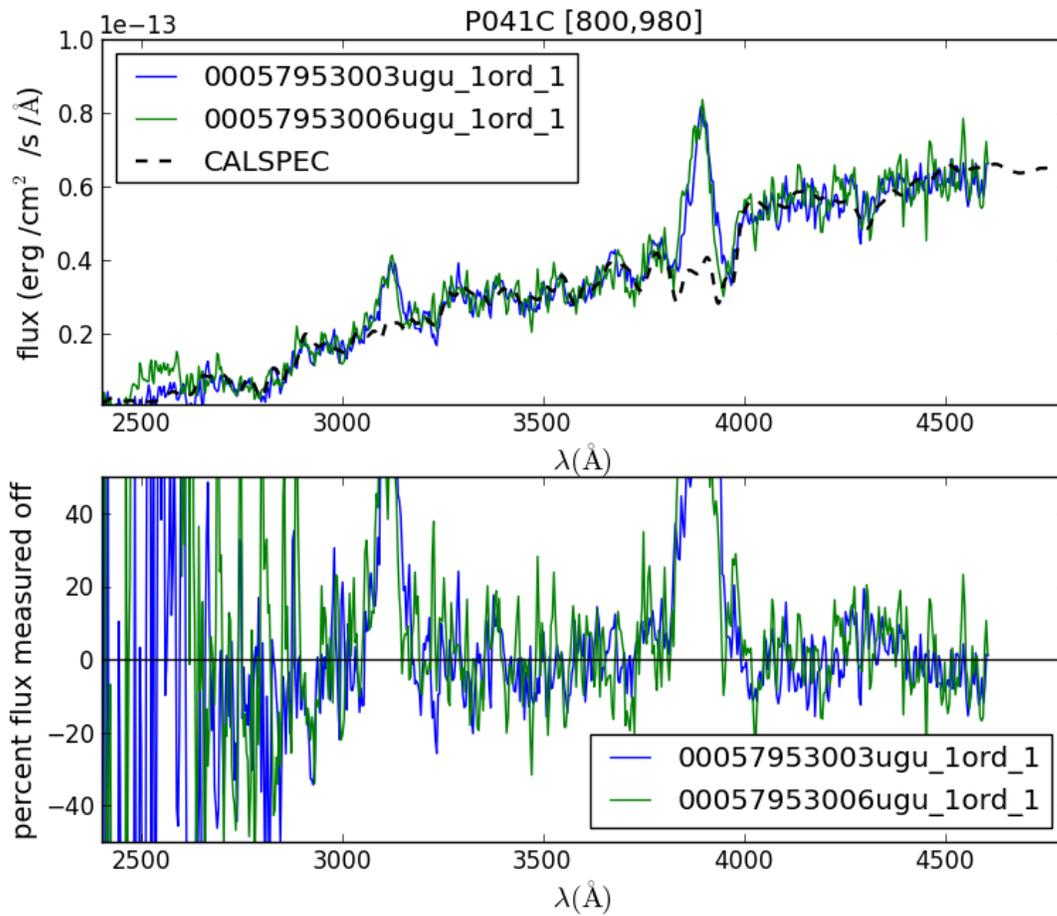


Figure 1.49: This spectrum with anchor at [800,980].

GSPC-P041C Figure 8

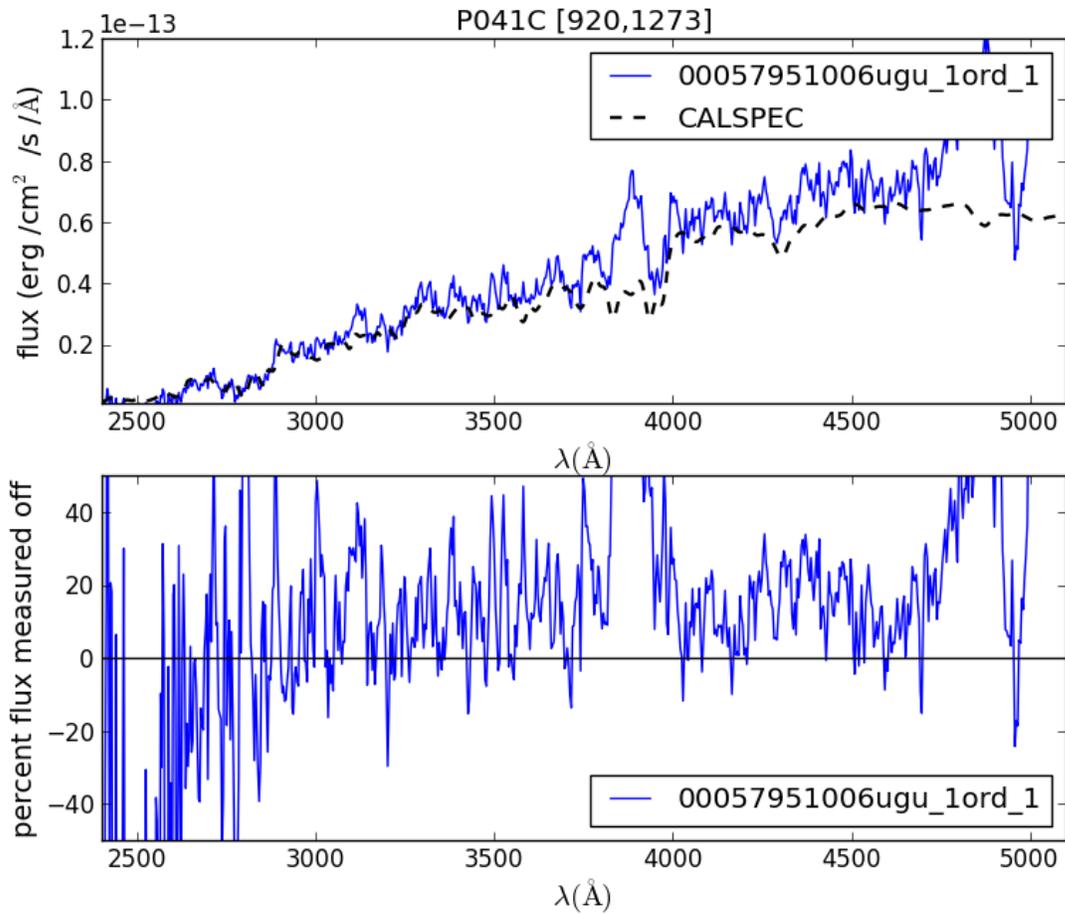


Figure 1.50: This spectrum with anchor at [920,1273].

GSPC-P041C Figure 9

GSPC-P041C Figure 10

GSPC-P177D spectra (F0 V)

GSPC-P177D Figure 5

WD1057+719 spectra (DA)

WD1057+719 Figure 5

WD1057+719 Figure 6

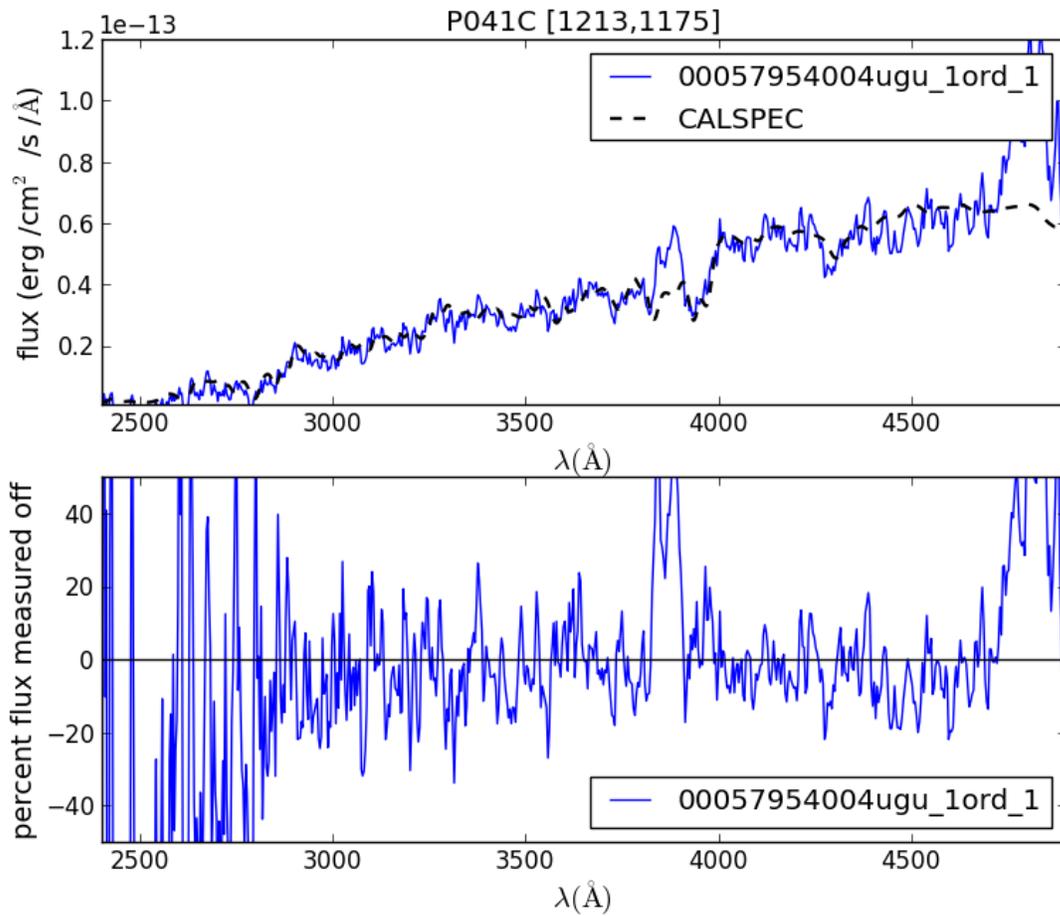


Figure 1.51: This spectrum with anchor at [1213,1175].

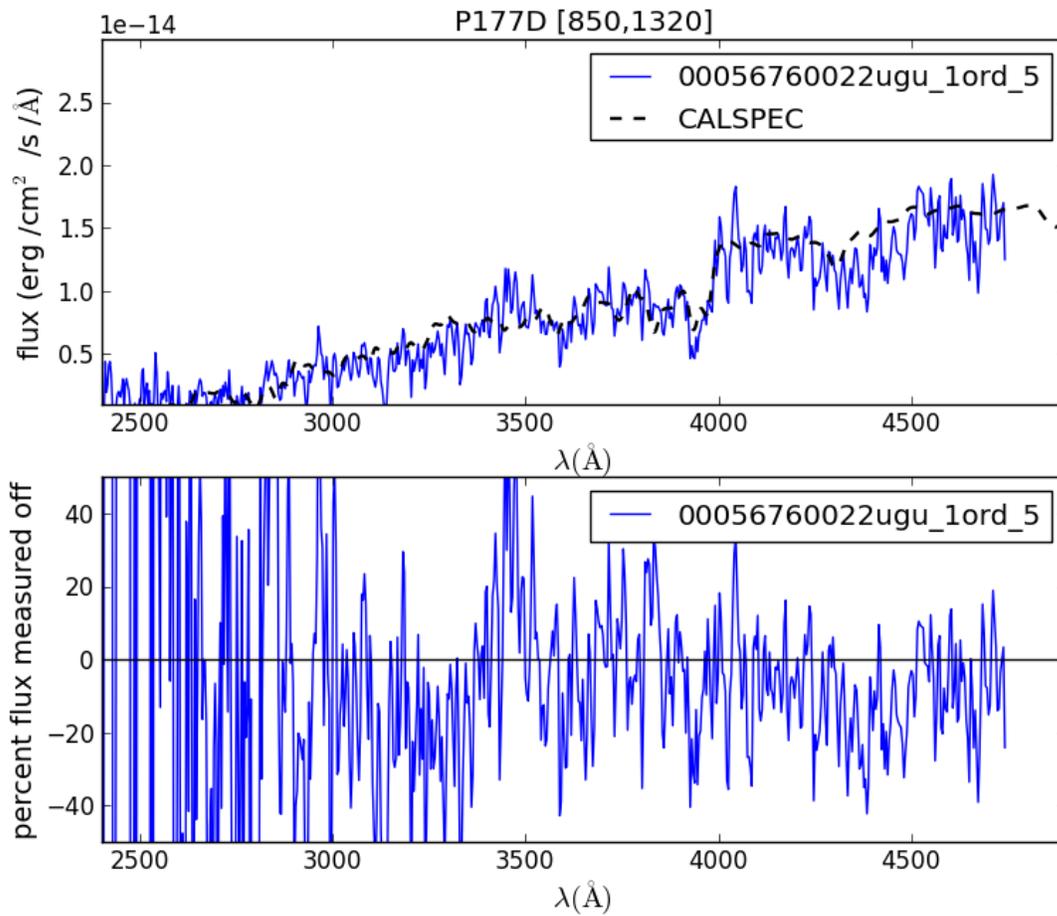


Figure 1.52: This spectrum with anchor at [850,1320].

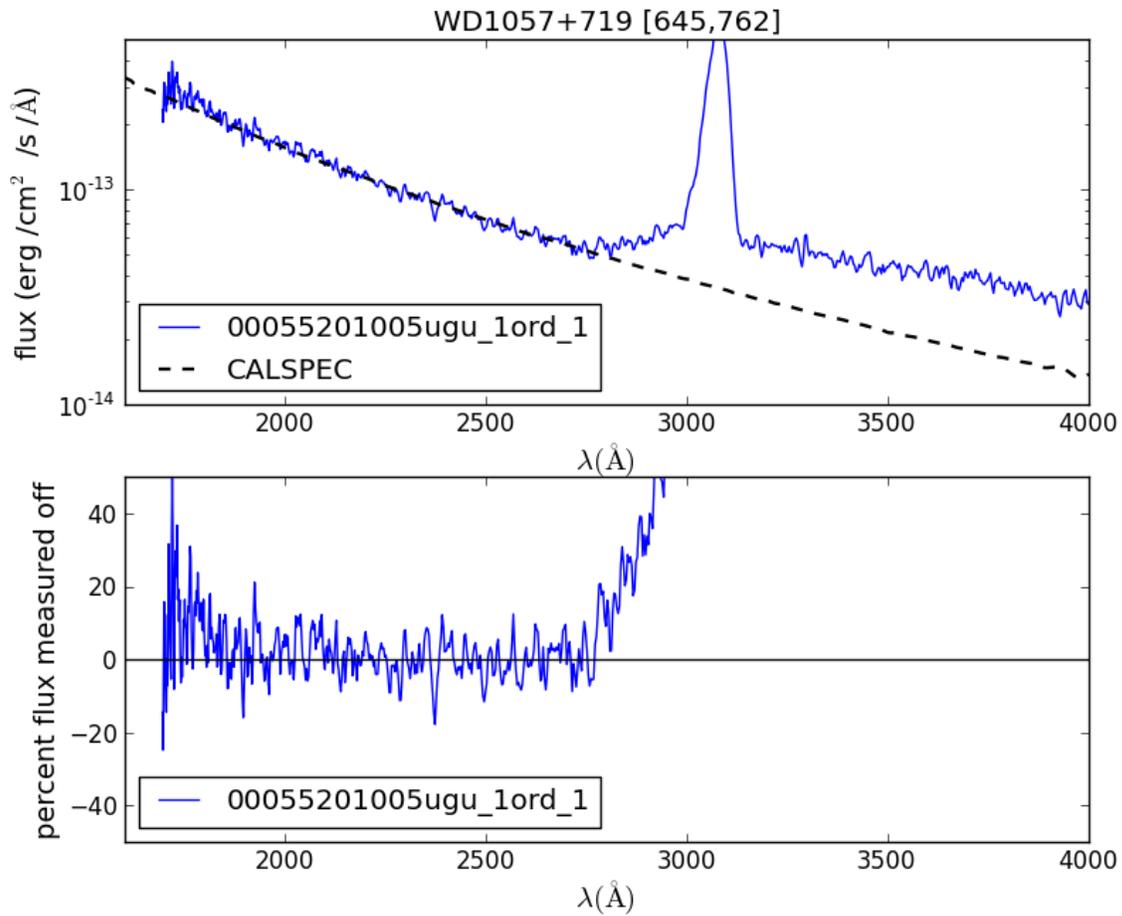


Figure 1.53: This spectrum with anchor at [645,762]. AI zeroth order is present contaminating the 3100Å region.

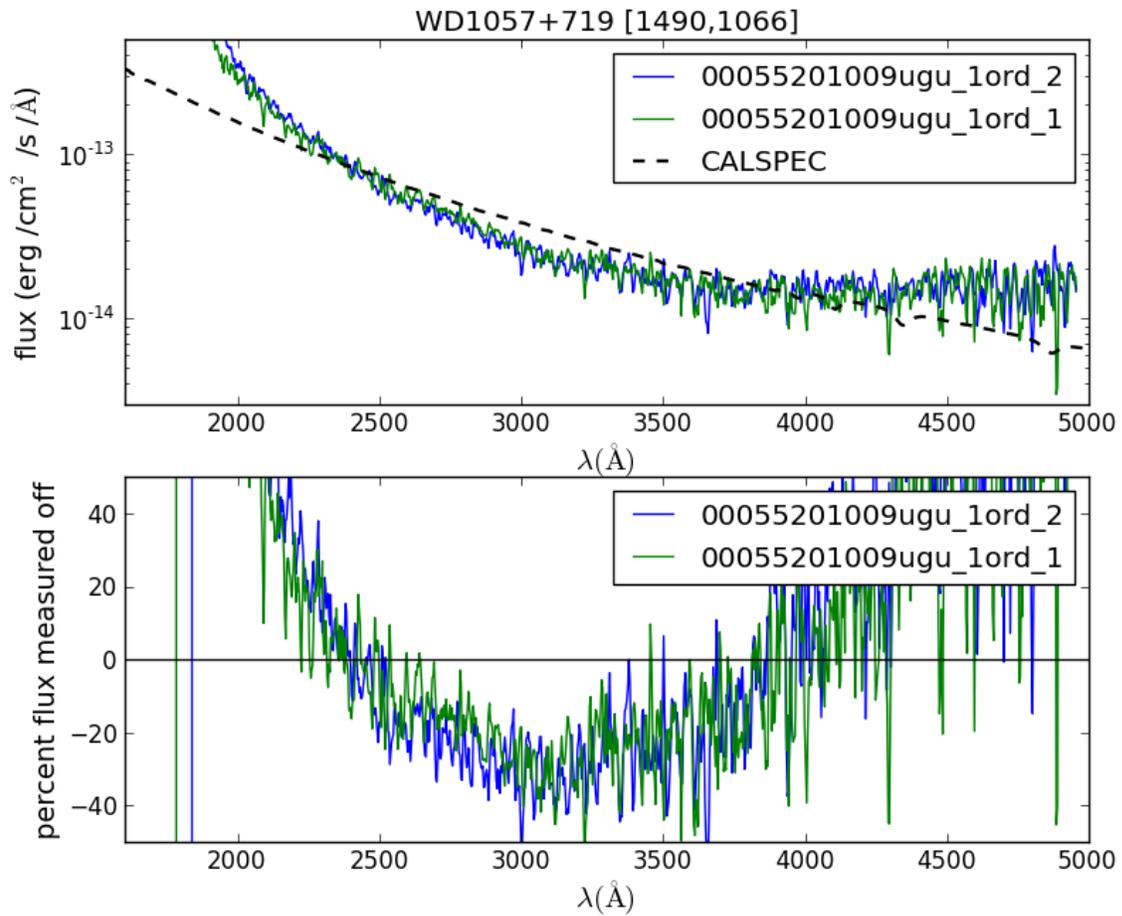


Figure 1.54: This spectrum with anchor at [1490,1066].

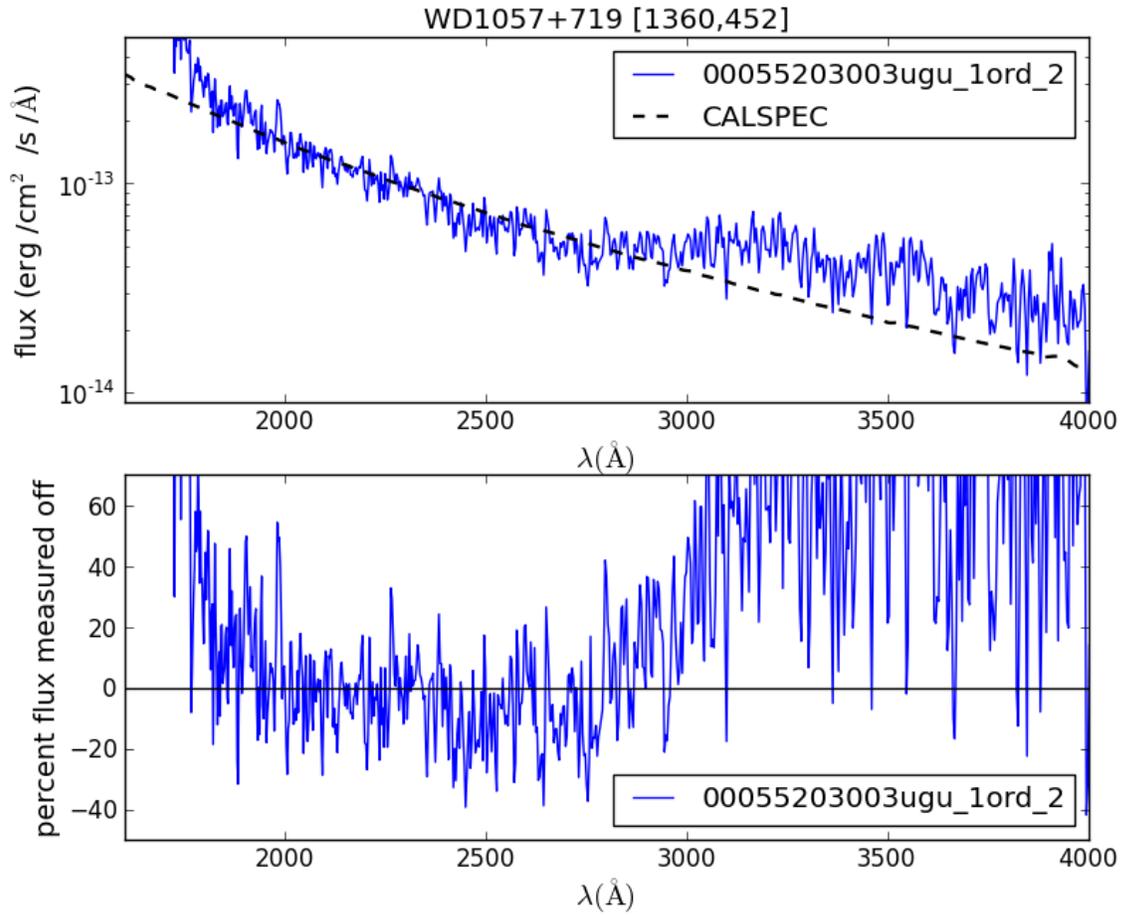


Figure 1.55: This spectrum with anchor at [1360,462].

WD1057+719 Figure 7

WD0320-539 spectra (DA)

WD0320-539 Figure 1

WD0320-539 Figure 2

WD0320-539 Figure 3

GD153 spectra (DA)

GD153 Figure 1

GD153 Figure 2

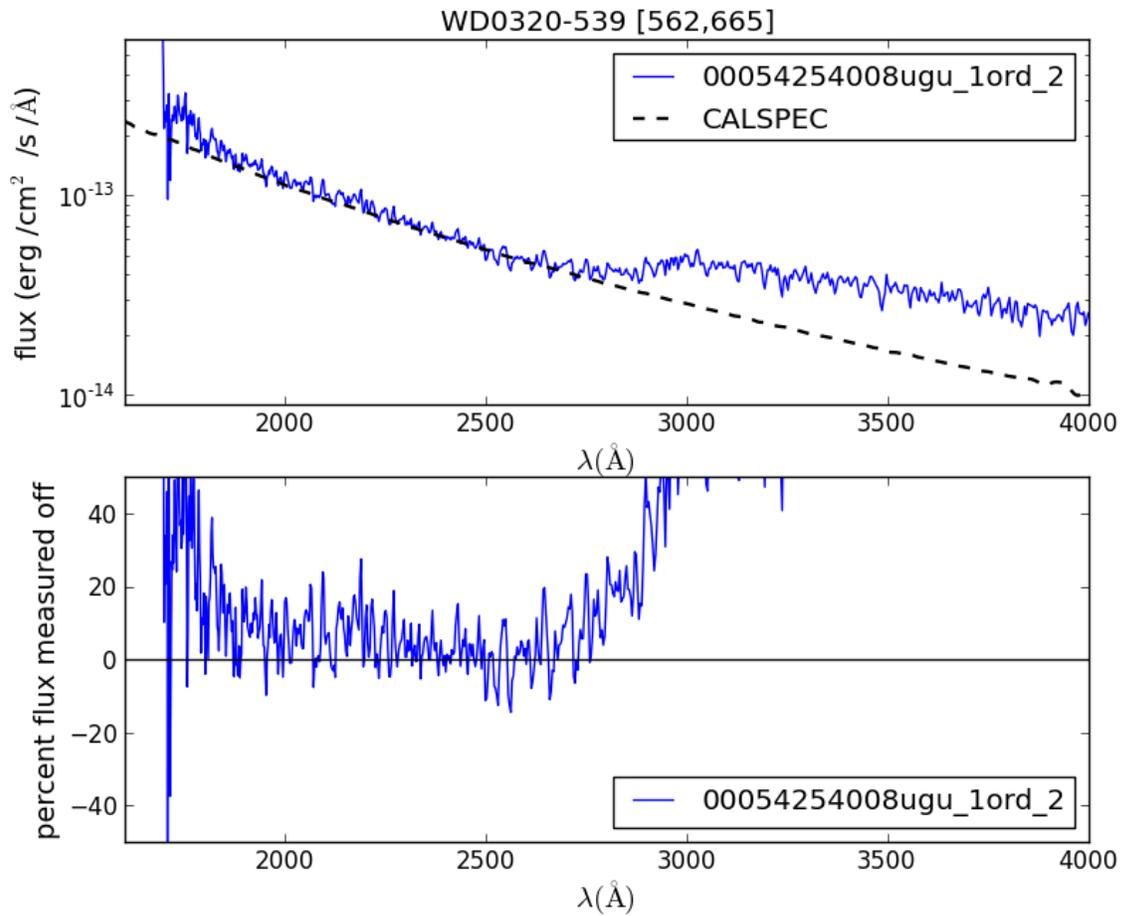


Figure 1.56: This spectrum with anchor at [562,665].

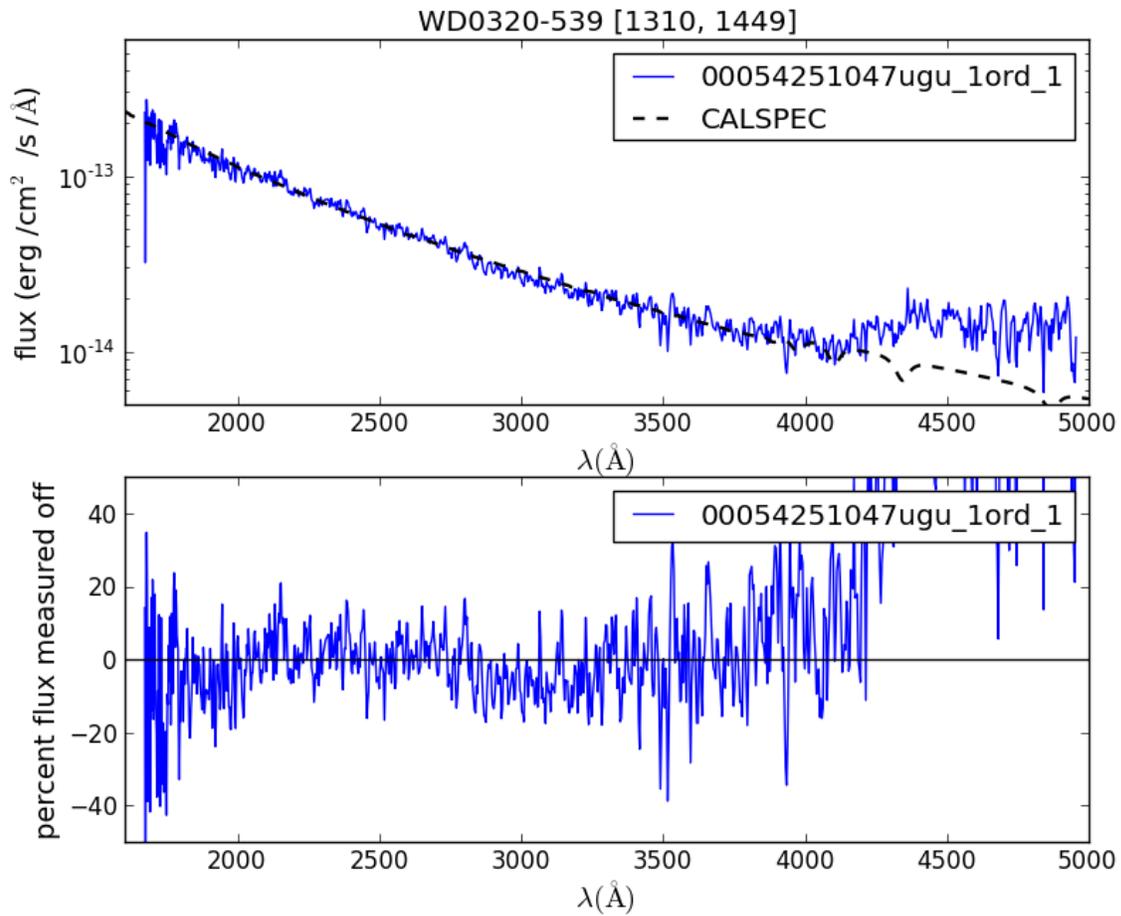


Figure 1.57: This spectrum with anchor at [1310,1449].

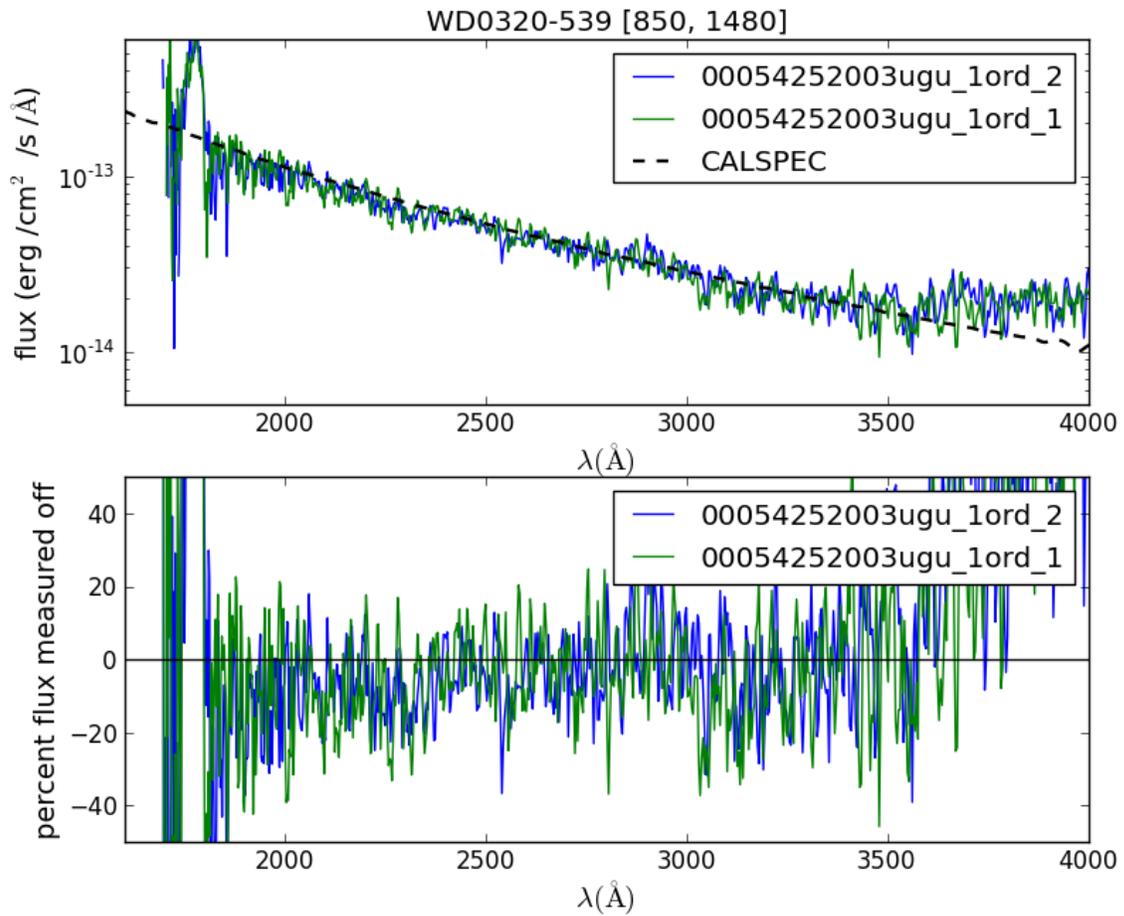


Figure 1.58: This spectrum with anchor at [850,1480].

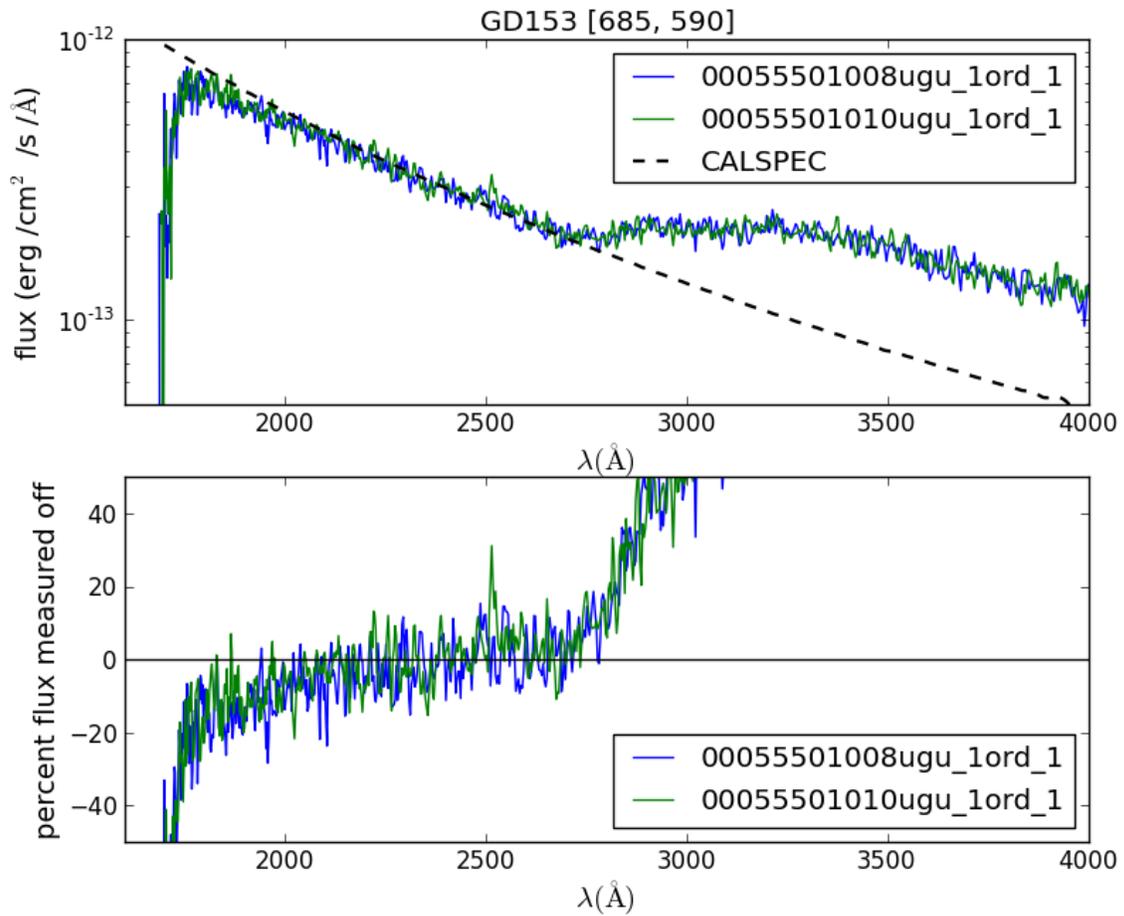


Figure 1.59: This spectrum with anchor at [685,590].

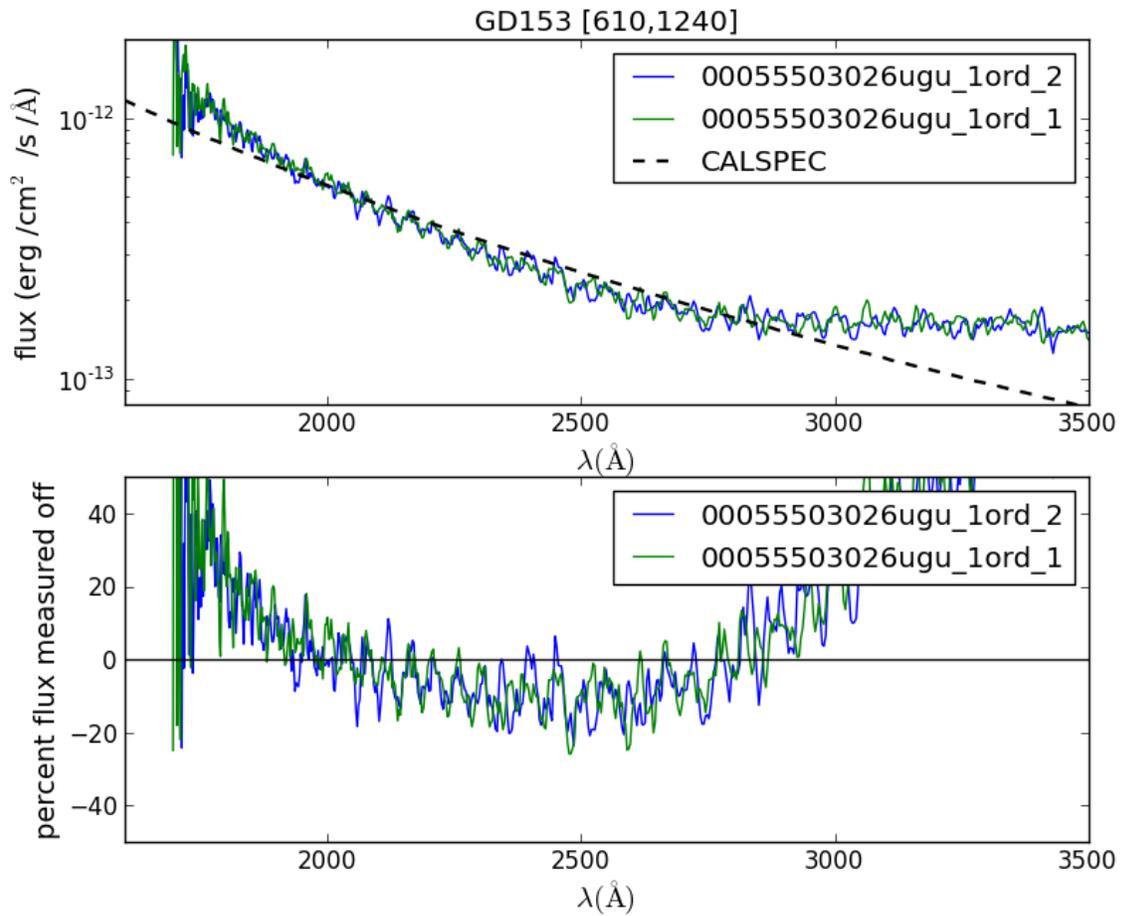


Figure 1.60: This spectrum with anchor at [610,1240].

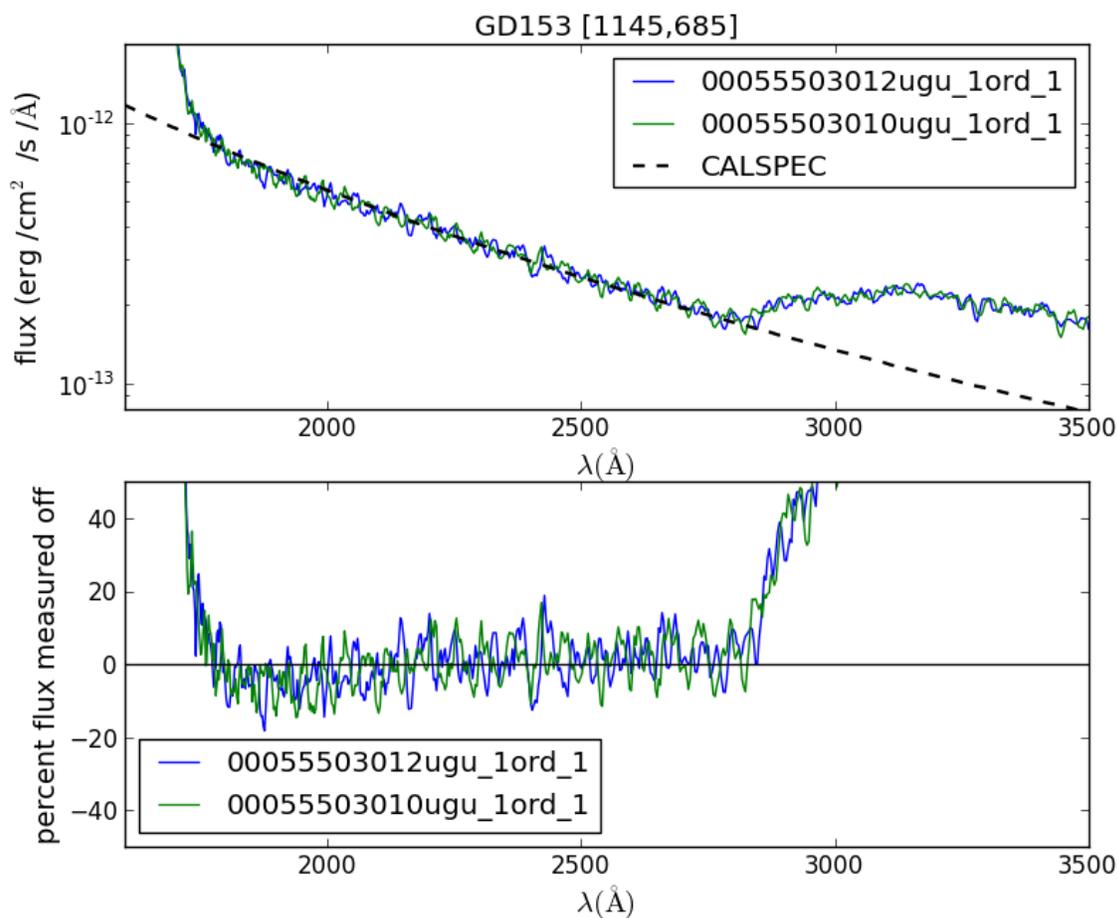


Figure 1.61: This spectrum with anchor at [1145,685].

GD153 Figure 3 Some of the spectra displayed have flux levels inconsistent with the known flux. In the 2014 calibration of the effective area we found that nearly all these observations had problems related to field source spectra overlying the target or background.

Verification of the flux calibration in the visual grism: clocked mode

Overview

The new v-grism flux calibration is valid over the whole detector, and makes also a correction for coincidence-loss. The first integrated software version is UVOTPY-0.9.7.3.

With the choice of data, we have an independent set from the spectra used to construct the effective area. Of course, spectra from the same targets are used.

Method

Calibration spectra that were not used in the flux calibration are reprocessed using the new flux calibration and then compared to the known stellar flux. The spectra for verification are located on the detector between the areas with the spectra used to determine the effective areas for the flux calibration.

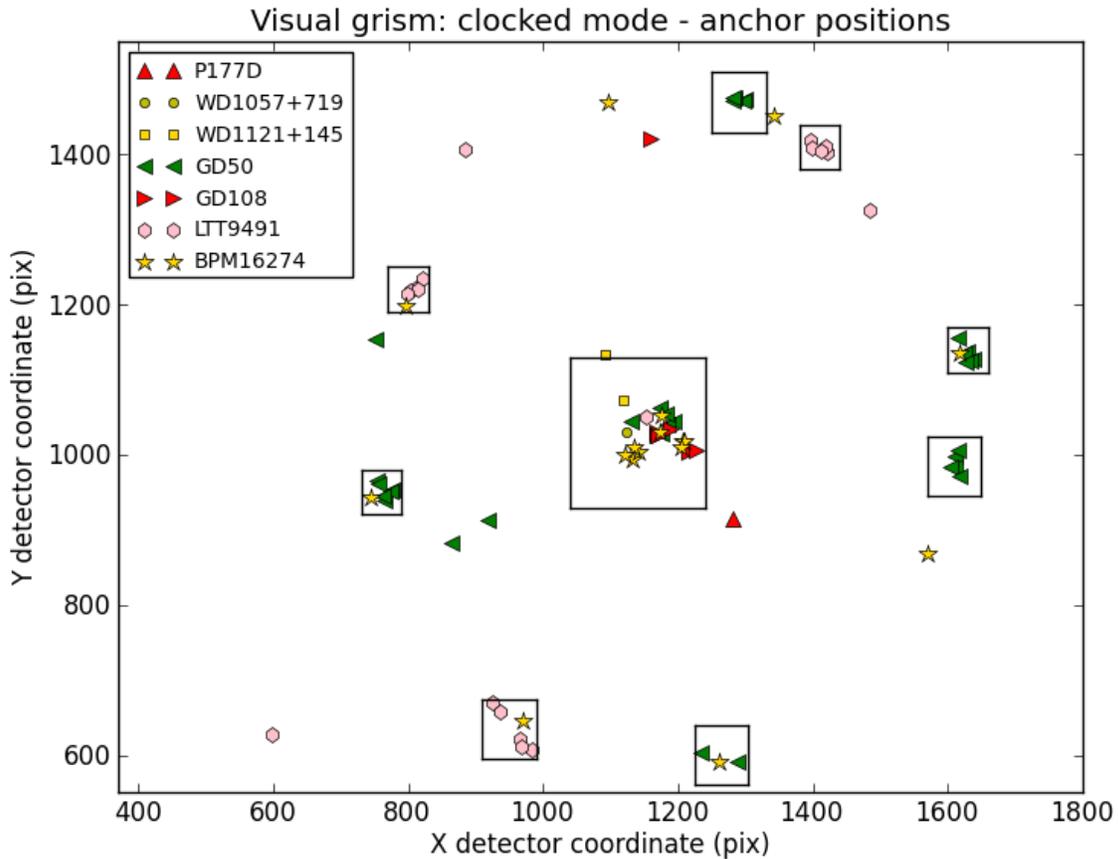


Figure 1.62: The anchor positions of the calibration spectra marked by target. The boxes are enclosing sources used to derive an effective area for the location of the box.

Data used for verification

The approximate location of the anchor of the spectra, their `obsid`, and number of the `fits` extension is listed. First the cool stars are listed, followed by the hot white dwarfs.

GSPC P177D spectra (F0 V)

anchor	obsid	ext	% err	plot	aspect correction
1281, 914	00056760001	1	10	<i>GSPC-P177D Figure 31</i>	yes

BPM16274 spectra (DA White Dwarf)

anchor	obsid	ext	% err	plot	aspect correction
1095,1470	00054003006	1	10	<i>BPM16274 Figure 1</i>	None
1570, 869	00054004004	1	15	<i>BPM16274 Figure 2</i>	None

LTT9491 spectra (DB3 White Dwarf)

anchor	obsid	ext	% err	plot	aspect correction
597, 628	00056202005	1	50	<i>LTT9491 Figure 1</i>	yes
885,1408	00056201003	1	25	<i>LTT9491 Figure 2</i>	yes
1485,1326	00056202003	1	20	<i>LTT9491 Figure 3</i>	yes

GD50 spectra (DA2 white Dwarf)

anchor	obsid	ext	% err	plot	aspect correction
562, 665	00054300019	1,2	18	<i>GD50 Figure 1</i>	None
750,1150	00054303003	1	15	<i>GD50 Figure 2</i>	yes

GD108 spectra (sdB)

anchor	obsid	ext	% err	plot	aspect correction
1150,1400	00055051008	1	10	<i>GD108 Figure 1</i>	yes

Result of the verification

The errors are typically 20% or less. The response below 2900Å shows some inconsistencies which is partly due to the low sensitivity, partly to the coincidence-loss correction. One spectrum of LTT9491 is located with anchor at [597,628], far from the calibrated region and the effective area has been extrapolated. A future update to the calibration based on extra coverage of the detector region where the response drops off steeply is needed to refine this. All other spectra are located in the calibrated region.

Figures for the visual grism : clocked mode

These figures go with the description *Verification of the flux calibration in the visual grism: clocked mode*, and show in the top panel the observed flux and the reference spectrum which is usually taken by HST.

In the bottom panel is plotted (observed - reference)/reference as a percentage. Notice that the bottom plot gets very noisy when the source flux is low. This is due to noise in the data. Also, at longer wavelengths second order contamination raises the observed flux, setting in at a wavelength that depends on the location of the blue rise of the spectrum.

GSPC-P177D spectra (F0 V)**GSPC-P177D Figure 31****BPM16274 spectra (DA)****BPM16274 Figure 1****BPM16274 Figure 2****LTT9491 spectra (DB3 White Dwarf)****LTT9491 Figure 1**

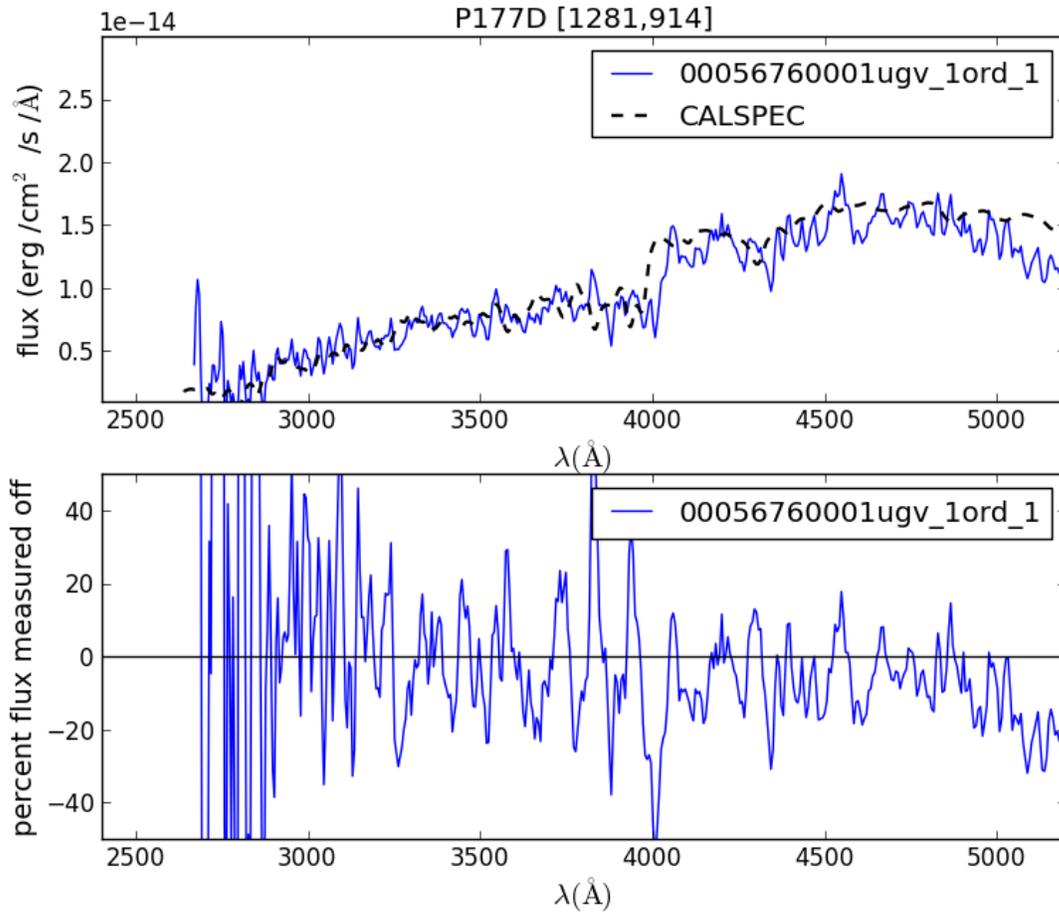


Figure 1.63: This spectrum with anchor at [1281,914].

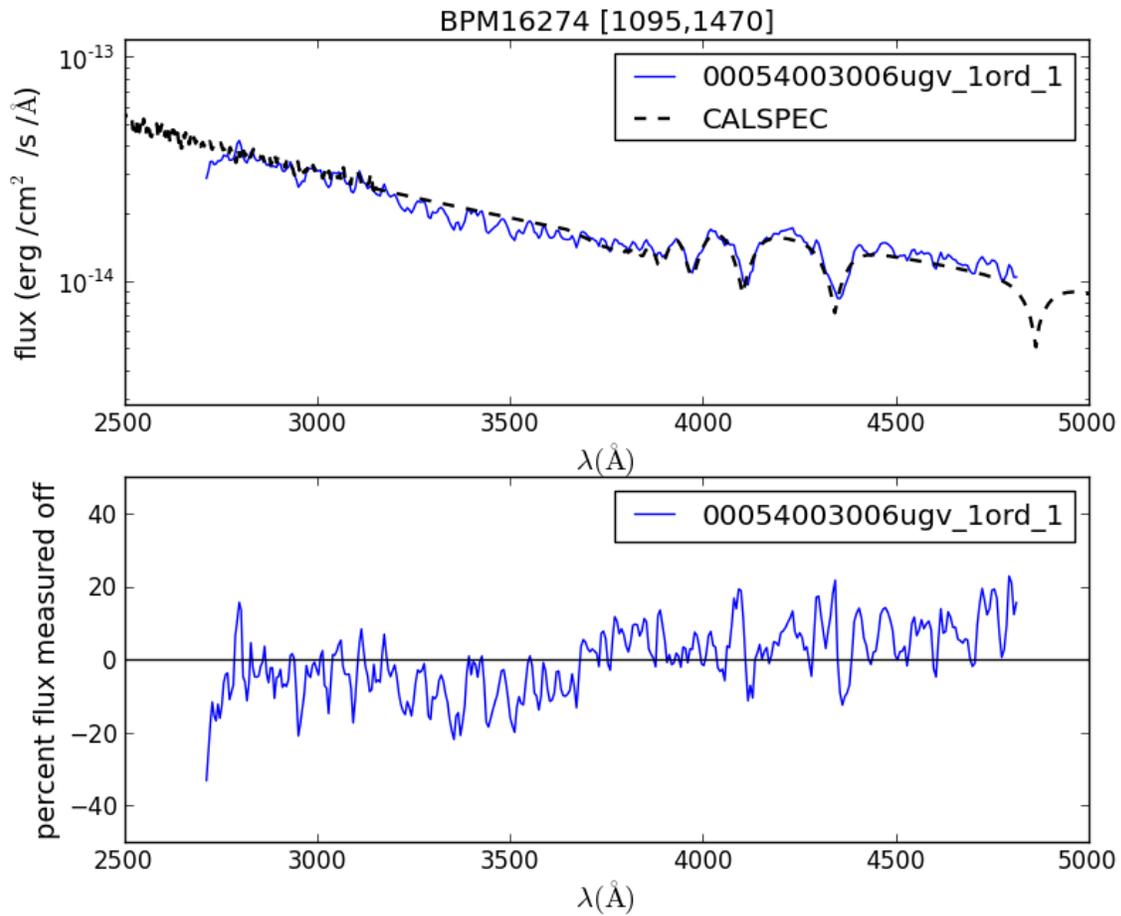


Figure 1.64: This spectrum with anchor at [1095,1470].

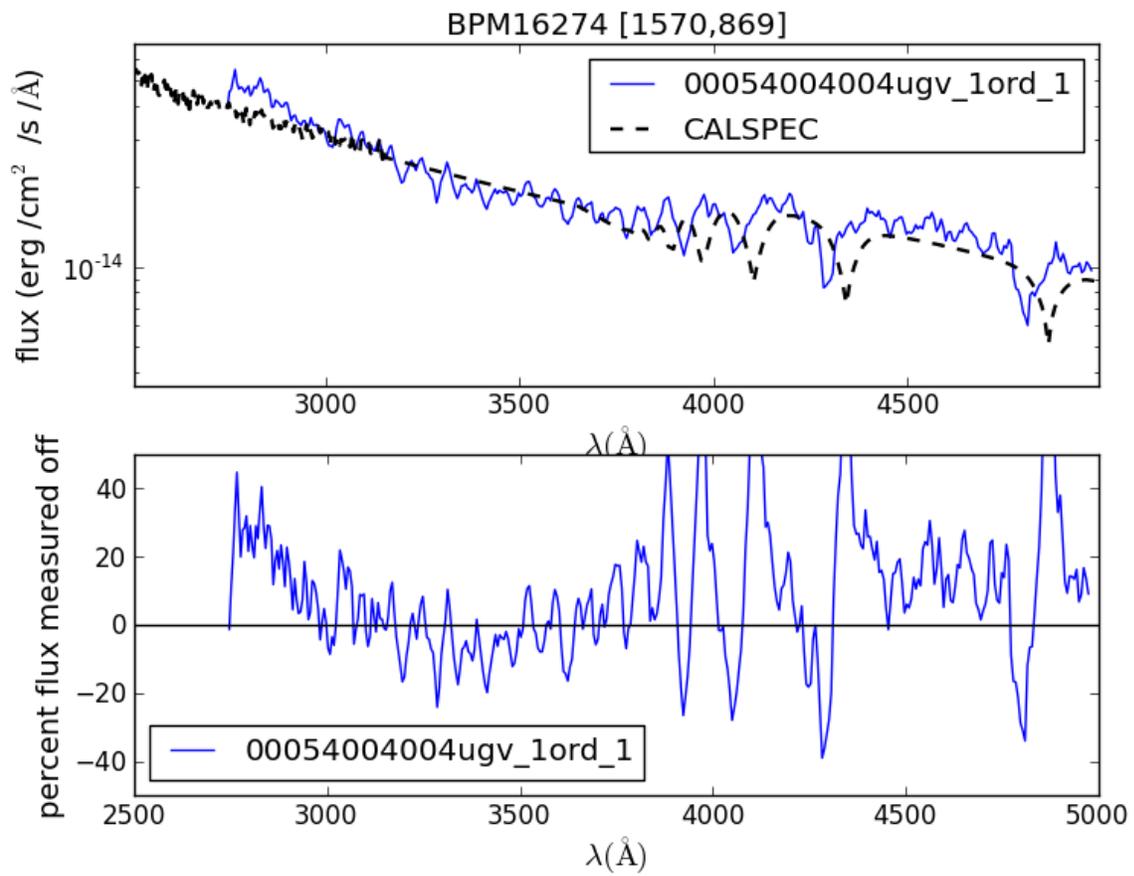


Figure 1.65: This spectrum with anchor at [1570, 869].

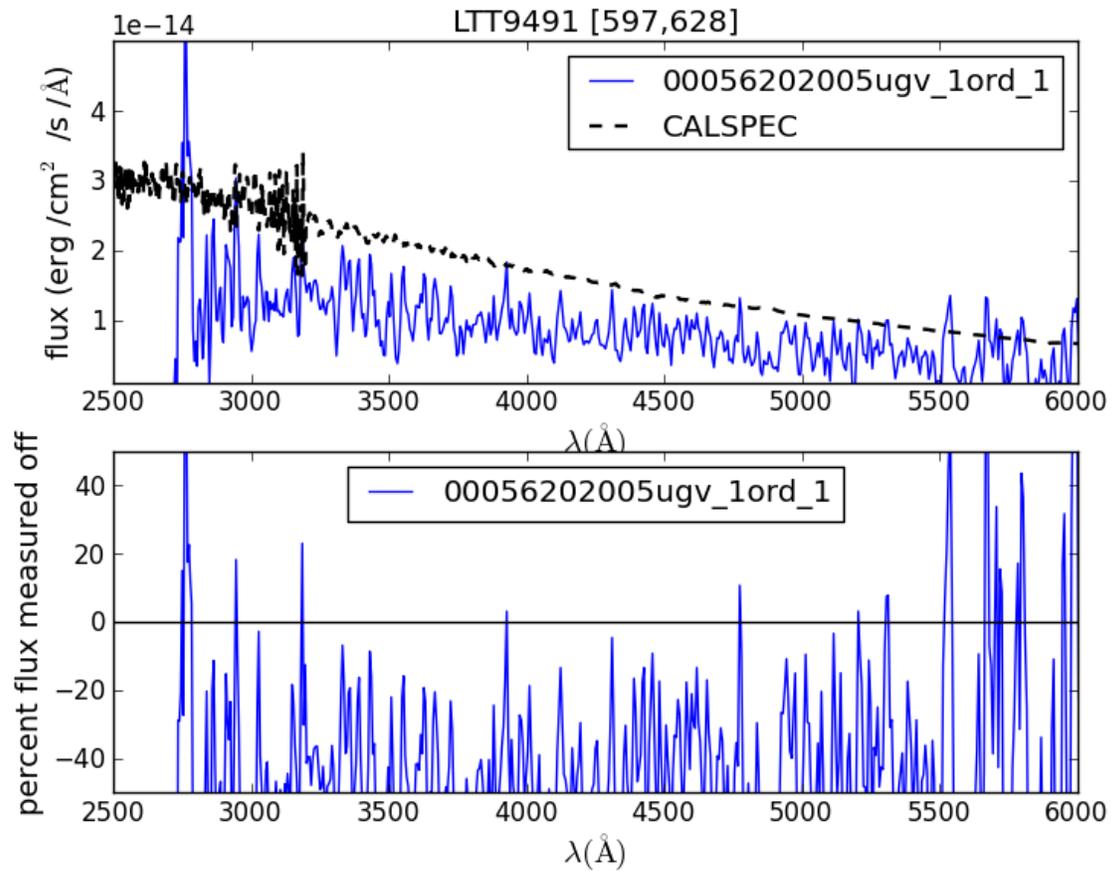


Figure 1.66: This spectrum with anchor at [597,628].!!! LOOKS LIKE ERROR IN EXTRACTION !!!

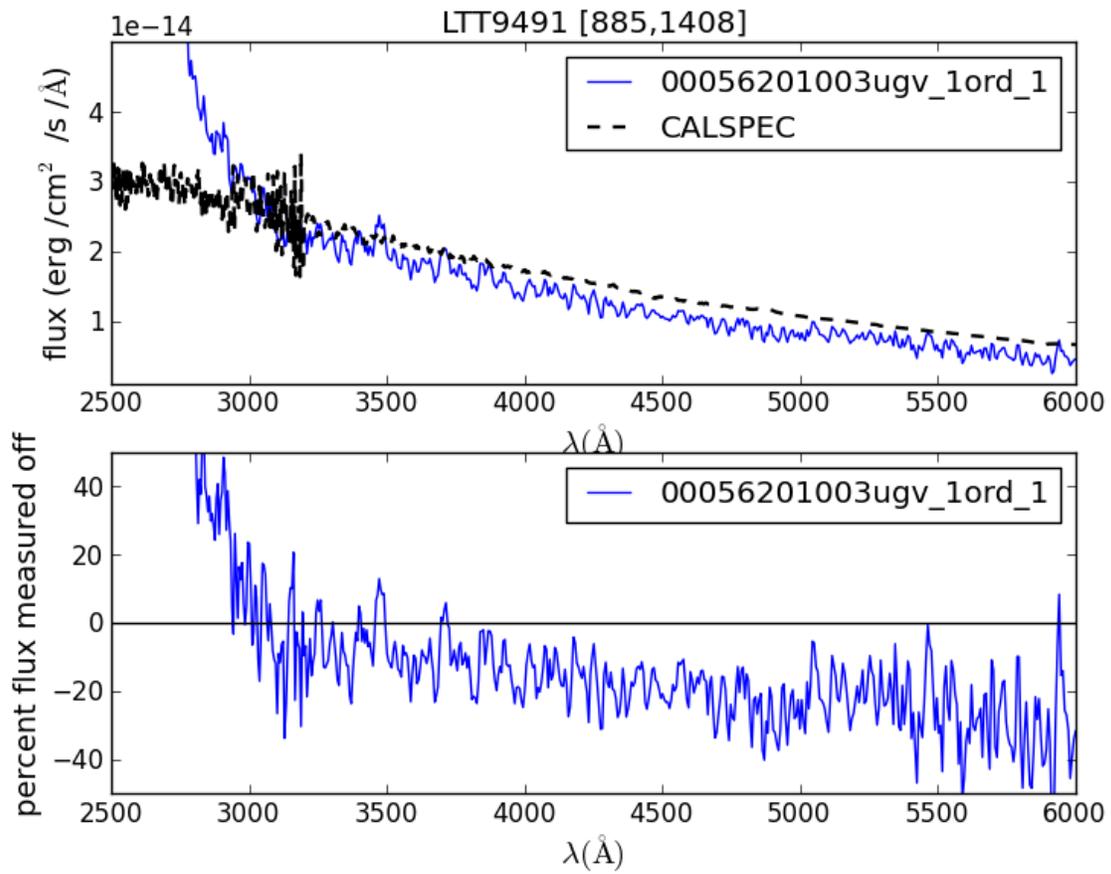


Figure 1.67: This spectrum with anchor at [885,1408].

LTT9491 Figure 2

LTT9491 Figure 3

GD50 spectra (DA)

GD50 Figure 1

GD50 Figure 2

GD108 spectra (sdB)

GD108 Figure 1

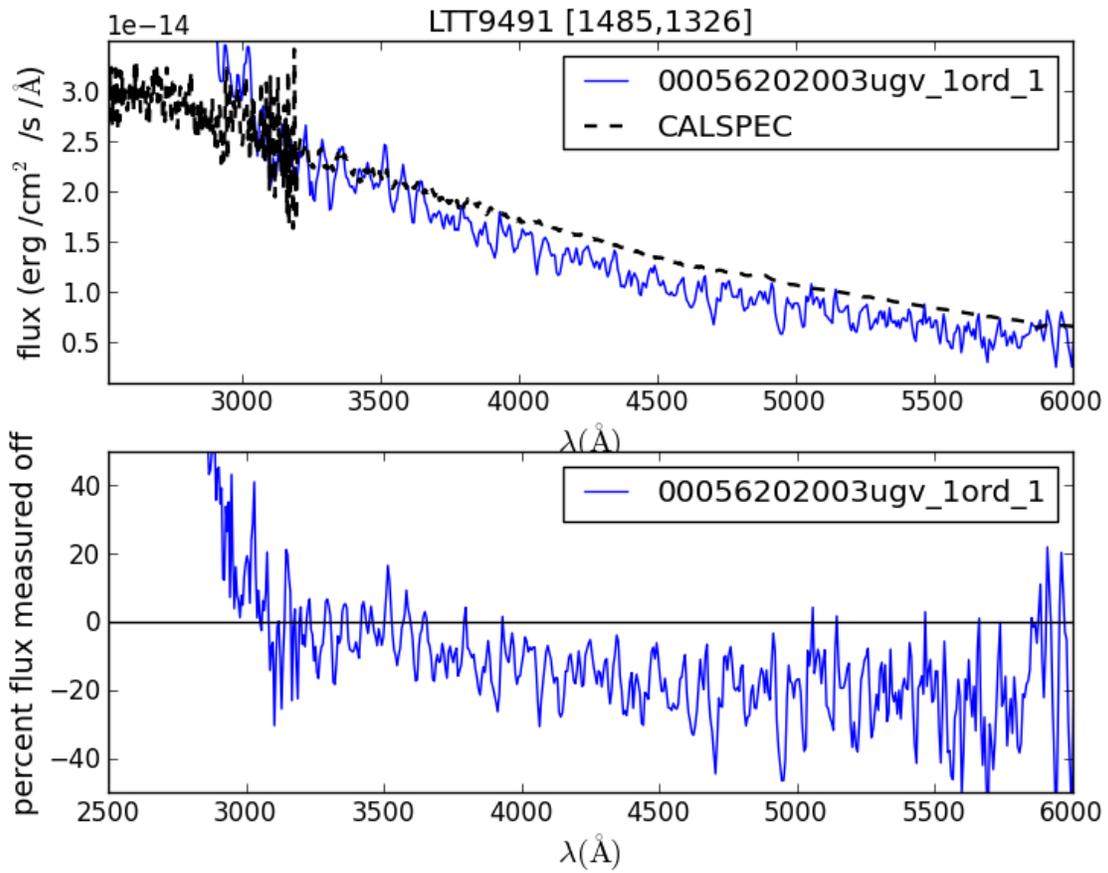


Figure 1.68: This spectrum with anchor at [1485,1326].

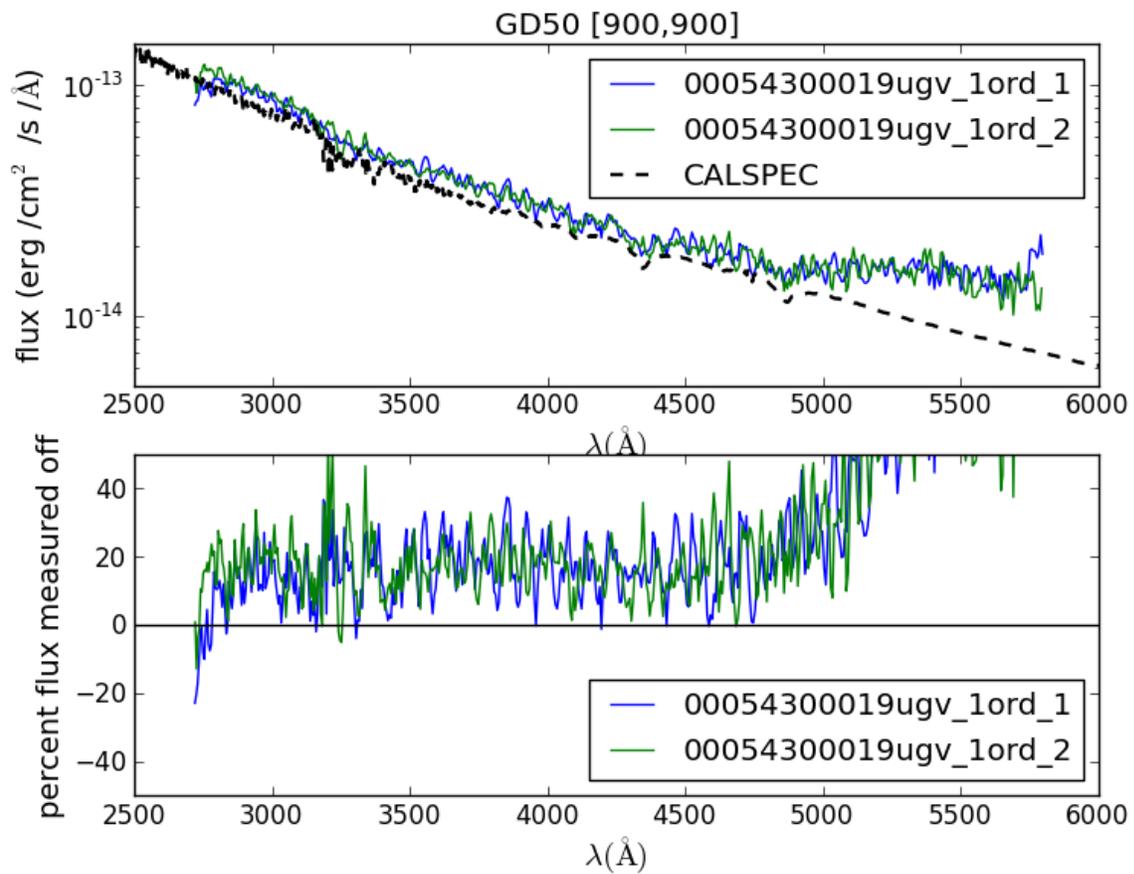


Figure 1.69: This spectrum with anchor at [900,900].

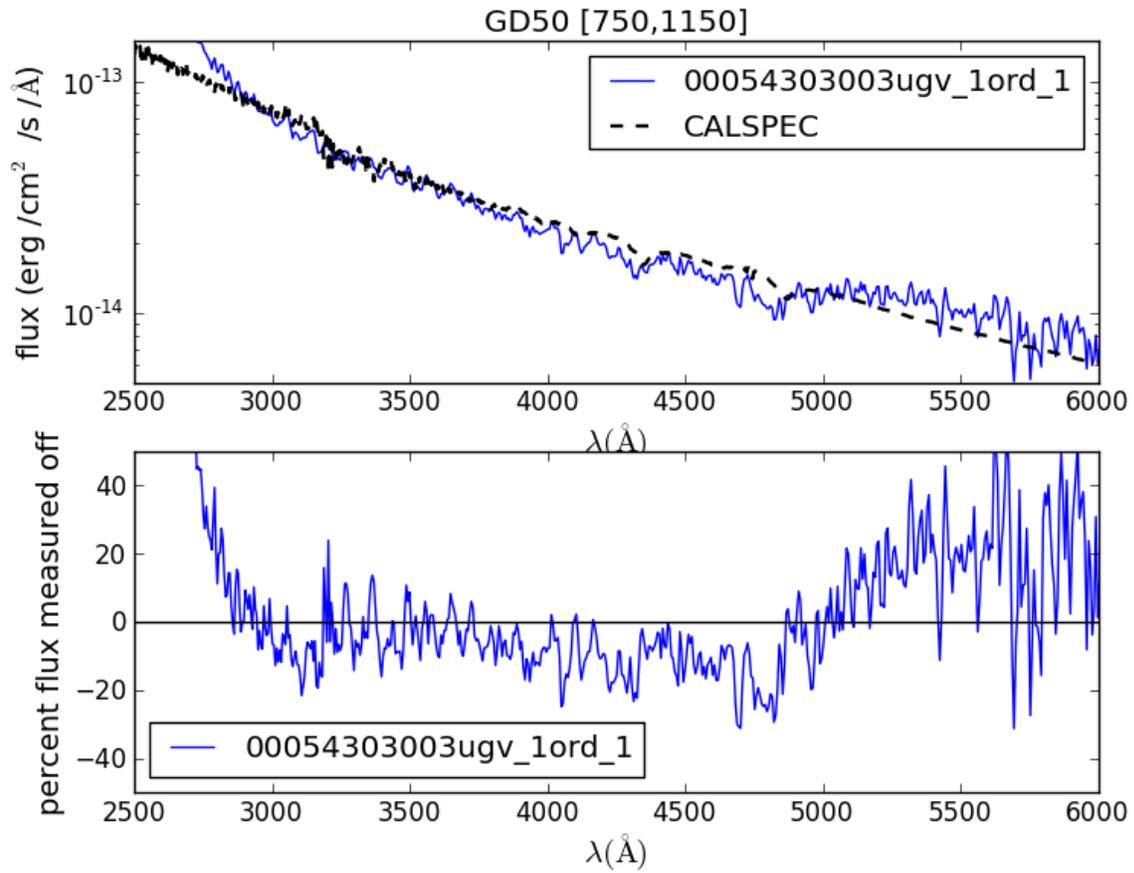


Figure 1.70: This spectrum with anchor at [750,1150].

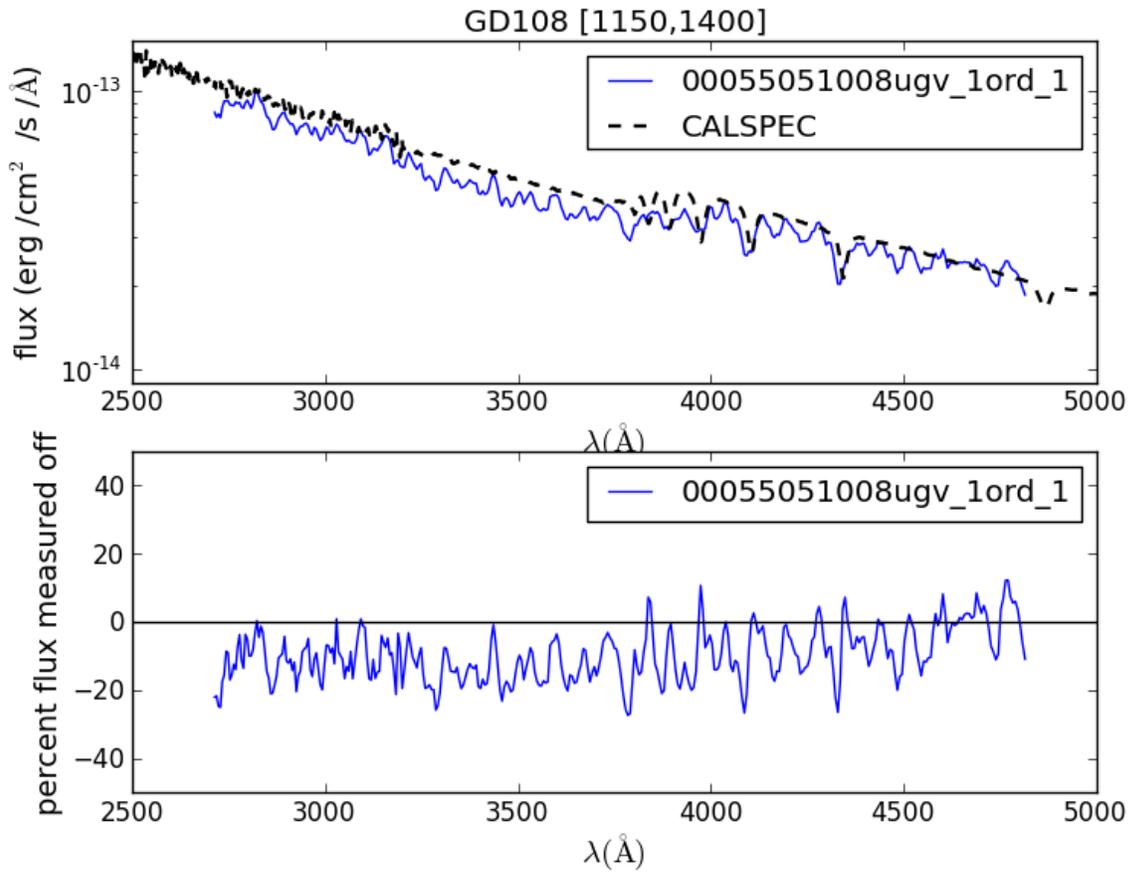


Figure 1.71: This spectrum with anchor at [1150,1400].

Verification of the flux calibration in the visual grism: nominal mode

Overview

The new v-grism flux calibration is valid over the whole detector, and makes also a correction for coincidence-loss. The first integrated software version is UVOTPY-0.9.7.3.

With the choice of data, we have an independent set from the spectra used to construct the effective area. Of course, spectra from the same targets are used.

Method

Calibration spectra that were not used in the flux calibration are reprocessed using the new flux calibration and then compared to the known stellar flux. The spectra for verification are located on the detector between the areas with the spectra used to determine the effective areas for the flux calibration.

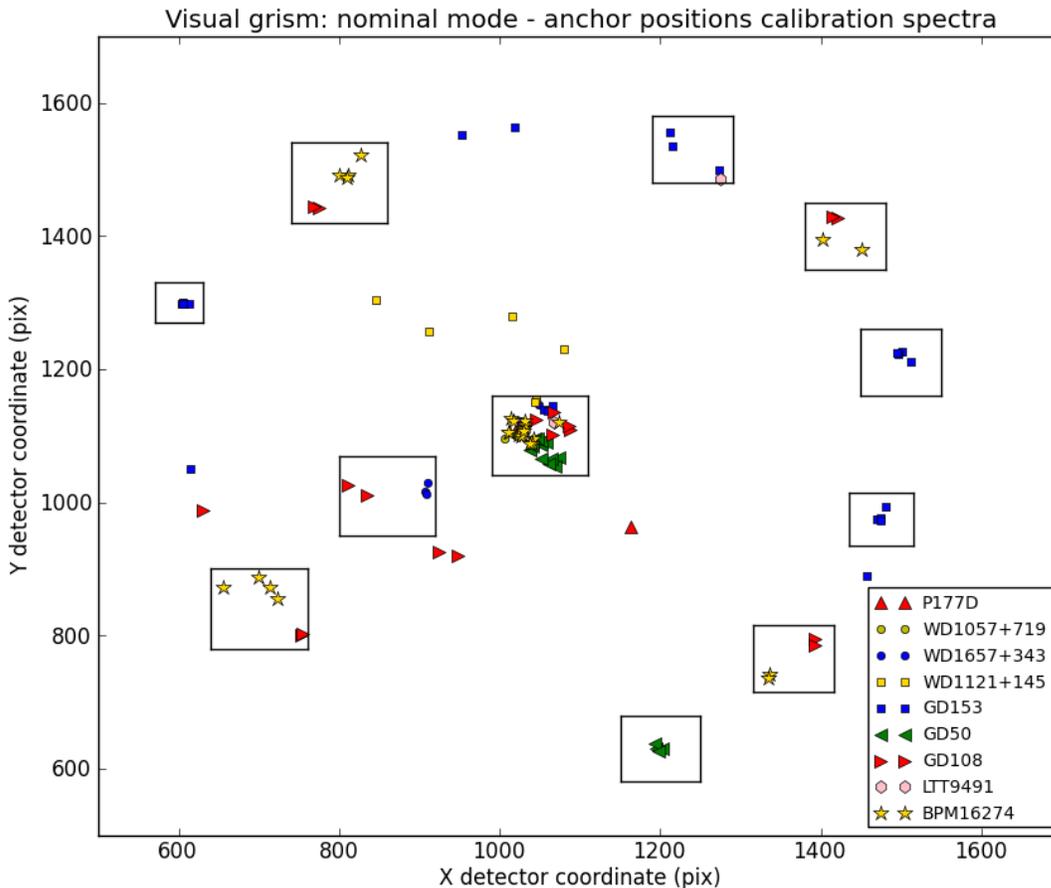


Figure 1.72: The anchor positions of the calibration spectra marked by target. The boxes are enclosing sources used to derive an effective area for the location of the box.

Data used for verification

The approximate location of the anchor of the spectra, their `obsid`, and number of the `fits` extension is listed. First the cool stars are listed, followed by the hot white dwarfs.

WD1121+145 spectra

anchor	obsid	ext	% err	plot	aspect correction
845,1304	00055250072	2	20	<i>WD1121+145 Figure 1</i>	None
912,1256	00055250072	3	20	<i>WD1121+145 Figure 2</i>	None
1015,1280	00055250072	4	15	<i>WD1121+145 Figure 3</i>	None
1080,1230	00055250072	1	15	<i>WD1121+145 Figure 4</i>	None

GD108 spectra

anchor	obsid	ext	% err	plot	aspect correction
631, 989	00055050016	1	20	<i>GD108 Figure 1</i>	yes
924, 924	00055051003	1	20	<i>GD108 Figure 2</i>	yes
949, 920	00055050012	1	20	<i>GD108 Figure 3</i>	yes

GD153 spectra

anchor	obsid	ext	% err	plot	aspect correction
615,1050	00055501004	1	15	<i>GD153 Figure 1</i>	yes
952,1550	00055502008	1	10	<i>GD153 Figure 2</i>	yes
1019,1563	00055502004	1	20	<i>GD153 Figure 3</i>	yes
1456, 889	00055503020	1	10	<i>GD153 Figure 4</i>	yes

Result of the verification

Errors were read off the figures as being typical. The errors are typically 20% or less. For a large part these are due to inaccuracy in the coincidence loss correction and the Poisson error in the counts.

Figures for the visual grism : nominal

These figures go with the description *Verification of the flux calibration in the visual grism: nominal mode*, and show in the top panel the observed flux and the reference spectrum which is usually taken by HST.

In the bottom panel is plotted $(\text{observed} - \text{reference})/\text{reference}$ as a percentage. Notice that the bottom plot gets very noisy when the source flux is low. This is due to noise in the data. Also, at longer wavelengths second order contamination raises the observed flux, setting in at a wavelength that depends on the location of the blue rise of the spectrum.

WD1121+145 spectra (DA)

WD1121+145 Figure 1

WD1121+145 Figure 2

WD1121+145 Figure 3

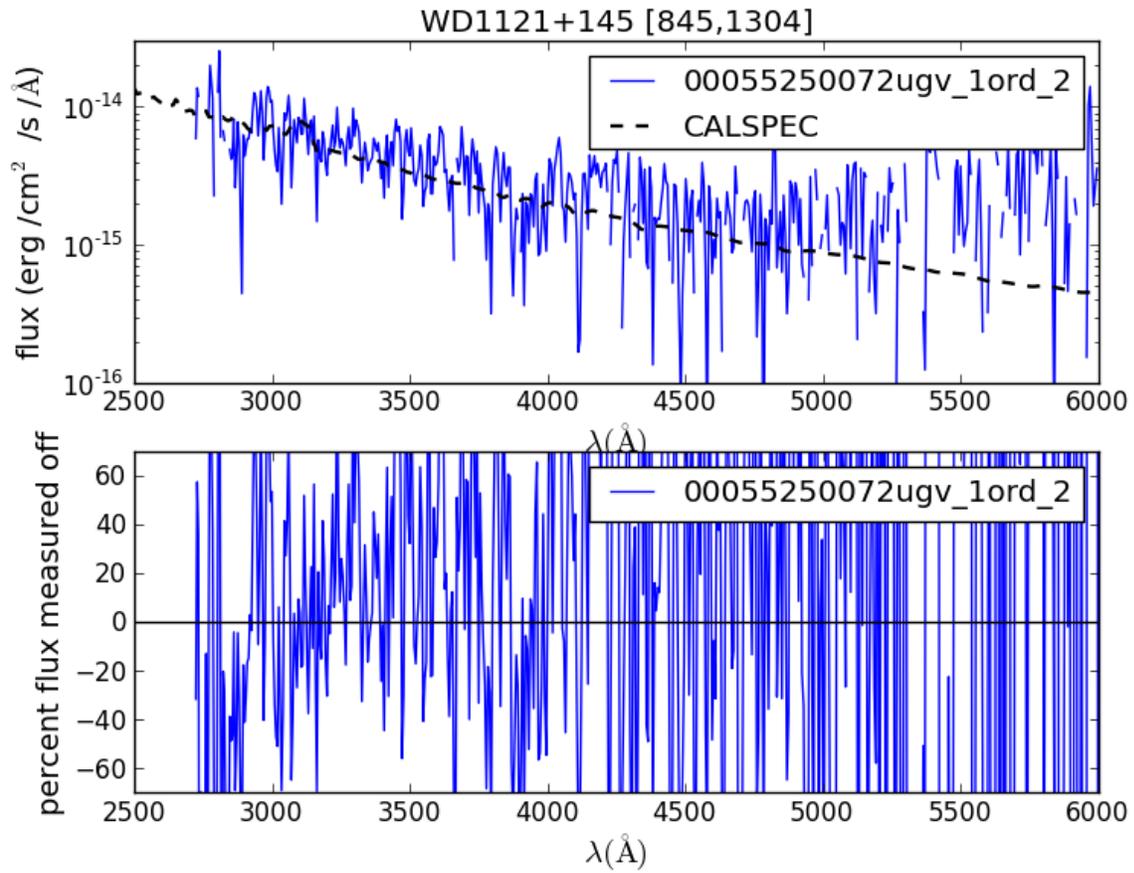


Figure 1.73: This spectrum with anchor at [845,1304].

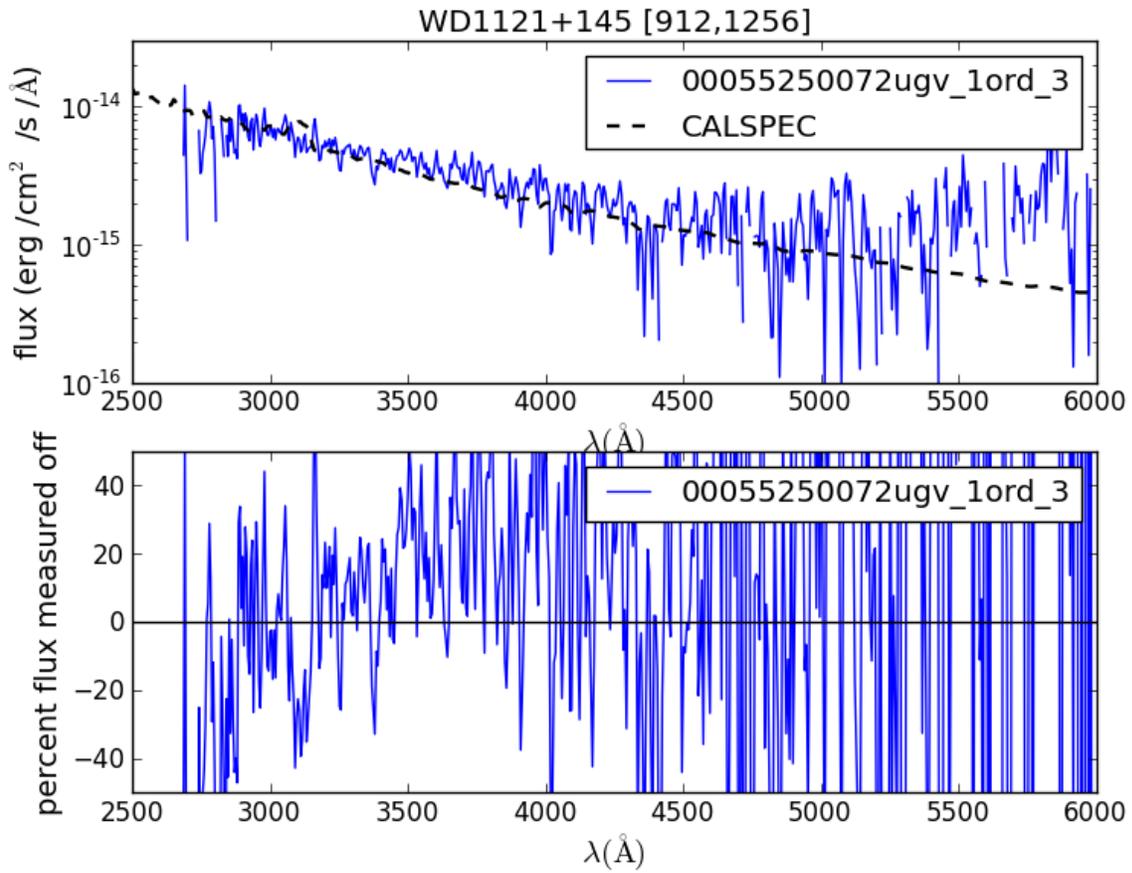


Figure 1.74: This spectrum with anchor at [912,1256].

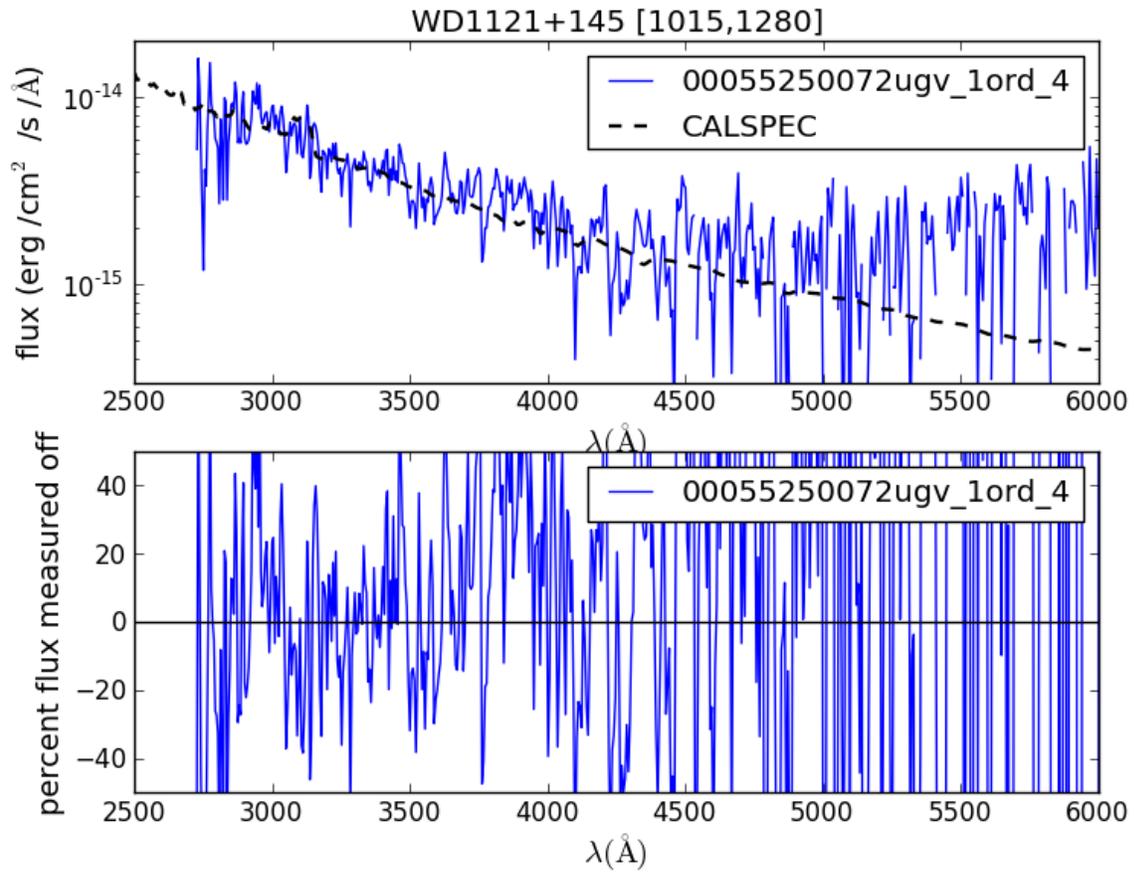


Figure 1.75: This spectrum with anchor at [1015,1280].

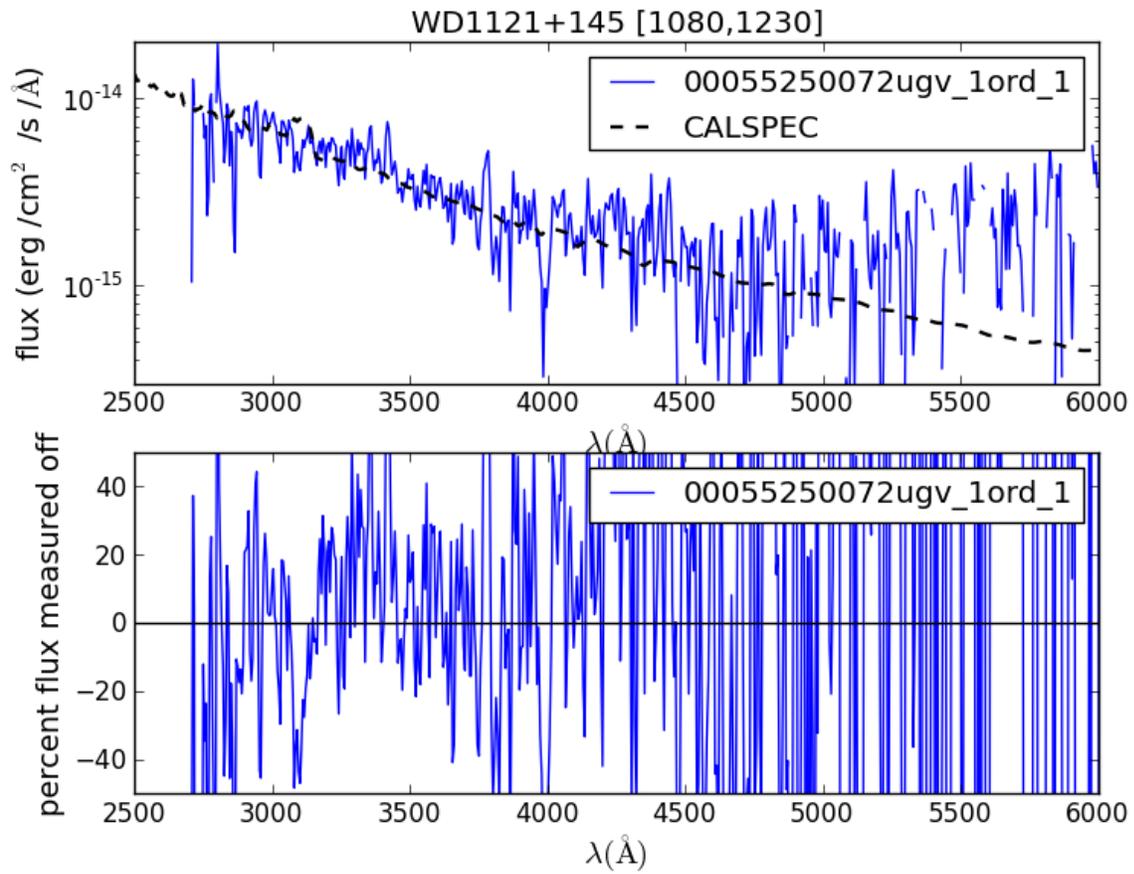


Figure 1.76: This spectrum with anchor at [1080,1230].

WD1121+145 Figure 4

GD108 spectra (sdB)

GD108 Figure 1

GD108 Figure 2

GD108 Figure 3

GD153 spectra (DA)

GD153 Figure 1

GD153 Figure 2

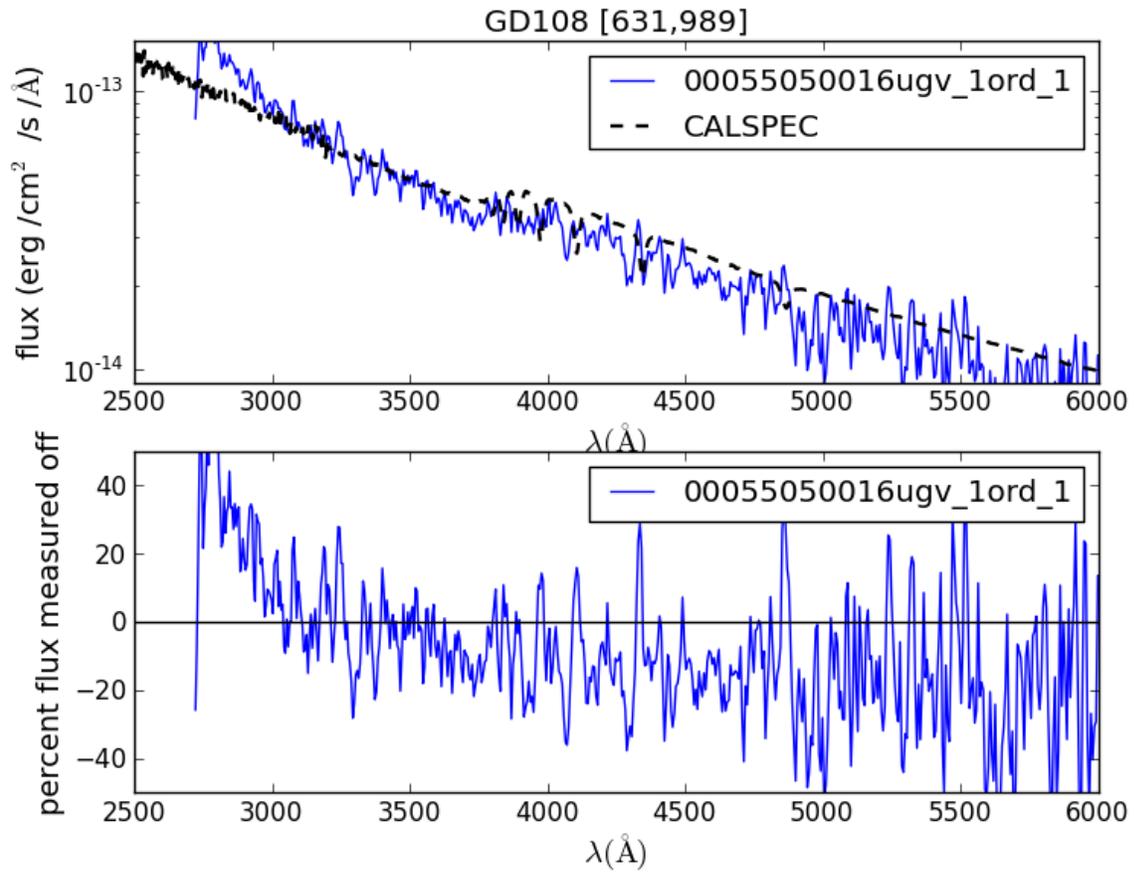


Figure 1.77: This spectrum with anchor at [631,989].

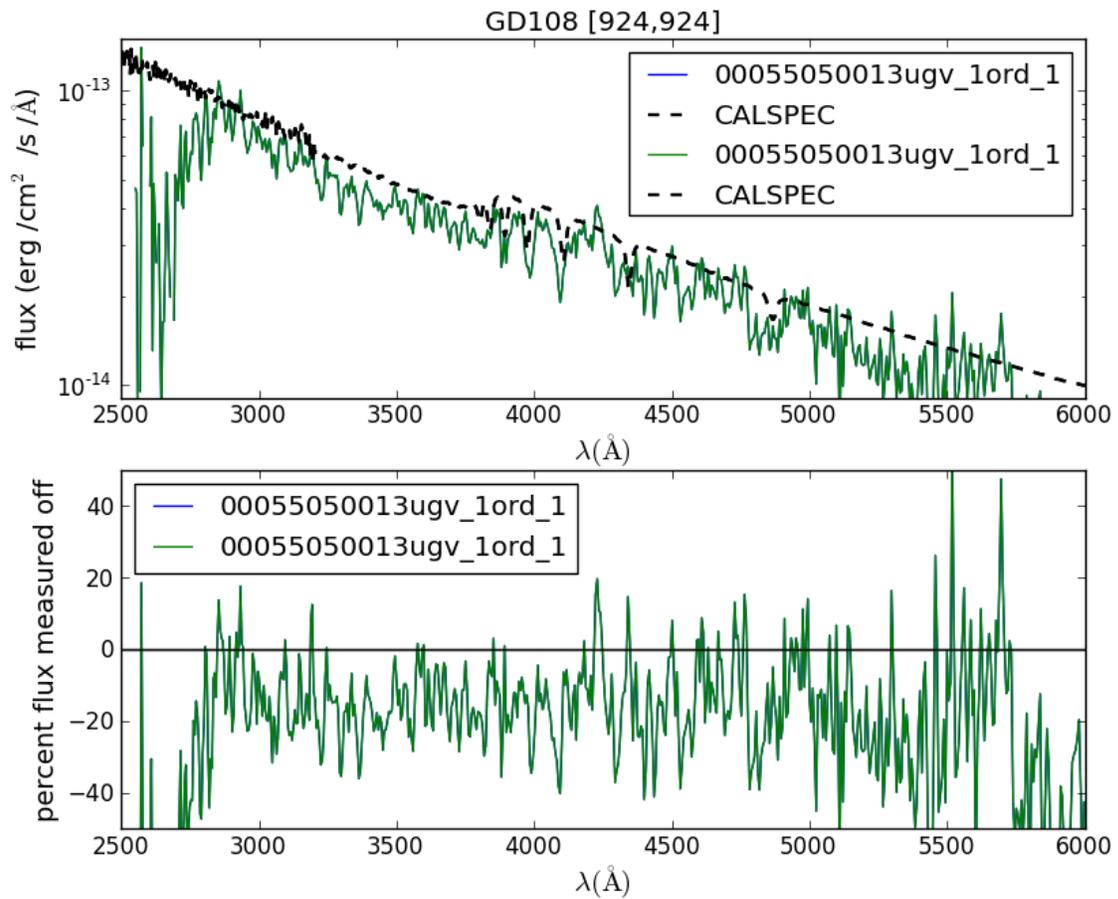


Figure 1.78: This spectrum with anchor at [924,924].

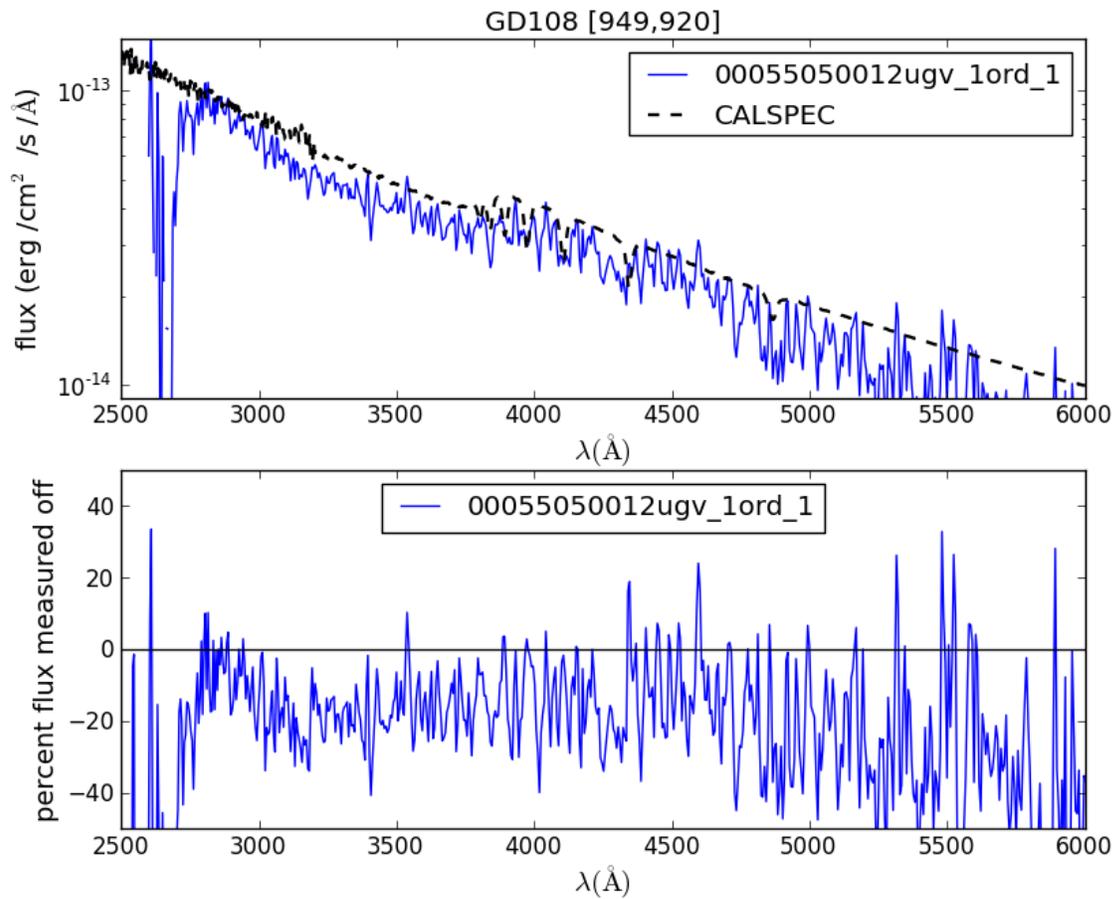


Figure 1.79: This spectrum with anchor at [949,920].

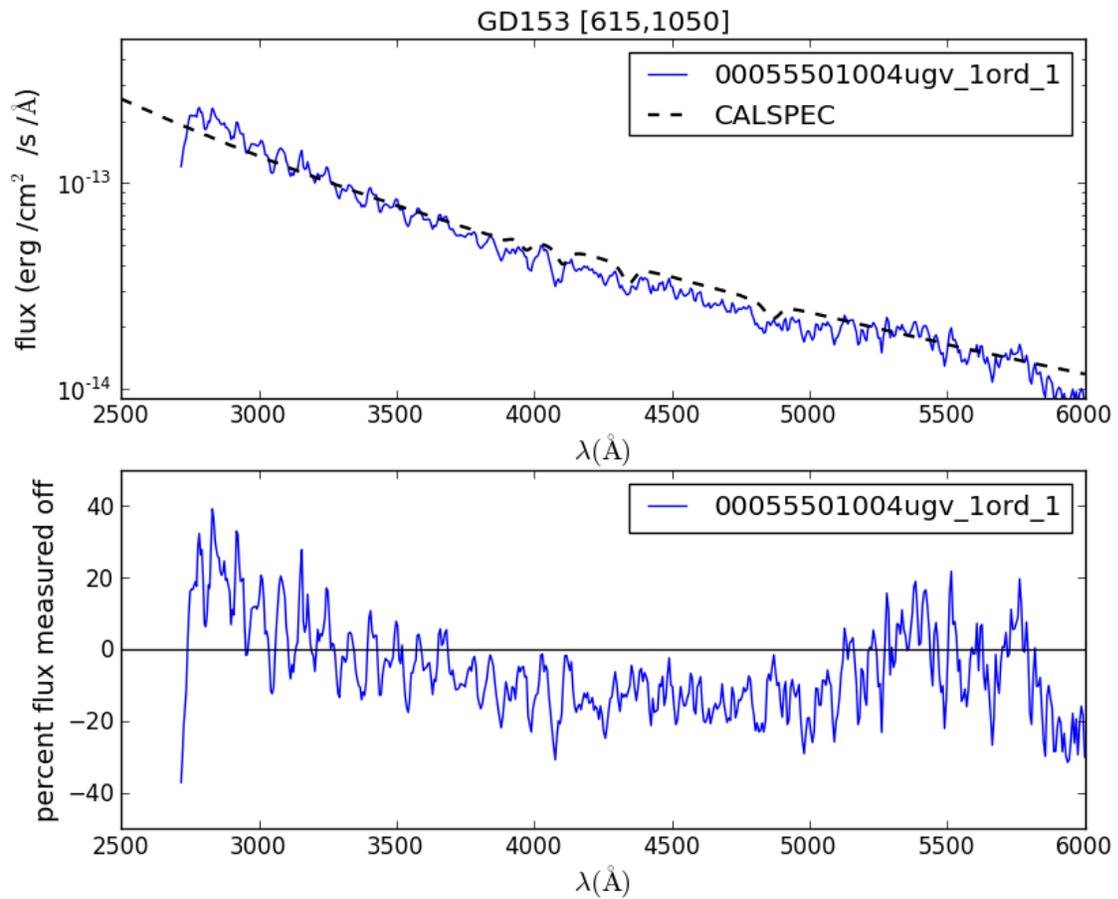


Figure 1.80: This spectrum with anchor at [615,1050].

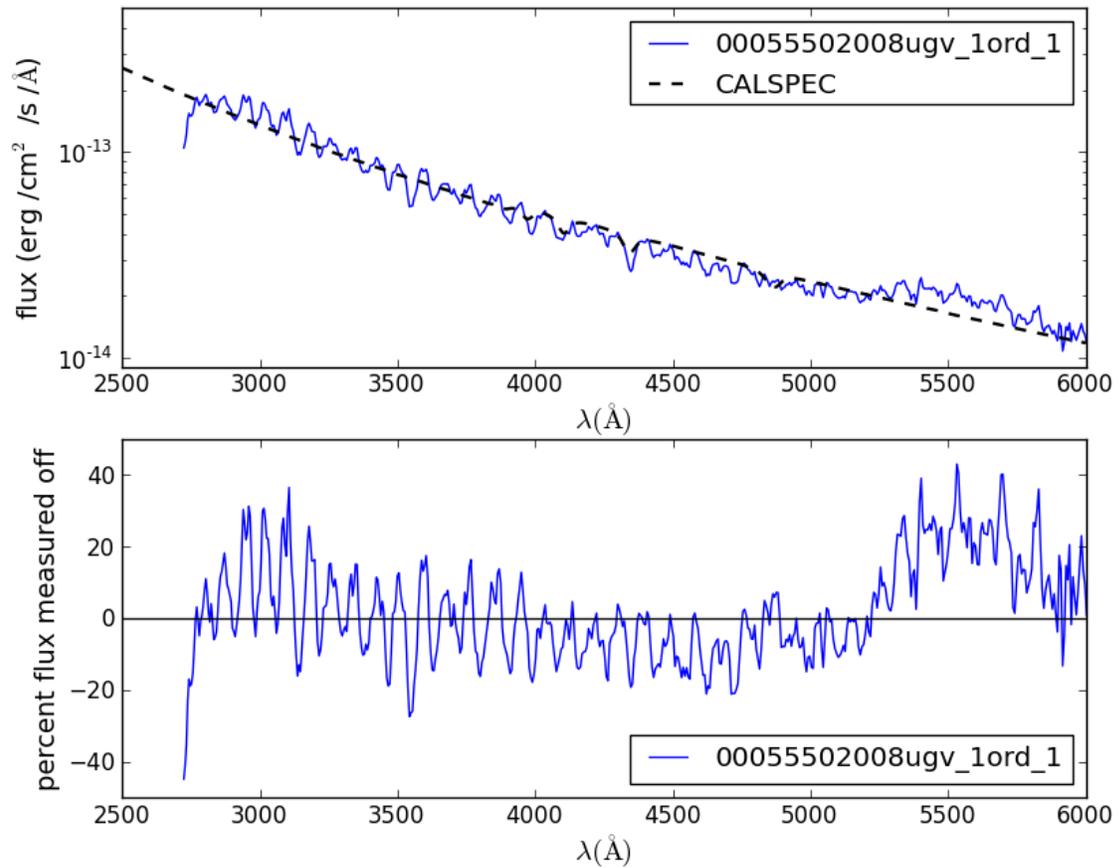


Figure 1.81: This spectrum with anchor at [952,1550].

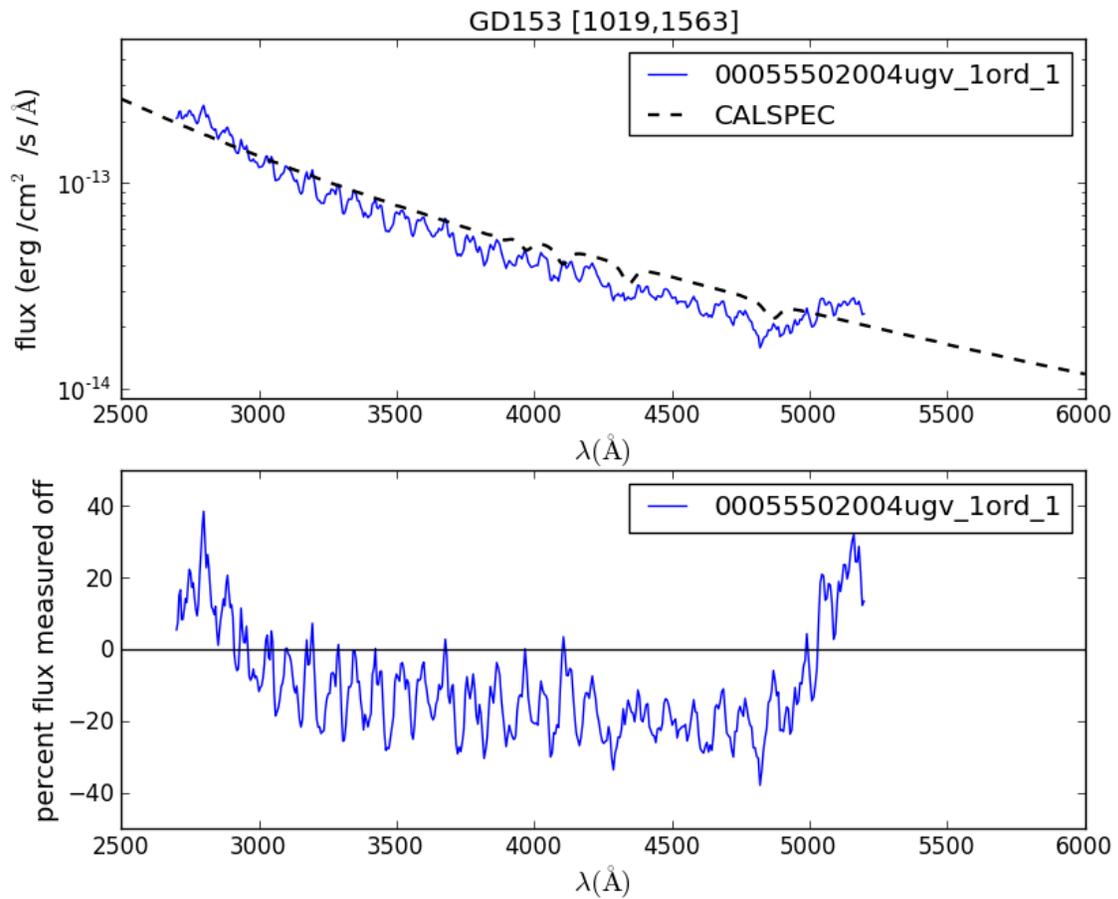


Figure 1.82: This spectrum with anchor at [1019,1563].

GD153 Figure 3

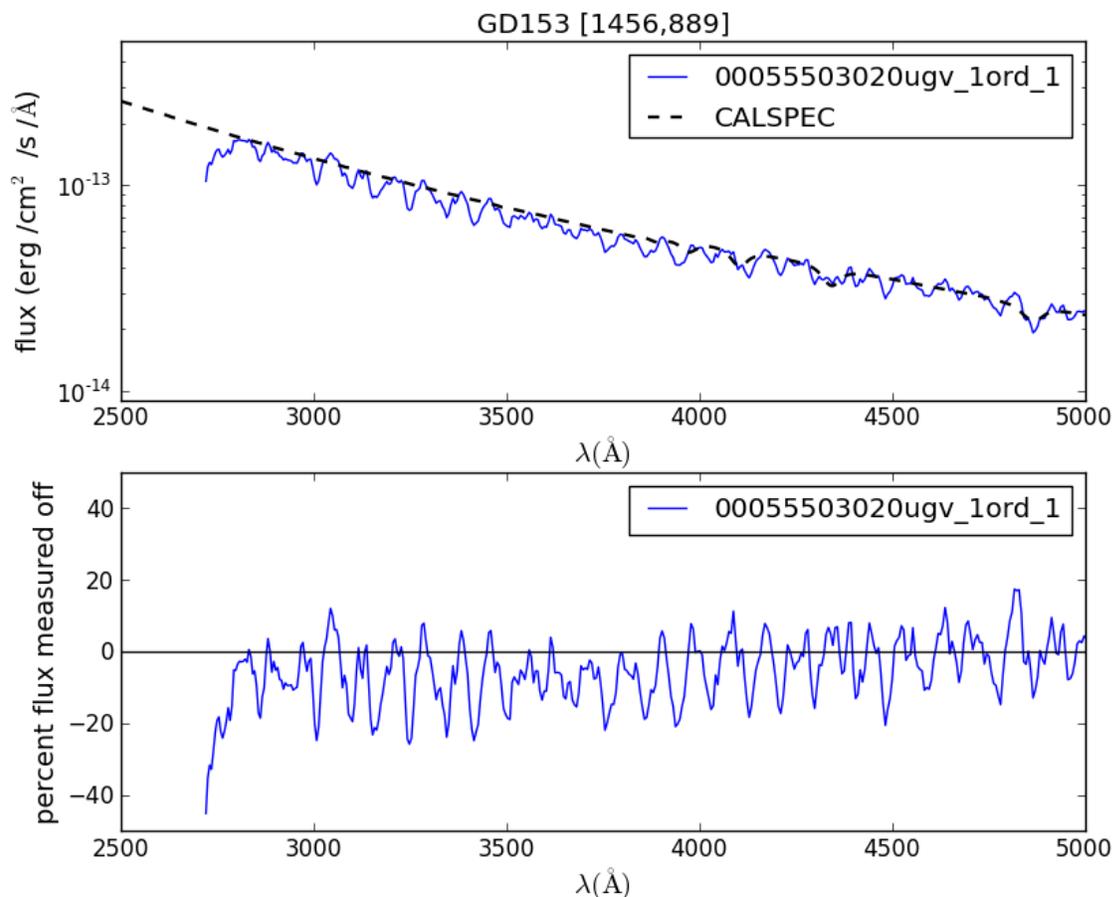


Figure 1.83: This spectrum with anchor at [1456,889].

GD153 Figure 4

1.6 Software

1.6.1 UVOTPY: The Swift UVOT Grism Python Software

Users Guide

The grisms of the UVOT instrument provide a dispersed image for a region of about $17' \times 17'$ on the Sky. This is a description of the Python software written at UCL/MSSL to help reduce the data. The software extends the earlier `Ftool` version as distributed in the `Heasoft` Swift instrument software. This software implements all the recent calibrations, including a correction for coincidence-loss (also known as pile-up) in the photon-counting detectors. It therefore produces a superior product to that from the “`uvotimgrism`” `Ftool`.

The calibration files are bundled with the source code for the software (see *Fetching the program and data*) and you can grab a harcopy of this document PDF .

A simple call can in many cases generate a quick spectrum. The uvotpy module `uvotgetspec` provides the function `getSpec()` which only needs the grism data files, including attitude (auxil) files, target position in decimal degrees, and extension with the desired spectral data.

UVOTPY News

2014-Nov-14

A year ago, uvotpy-1.0 was released. Since then, the coincidence loss theory was finalized, allowing the determination of the corrections for the brightest observable spectra. The UVOT grism calibration will be described in a paper that is in its final stages of preparation and submission to a Monthly Notices of the Royal Astronomical Society is expected by the end of this year. These web pages include the latest updates already.

We continue to suggest using **Ureka**: software together with the *HEADAS* software for spectral extraction with *UVOTPY*.

The *UVOTPY* software is also publicly available at a [github](#) repository and has been registered with the *Astrophysics Source Code Library* <<http://ascl.net>> and the Digital Object Identifier (DOI) is 10.5281/zenodo.12323.

If you use this software, please cite as follows:

Kuin, N.P.M., 2014, *Astrophysics Source Code Library*, record ascl:1410.004

Please report any issues to n.kuin@ucl.ac.uk.

Getting started

Installing UVOTPY You may already have **Python** installed on your system. The current version of *UVOTPY* was written for Python 2.7, though it is expected to work for Python versions 2.5 and 2.6 as well. However, *UVOTPY* has not been adapted for Python 3.x. You can check the Python version by opening an X-window or terminal window, and issuing the command:

```
python --version
```

Other Python software is required. In particular, **Numpy** (version ≥ 1.5), **Scipy** (version $\geq 0.10.1$), **matplotlib** (version ≥ 1.0), **Astropy**, **stsci.convolve** and **stsci.imagestats**, and **ipython** (version ≥ 0.11) is recommended for interactive sessions.

Lately, **Astropy** has been absorbing some packages, like *Pyfits* and *pywcs* and is used to replace some dependencies.

If you do not have the required software on your computer, and want to install everything in close to one go, there are several options.

I now recommend the new **Ureka** package, see *ureka*, which was an easy install.

A quite popular choice is the binary **Enthought Python Canopy Distribution** which is free for academic use (see their Licence). I have not tested this option successfully so far. Canopy may need an additional installation of **stsci.convolve**, **stsci.imagestats**, and **Astropy**.

If you are on the Mac, installing the required packages using **macports** is another option. At one time I also have used **Fink**, with the Wx backend, but recent updates of the packages were often missing.

Headas Not only Python packages are required. The software requires a working, configured **Heasoft Swift distribution** (version > 6.11), and a recent version of the **Swift calibration database (CALDB)**.

Fetching the program and data If you have all those installed and configured, the “UVOTPY” package (2014-10-21) can be downloaded from `uvotpy-2.0.3.tar.gz`, or the latest release from the [github repository](#) .

For testing this installation, some test data can be downloaded from `uvot_testdata.tar.gz`.

Installing the program The UVOTPY package can be installed in your python distribution, although you can also install it somewhere else, like in \$HOME.

To install into the python distribution, make sure that you first activate your python environment. Often there are other distributions of Python on your system, and it is important to start with the environment you will use. I assume the software version is -1.0.:

```
> tar xzf uvotpy-latest.tar.gz
> cd uvotpy-1.0
> sudo python setup.py install
```

To install in your home directory, use

```
> cd $HOME
> tar xzf uvotpy-latest.tar.gz
> cd uvotpy-1.0
> python setup.py install --home=$HOME
```

The scripts will be placed in your \$HOME/bin and the *UVOTPY* modules are in \$HOME/lib/python/uvotpy.

Installing the test data Unpack the `uvot_testdata.tgz` file in one of your local directories. I’ll assume under your HOME directory for this example:

```
> tar xzf $HOME/uvot_testdata.tgz
```

When discussing the example below, I’ll assume it is in the \$HOME/uvotpy_test/ directory.

Next thing to do is to set up the environment. I discuss that for *unix*-like systems only.

Setting up the environment The package installs a number of calibration and other template files which it needs to find. That is done through defining in the environment the variable *UVOTPY*. Say your python packages live in */usr/stsci/pyssg/Python-2.7.3/lib/python2.7/site-packages*, as you can find out by the command:

```
which python
```

and you should now have *uvotpy* installed there. Then, from the command line, use

```
setenv UVOTPY <your-python-site-packages-dir>/uvotpy
```

to define the required environment variable.

If you want to keep things simple, you can just do the following to install in your home directory (assuming the uvotpy version is 1.0):

```
> cd
> tar xzvf uvotpy-latest.tar.gz
> setenv UVOTPY $HOME/uvotpy-1.0
> setenv PYTHONPATH $HOME/lib/python/uvotpy
> setenv PATH {$PATH}:{$HOME}/bin
```

Running from the command line After installation of *uvotpy* and setting up the environment, the program *uvot-grism* can be called to extract spectra.

an example would be:

```
uvotgrism --ra=254.7129625 --dec=34.3148667 --obsid=00055900056 \  
--dir=uvot_test/VGRISM/00055900056/uvot/image
```

There is also a spectrum for the UV grism, but for that example the slit fails to automatically find the right spectrum location. The program is then forced to prompt for a different offset as follows:

```
uvotgrism --ra=254.7129625 --dec=34.3148667 --obsid=00055900056 \  
--anchor_offset=No --dir=UVGRISM/00055900056/uvot/image
```

when prompted enter for example “1” to position the spectrum correctly. The “1” is a one pixel vertical offset which works in this case.

The data for these examples is included the *uvot_testdata.tgz* which needs to be downloaded separately as discussed in *Fetching the program and data*. To save space a set without lenticular images has been used. The test writes the extracted spectrum (fits) files to the current directory. The fits output file names are of the form:

```
"sw"<10-character obsid number>ug<u or v>_lord_<ext>_<method>.pha
```

where ‘ext’ is the extension number of the image in the grism detector image file. <method> flags the way the wavelength anchor position was derived from the sky position, where it is *f* when a lenticular filter was used or *g* when the *uvotgraspcorr.ftool* was used. If you provide your own name for the output files, the “sw”<obsid> will be replaced by your string.

Getting a simple ASCII text file Some people are used to a simple text file with columns for wavelength (in Å), flux (in erg/cm²/s/Å), and flux error. To extract these from your output fits file, do the following.

Assuming that your output file name is *sw00032323001ugu_lord_4_g.pha*, issue the following command:

```
> ftlist sw00032323001ugu_lord_4_g.pha+2 t columns=lambda,flux,fluxerr \  
rownum=no colheader=no > spectrum.txt
```

The file usually contains in the start and end rows with *NULL* values when the flux was negative. Bad data is included, and I recommend checking. The fits output file also lists the coincidence correction factor which should be less than 2 for a good flux.

Getting a quick spectrum with default parameter settings

There are three ways to run the program. The preferred method for me (the developer) is while using an *ipython* shell and the how-to was briefly shown earlier as *Running from the command line*. The second method is as a command-line call, while the third method would be to run within a Python shell. The command line call has not been updated since version 1.0.

Preparations Before starting the spectral extraction, the data for your observation(s) needs to be available on your system. Besides the image files, the spacecraft attitude file and its directory should be available. Although some options were built-in to the program for other arrangements, the only known way that works has the same directory and file structure as that distributed by the Swift archive:

```
<obsid>/\  
<obsid>/auxil/sw<obsid>pat.fits  
<obsid>/auxil/sw<obsid>uat.fits  
<obsid>/auxil/sw<obsid>sat.fits
```

```

<obsid>/uvot/image/sw<obsid>ugu_rw.img (optional)
<obsid>/uvot/image/sw<obsid>ugu_dt.img (required)
<obsid>/uvot/image/sw<obsid>ugu_sk.img
    (example for UV GRISM, *ugv_sk.img for V GRISM)
<obsid>/uvot/image/sw<obsid>uw1_sk.img (if present)
<obsid>/uvot/image/sw<obsid>uw2_sk.img (if present)

```

The `<obsid>` is typically of the form "00032323001", and is one of the required inputs needed. During the ground processing at NASA-GSFC, the initial attitude file `sw<obsid>sat.fits` may be updated to `sw<obsid>uat.fits`, or `sw<obsid>pat.fits` based on further analysis.

The second most important item needed is the source position in **RA, Dec (J2000)** decimal degrees (not in sexagesimal).

Finally, the FITS extension of the spectrum image (`<obsid>/uvot/image/ file`) is needed; when not given it defaults to 1.

The analysis uses the detector image `*_dt.img` files which have been *distortion* corrected. This means that the same correction was applied to the grism image as was found for the lenticular filter images, but additional distortions remain in the grism spectra.

IMPORTANT: Internet access is needed so that the USNO-B1 on-line catalog can be queried from the servers at SAO.

A quick inspection of the sky images `*_sk.img` is useful to make sure the spectrum exposure was correct. There are several other pitfalls, like overlapping orders which can be diagnosed by using DS9 in combination with a DSS image. That is discussed elsewhere in *Basics of UVOT Grism analysis*.

Extracting with IPython The *ipython* shell is more versatile and user-friendly in interactive sessions, with command line finish and recall, for example. A session can be started from the `<obsid>/uvot/image/` directory, or use `cd '<obsid>/uvot/image/'` from within the *ipython* shell to go to the directory. Use the `-pylab` argument to load *matplotlib* for the session, and define the required parameters:

```

ipython --pylab
from uvotpy.uvotgetspec import getSpec
ra, dec = 123.45678, +65.789012
obsid = "00032109002"
ext = 2

```

Note that we imported the `getSpec` program from the `uvotpy.uvotgetspec` module of the *uvotpy* package. We can now do a simple call to the program. Assuming the `obsid` and position are all correct:

```
getSpec(ra,dec,obsid,ext, fit_second=False, clobber=False)
```

I have set here one more parameter "fit_second" to "false", since this can cause problems when the second order completely overlaps the first order, i.e., is not offset in the cross-dispersion direction.

Three graphs will appear.

- The first one to quickly view the extraction is positioned approximately right, and crosses are plotted at about 200A offsets at the estimated location of the spectrum.
- In the second graph the extracted raw data in the form as total counts, i.e., (no background subtraction, and not as rates), aligned with the image section obtained after rotating to the dispersion). Also, a *very* rough guess of the second order counts is made based on those in the first order. It tends to over-estimate, sometimes as much as a factor of two.
- In the third graph, the flux calibrated spectrum.

It should be noted that the flagging of bad areas is at the moment *pretty bad*, and needs to be looked at again. I noticed that when running `uvotgrism` from the command line the figures don't always show.

The spectra will be written to a file, while also a background file is made for use with XSPEC. However, the RMF file is not produced automatically, since it involves a very slow piece of code. The RMF is useful, since the spectral resolution gets worse to the red, and is especially affecting wavelengths above 400nm in the UV Grism spectra.

Basics of UVOT Grism analysis

The main reason for writing this section is to help you navigate around some of the pitfalls that I have encountered when analysing the UVOT Grism spectra; some of which are not always obvious.

Use DS9 with the grism sky image and DSS Even for one used to see many grism images, locating the source spectrum can be difficult. The automated processing lines up the peaks of the zeroth orders of the sources in the image with the WCS. If everything goes perfect, we can load the grism image and DSS image in DS9, *blink* them, and the zeroth orders and DSS sources line up nicely. If they don't, and there are close-by orders that make interpretation too confusing, see *Problem with positions in the grism image*.

Blinking the DSS and grism sky image while zooming in on the first order will allow a quick determination of contaminating zeroth orders that fall on the first order spectrum. Also, bright sources in the dispersion plane, behind the source can easily be a problem especially in the UV part of the spectrum where the target photon sensitivity is low.

Problem with positions in the grism image In the following steps you can either use the *sky* image (filename includes *_sk.img*), or the *detector* image (filename includes *_dt.img*). The detector image only works for blinking in later versions of DS9 when the "WCS" in DS9 for the image has been set to WCS-S.

First of all, we need to be sure that we have the header of the sky(det) image in the best possible form. That is, check the header keyword ASPCORR is set to "graspcorr". From the terminal command line, assuming the present working directory is the <obsid>/uvot/image directory and we have a UV Grism image:

```
$ftlist sw<obsid>ugu_sk.img hk include=aspcorr
```

The value should be printed for each extension in the FITS file. If not set, the `uvotgraspcorr` script should be executed, though `uvotgraspcorr` fails in a few percent of cases if it cannot reach the catalog server; for some fields with not enough sources of the right brightness in the field; and for some images with large galaxies. It is very difficult to say off-hand how to change the internal parameters for the program to ensure success. This usually means that the keywords accounting for the distortion of the zeroth order positions are not added to the WCS header keywords. The distortions are increasing to the edges of the detector, where the problem is worst.

Even when `uvotgraspcorr` finds a solution, there remains some error in the aspect solution larger than that when using a lenticular image for the anchor position, see *anchor_error*.

If the zeroth orders are clearly offset from the expected positions when blinking the grism sky image with the corresponding DSS image, the header WCS keywords in the grism image sky file can be edited "by hand" to line up the grism zeroth order coordinates. It is smartest to do that for the region close to the target source location, as that will clearly minimize the errors in the analysis.

Note that the grism extraction is *not* dependent on the aspect correction when the grism image was taken paired with an image in one of the lenticular filters.

Guide for advanced processing of the grism spectra

Various topics are discussed. This is a work in progress, and more in-depth examples are planned, including perhaps example by video.

Observing strategies

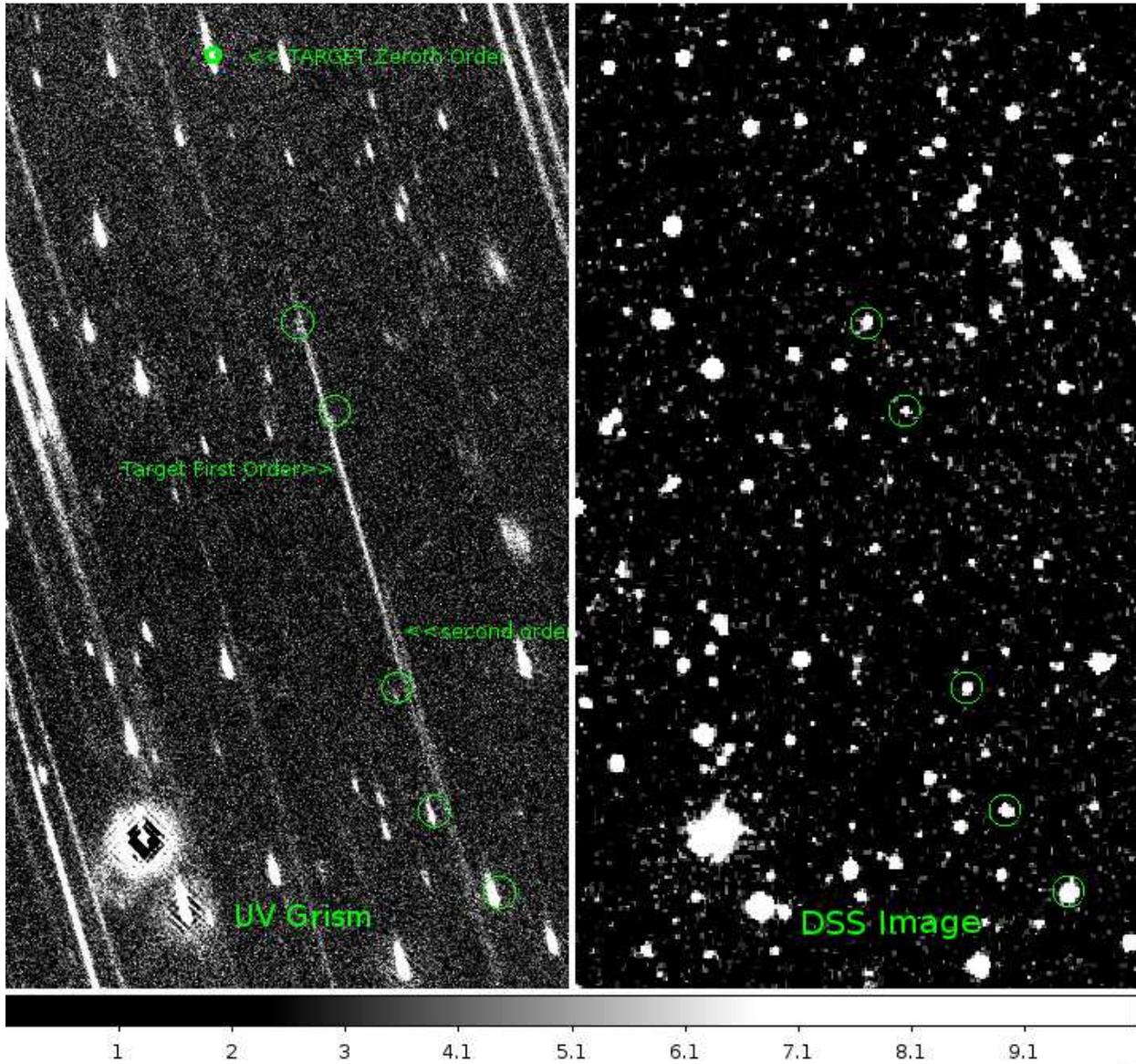


Figure 1.84: The background zeroth orders that lie close to the first order have been marked.

Slew-in-place To start the observation, the spacecraft will execute a slew to the desired sky location. The accuracy of the slew depends on the size of the slew, and is usually of the order of 1-2 arc minutes. When observing with the target in the centre of the field of view, that is usually a good enough accuracy. However, if multiple spectra are required or when using an offset (see *observing_with_offset*) a better positioning is obtained by executing a second slew after the first one, called a "slew-in-place" which will refine the positioning of the source to a few (5-10) arc seconds.

Early grism observations did not use a slew-in-place, and were often found at offsets on the detector that were large enough that they could not be calibrated using the early calibration which was valid for the centre of the detector only.

Using an offset to place the spectrum somewhere else on the detector The *Swift* operations were designed as to place the target in the centre of the field of view. This has obvious advantages. During the year, the spacecraft position with respect to the sun is adjusted, leading to changes in the optimal roll-angle range. Since changes in roll angle rotate the observed field with respect to the sky around the boresight, sources near the boresight will be least affected.

The roll angle affects where a source will fall on the detector when there is an offset in pointing. A source that would fall on the upper right of the detector in April, will fall on the lower left 6 months later in October.

For the planning of offset observations this has implications. For a given offset, spacecraft roll angle and target sky coordinates, the sky coordinates for the centre of the field of view (the boresight) have to be calculated. These are then assigned an observation ID, and the observation is commanded using the coordinates that were calculated for the centre of the field of view. At present that is done by hand by the planners. This means that these kind of observations need some lead time to be done.

For the grism, the first offset observations were done for the calibration, nova U Sco and for SN2011fe in M101. In the case of the calibration observations it was done to determine the instrumental characteristics enabling the anchor point, wavelength, and effective area calibrations. For the supernova the offset was done since the source was very bright, and an offset would give less contamination of the first order, and also provide some indication of the nova emission lines in second order.

Faint sources For this discussion, faint sources are those with count rates up to the count rate in the background.

As the noise in faint source observations is background limited, it is natural to use the clocked mode for the grism which has a lower background in the upper left of the detector. Usually that mostly benefits the longer wavelengths most, as the blue part of the spectrum usually still falls in the higher backgrounds.

A possibility is observing with an offset, see *observing_with_offset*, placing the spectrum closer to the area of lower background, but this carries the risk of the spectrum falling in the area of reduced sensitivity, or even being lost.

Spectral Extraction

Centering the extraction slit on a different spectrum Sometimes the target is fainter than a nearby spectrum, and the optimisation of the extraction slit picks the wrong spectrum. The reason is that the automated extraction searches a slightly larger region for a spectrum in order to allow for anchor errors. It pays therefore to examine the position of the slit on the image and make sure that you got the desired target.

In some cases I have had to go and blink the detector image in DS9 with a catalog of source positions. Lately my preferences is the GSC2.3 catalog, set with a filter:

```
"$Fmag >2 && $Fmag < 15"
```

Once you know what your target spectrum is in the rotated image, you can re-extract with an offset range for the optimisation, where the range is given in the center pixel y-coordinate, and pixel range. The default position is centered on y=100, so a spectrum higher up, say around y=115 can be extracted by passing the parameter:

```
offsetlimit=[115,4]
```

will search the rotated image between y=110 and y=119.

Extraction with a smaller aperture Due to coincidence loss using a smaller aperture than the default is only recommended for faint sources. Coincidence loss is seen to change the width and shape of the point spread function. Therefore the aperture is set to a fixed size of 2.5 in terms of the width parameter of the fitted gaussian to the spectrum normal to the dispersion.

In order to change the aperture, you need to *import* the *uvotgetspec* module while running the extraction from *ipython*. Then, set the following parameter:

```
trackwidth = 2.5
```

Change the 2.5 value to, for example, 1.5 to get a smaller extraction aperture. Then do the spectral extraction again.

Post-extraction processing

Correcting wavelengths errors Errors in the predicted anchor position cause shifts in the wavelength at the anchor. These can get quite large when the anchor position was determined using *uvotgraspcorr* only. Since the dispersion equation is not completely linear large wavelength errors affect the wavelength at the end of the wavelength range the most. Once the main wavelength error has been corrected by a linear term (a shift), the dispersion equation has to be reapplied and the wavelength scale must be recalculated.

To do this, I provide now code to post-process the extracted spectrum. Since version 2.0.3 the *UVOTPY* distribution includes a module *uvotspec*. The *adjust_wavelength_manually* program allows interactive adjustment of the wavelengths, by sliding the spectrum till the wavelengths are better. Next the wavelength scale is recalculated using the dispersion equation, and the new solution is overlotted. Sometimes the process needs to be repeated twice, because the main parameter for adjustment is the wavelength shift at the anchor.

Summing spectra A faint source observation of 5ks will be spread over several orbits and images. After processing each one of the images into spectral files they can be aligned in wavelength and summed using the *sum_Extimage* program. Input is a list of the filenames to be processed.

The program was written to align the spectra interactively, or non-interactive from a list of wavelength shifts. Lately, I use correct the spectral files using *_fixing_wavelength_offsets* first, then call *sum_Extimage* with a list of zero shifts. The program also has an autocorrelator, but I have not been completely happy with the result of using the autocorrelator. The program is due to be rewritten at some point.

Lexicon

Explanation of terms:

Term	Explanation
Lenticular filter	A flat filter. One of uvw2, uvm2, uvw1, u,b,v,wh
V Grism	Grism with peak response in the optical (420nm)
UV Grism	Grism optimised in the UV (260nm)

UVOTPY API

API Documentation

To get the options to use with the command-line program *uvotgrism*, use

```
uvotgrism --help
```

Information on the main interactive program can be found from the command-line by typing:

```
pydoc uvotgetspec.getSpec
```

Release notes for UVOTPY

Notes to version 2.0 Versions 2.0.0 and 2.0.1 had the initial version of the improved treatment of the coincidence loss, but without the fine-tuned coi-area.

In versions 2.0.2 and 2.0.3 the coincidence loss has been treated using the standard theory of coincidence loss corrections with fine tuned coi-area to each grism mode.

Functionality is basically the same as described below for version 1.0, with some bug fixes and updates, many due to the inclusion of increasingly more [Astropy](#) software.

Notes to version 1.0 Version 1.0 2013-11-01

General

This code is designed for processing Swift UVOT Grism images

The goal is to process the spectra in the image, to do quality control and source identification. Because the UVOT is a photon counting detector, the error handling must keep track of the errors which require in principle the total exposure, the background exposure, and the exposure time. The data quality for each pixel should be flagged also (e.g., data dropouts, scattered light rings, halo around bright features), but flagging is only done for nearby zeroth orders. While the count rate errors follow from the observed binomial nature of the detector, e.g., Kuin & Rosen (2008, MNRAS, 383,383) - though that prescription was for point sources. A heuristic method was used to develop a correction for coincidence loss used in this program.

Details of the accuracy and reliability of the calibration and software are soon to be submitted to the Monthly Notices of the RAS.

This program extracts the spectra, applies the wavelength calibration file to find anchor position and wavelength dispersion. The flux calibration is valid over the whole detector and depends on the effective area and coincidence loss correction. In *both grisms* the effective area was determined at several offset positions. The accuracy of the flux in the *uv grism* is of order 5% in the centre and about 10% at other locations on the detector. In the *visible grism* the accuracy of the flux is 20%, which error is dominated by that in the coincidence loss correction.

In the nominal grism mode (wheelpos = 200, 1000) the response varies by about 5% from centre to about 200 pixels from the edge. In the clocked grism modes (wheelpos = 160, and 955) the response has a strong drop when the spectrum falls in the upper left corner. For the rest of the image the response varies less than ~20% in the clocked modes.

Before using this code, it is recommended to reprocess the raw image using the *mod-8 correction*. Use a CALDB later than May 2010 for an improved distortion correction. The grism detector image should be attitude corrected using *uvotgraspcorr*. Check the header keyword `ASPCORR='GRASPCORR'`.

Find out the RA and DEC position in decimal degrees from the USNO-B1 catalog for your object since the UVOT aspect corrections also use the USNO-B1 positions, so you will avoid a systematic error in positions which will translate to a shift in the wavelengths derived.

You need to set the environment variable UVOTPY to point to the directory with the UVOTPY code and spectral calibration files.

The input data files may have to be decompressed before running the program, although it will try to do that itself, but does not recompress the files again.

The program was developed running in iPython, and it is suggested to run interactively in iPython rather than run as it this script.

Second order extraction and calibration are only treated very roughly at this time. Zeroth orders are only minimally identified. Third order location is approximate.

Main functions

getSpec : main call for spectral data extraction

Other uvotpy functions

curved_extraction: set curvature of orders, set quality flags, get spectral data
extractSpecImg : get sub image
findBackground : get background
get_components : extract first, second, third order components
getCal : get the (wave) calibration files
predict_second_order : use first order to predict second order (very rough)
coi_func : return an interpolating function(wave) for the coincidence loss of a spectrum [experimental]

Specialized functions are in modules

uvotgetspec: repository of main functions
uvotio: writes file output
uvotio.rate2flux : convert count rate to flux
uvotplot : plot routines
uvotmisc : miscellaneous routines

Files

The program assumes all data files are available in either the working directory, or the directory structure complies with the Swift project standard and is run from the <obsid>/uvot/images directory, while the attitude file is available in the <obsid>/aux directory. There is rudimentary support for running from a remote directory on the same device, but the program will write some files to both the current and the data directories.

The flux-calibrated 1st order spectrum is available in the second extension of the output file.

From version 1.0 onwards, the file name includes a flag “_f” for when lenticular filter image(s), or “_g” when “uvotgraspcorr” aspect corrections were used to derive the anchor position. Both methods give similar uncertainties, but for the same field uvotgraspcorr will give more consistent results, while the lenticular filter method works when uvotgraspcorr cannot find an aspect correction (in that case the uncorrected pointing position from the star trackers will be used).

History

2013-Oct-31 Paul Kuin Rewritten uvotwcs to fix a bug in the calculation of the pointing. Small fixes all throughout. Output file names will have a flag for the anchor point method used.

2013-May-23 Paul Kuin revised the uv grism clocked mode wavecal for first and second order. Missing still is second order effective area.

March 2013: Flux calibration for the uv-grism The new calibration file has multiple extensions. The first extension is the original flux calibration, which did not correct for the coincidence loss and which is to be used with the “Ftool” **uvotimgism**. Then follow extensions with the measured effective areas at various positions on the detector. These have been corrected for coincidence-loss during the calibration. The number depends to some extent on how much the effective area varies with position. Finally, a normalised flux from a rescaled model is included that can be used to extrapolate from one of the effective areas at a nearby position. A new routine in uvotio reads the effective area and model scaling to offset if present.

A start has been made with using Sphinx (<http://sphinx.org>) for documentation on the web. This includes changing the inline documentation of the software. Keep an eye on http://ucl.ac.uk/mssl/astro/space_missions/swift.

A preliminary flux calibration for the visual grism has been made. It supersedes the old calibration which did not take any coincidence-loss correction into account. However, future improvements are expected.

October 2012: Initial extension of flux calibration uv-grism The software has been updated to support a working coi-correction, though not the final one, and use the new flux calibration where available. The calibration file closest to the position on the detector image will be used. More calibration files will be supplied when they become available. The comparison of the new calibration and coi-loss correction with calibration spectra can be seen on my web site.

The package has now been created using Python distutils.

March 2012: An experimental coincidence loss correction The background in the grism is quite high and in itself experiences about 3% coi-loss. The extended nature of the background has been cause for concern, but tests reported in Breeveld et al. (2010) did in fact show that the correction method from Poole et al. (2008) works well also in that case. However, the case for an extended linear feature, like a spectrum has not

March 2012: An experimental coincidence loss correction The background in the grism is quite high and in itself experiences about 3% coi-loss. The extended nature of the background has been cause for concern, but tests reported in Breeveld et al. (2010) did in fact show that the correction method from Poole et al. (2008) works well also in that case. However, the case for an extended linear feature, like a spectrum has not been studied. The symmetries are different, and it is not a priori clear how to best generalize the method from Poole et al. But with estimated coi-loss peaking at 3% for a 16th mag WD and estimated 7% for a 14th mag WD, with maximum coi-loss close to 50% for a 12th mag WD, it is clear that a solution is desirable as part of a flux calibration that is meaningful. The WD are probably the worst case, since they are bright around 3000Å where the effective area (and the count rate) peaks

March 2012: discovery of variable sensitivity over face of uv clocked detector image Over most of the detector, the sensitivity only changes by a few percent. So it was a bit of a shock to see that in the clocked uv grism images, the detector sensitivity drops quite a lot in the upper left corner of the detector. More so, since this area had been selected to place spectra to completely avoid the zeroth orders. This is under investigation. We are using the Zemax optical model to get a quantitative, though approximate, measure of this effect.

April 2012: predicting the uv grism exposure For some faint objects, there is often uvot photometry in the uv filters available. It is of a given magnitude, but what does that mean for the uv grism. I made a tool that will make a rough estimate of the exposure time needed to get a certain signal-to-noise, assuming a certain background and given a magnitude of the object as observed in a uvot filter. There is loads of uncertainty when the source is faint, and the background over the UV clocked grism varies. It defaults to 0.16c/s/pix(across the spectrum) but a value of 0.05 may happen depending of where the (uv part of) the spectrum lies on the detector. So for now, I put in the most reasonable value. If the source is getting faint, the exposure time will climb through the roof. Experience must tell how good it is and how to use it.

Alternatively a spectrum can be put in a will give a magnitude in the lenticular filters. The `uvotphot.py` file will be needed.

April 2012: estimate of the zeroth order effective area The uv grism efficiency in the zeroth first and second order estimate were needed for calculating the flux sensitivity over the detector using the Zemax optical model. A rough approximation of the zeroth order effective area was determined in the course of that work. Details have been written in a report.

April 2012: determining the effective area in the uv detector

Testing The software was tested with the following installations:

A. dated October 2013 Debian Squeeze4

1. I have a working installation of the HEASARC CALDB and HEASOFT (swift)
2. I installed Ureka as described in *ureka*.
3. tested using various spectra, both from the command line and interactive.

B. dated March 2013 (1) an existing installation of the STScI python, including the Scipy software, for Python version 2.7.3 A package can be downloaded from STScI. Note that you also will need to install `gfortran` (link at the bottom of the STScI Python page) to make it work. If you have `pyraf` installed, it may already be there.

(2) Installation of the Heasoft Ftools/SWIFT software, release not earlier than version 6.10. Download from <http://heasarc.gsfc.nasa.gov/docs/software/lheasoft/> .

(3) Installation of the HEASARC CALDB for Swift, version later than Oct 13,2011. Download the CALDB for Swift from: http://heasarc.gsfc.nasa.gov/docs/heasarc/caldb/caldb_supported_missions.html

(4) Installation of Mink's WCSTOOLS, in particular *scat*. On unix/linux type system the command *which scat* should return the location. If it does not return anything, install: <http://tdc-www.harvard.edu/software/wcstools/wcstools-3.8.4.tar.gz> and remove or rename the program *cphead* that it installs, since it clashes with *cphead* in Heasoft as used by the *uvotproduct ftool*.

(5) Installation of *CDSCLIENT* which provides the 'sesame' name resolver. [optional] <http://cdsarc.u-strasbg.fr/doc/cdsclient.html>

(6) setup of the environment: (“\$HOME/.cshrc” for *csh* users or “\$HOME/.bashrc” for *bash* or *sh* users)

- add the directory with this software to the PYTHONPATH environment variable.
- set UVOTPY to the directory with this software
- add the WCSTOOLS/bin and cdsclient directories to your PATH

uvotgetspec

Major functions

getSpec ()

`uvotgetspec.getSpec (RA, DEC, obsid, ext, indir='.', wr_outfile=True, outfile=None, cal-file=None, fluxcalfile=None, use_lenticular_image=True, offsetlimit=None, anchor_offset=None, anchor_position=[None, None], background_lower=[None, None], background_upper=[None, None], background_template=None, fixed_angle=None, spextwidth=13, curved='update', fit_second=False, predict2nd=True, skip_field_src=False, optimal_extraction=False, catspec=None, write_RMF=False, get_curve=False, fit_sigmas=True, get_sigma_poly=False, lfilt1=None, lfilt1_ext=None, lfilt2=None, lfilt2_ext=None, wheelpos=None, interactive=True, sumimage=None, set_maglimit=None, plot_img=True, plot_raw=True, plot_spec=True, zoom=True, highlight=False, clobber=False, chatter=1)`

Makes all the necessary calls to reduce the data.

Parameters `ra, dec` : float

The Sky position (J2000) in **decimal degrees**

obsid : str

The observation ID number as a **String**. Typically that is something like “00032331001” and should be part of your grism filename which is something like “sw00032331001ugu_dt.img”

ext : int

number of the extension to process

kwargs : dict

optional keyword arguments, possible values are:

- **fit_second** : bool

fit the second order. Off since it sometimes causes problems when the orders overlap completely. Useful for spectra in top part detector

- **background_lower** : list

instead of default background list offset from spectrum as list of two numbers, like [20, 40]. Distance relative to spectrum

- **background_upper** : list

instead of default background list offset from spectrum as list of two numbers, like [20, 40]. Distance relative to spectrum

- **offsetlimit** : None,int,[center,range]

Default behaviour is to determine automatically any required offset from the predicted anchor position to the spectrum, and correct for that. The automated method may fail in the case of a weak spectrum and strong zeroth or first order next to the spectrum. Two methods are provided:

(1) provide a number which will be used to limit the allowed offset. If within that limit no peak is identified, the program will stop and require you to provide a manual offset value. Try small numbers like 1, -1, 3, etc..

(2) if you already know the approximate y-location of the spectrum at the anchor x-position in the rotated small image strip around the spectrum, you can give this with a small allowed range for fine tuning as a list of two parameter values. The first value in the list must be the y-coordinate (by default the spectrum falls close to y=100 pixels), the second parameter the allowed adjustment to a peak value in pixels. For

example, [105,2]. This will require no further interactive input, and the spectrum will be extracted using that offset.

- **wheelpos**: { 160,200,955,1000}

filter wheel position for the grism filter mode used. Helpful for forcing Vgrism or UVgrism input when both are present in the directory. 160:UV Clocked, 200:UV Nominal, 955:V clocked, 1000:V nominal
- **zoom** : bool

when False, the whole extracted region is displayed, including zeroth order when present.
- **clobber** : bool

When True, overwrite earlier output (see also outfile)
- **write_RMF** : bool

When True, write the rmf file (will take extra time due to large matrix operations)
- **use_lenticular_image** : bool

When True and a lenticular image is present, it is used. If False, the grism image header WCS-S system will be used for the astrometry, with an automatic call to uvotgraspcorr for refinement.
- **sumimage** : str

Name summed image generated using `sum_Extimage()`, will extract spectrum from summed image.
- **wr_outfile** : bool

If False, no output file is written
- **outfile** : path, str

Name of output file, other than automatically generated.
- **calfile** : path, str

calibration file name
- **fluxcalfile** : path, str

flux calibration file name or “CALDB” or None
- **predict2nd** : bool

predict the second order flux from the first. Overestimates in centre a lot.
- **skip_field_src** : bool

if True do not locate zeroth order positions. Can be used if absence internet connection or USNO-B1 server causes problems.
- **optimal_extraction** : bool, obsolete

Do not use. Better results with other implementation.
- **catspec** : path

optional full path to the catalog specification file for uvotgraspcorr.

- **get_curve** : bool
option to supply the curvature coefficients of all orders by hand. implemented but not tested.
- **fit_sigmas** : bool
fit the sigma of trackwidths if True (not implemented, always on)
- **get_sigma_poly** : bool
option to supply the polynomial for the sigma (not implemented)
- **lfilt1, lfilt2** : str
name if the lenticular filter before and after the grism exposure (now supplied by fileinfo())
- **lfilt1_ext, lfilt2_ext** : int
extension of the lenticular filter (now supplied by fileinfo())
- **plot_img** : bool
plot the first figure with the det image
- **plot_raw** : bool
plot the raw spectrum data
- **plot_spec** : bool
plot the flux spectrum
- **highlight** : bool
add contours to the plots to highlight contrasts
- **chatter** : int
verbosity of program
- **set_maglimit** : int
specify a magnitude limit to search for background sources in the USNO-B1 catalog
- **background_template** : numpy 2D array
User provides a background template that will be used instead determining background. Must be in counts. Size and alignment must exactly match detector image.

Returns None, (give_result=True) compounded data (Y0, Y1, Y2, Y3, Y4) which :
are explained in the code, or (give_new_result=True) a data dictionary. :

Notes

Quick Start *getSpec(ra,dec,obsid, ext,)* should produce plots and output files

Which directory?

The program needs to be started from the CORRECT data directory. The attitude file [e.g., “sw<OBSID>pat.fits”]is needed! A link or copy of the attitude file needs to be present in the directory or “././auxil/” directory as well.

Global parameters

These parameters can be reset, e.g., during a (i)python session, before calling getSpec.

- **trackwidth** : float width spectral extraction in units of sigma. The default is trackwidth = 2.5 The alternative default is trackwidth = 1.0 which gives better results for weak sources, or spectra with nearby contamination. However, the flux calibration and coincidence-loss correction give currently inconsistent results. When using trackwidth=1.0, rescale the flux to match trackwidth=2.5 which value was used for flux calibration and coincidence-loss correction.
- **give_result** : bool set to False since a call to getSpec with this set will return all the intermediate results. See returns

When the extraction slit is set to be straight `curved="straight"` it cuts off the UV part of the spectrum for spectra located in the top left and bottom right of the image.

Key functions

coi_func()

```
uvotgetspec.coi_func (pixno, wave, countrate, bkgrate, frametime=0.0110329, background=False,
                    wheelpos=160, area=414, option=1, fudgespec=1.0, coi_length=29,
                    siglcoef=[], trackwidth=0.0, sigmal_limits=[2.6, 4.0], ccc=[], ccb=[],
                    ca=[], cb=[], debug=False, chatter=5)
```

Compute the coincidence loss correction factor to the (net) count rate as a function of wavelength

Parameters **pixno** : array-like

pixel number with origin at anchor

wave : array-like

wavelength in A, *must be monotonically increasing*

countrate : array-like

input total count rate for the coi aperture (default coi_width pixels wide)

bkgrate : array-like

background rate for the coi aperture (default coi_width pixels wide)

kwargs : dict

- **frametime** : float
CCD frame time in seconds
- **option** : int
• **option = 1** [(default) classic coi-loss, for box 16 pixels wide,] 414 pix² area
- **background** : bool
if the background is *True* an interpolated function for the coi correction factor in the background count rate is returned
if the background is *False* an interpolated function for the coi correction factor in the net target count rate is returned
- **wheelpos** [[160,200,955,1000]] filter wheel position, one of these values.

Returns **coi_func** : `scipy.interpolate.interp1d`

if **background** is *True* an interpolated function for the coi correction factor in the background count rate while if **background** is *False* an interpolated function for the coi correction factor in the net target count rate is returned

v [bool] only for spectrum. **v=True** points are valid, **False** points mean observed rate per frame is too large.

Notes

defaults to the background coincidence loss equivalent to an area of “area=414” sub-pixels.

Both the `sprate` and `bgrate` are required, as the points in `sprate` that are not valid are used to mask the `bgrate`.

- 2012-03-21 NPMK initial version
- 2014-06-02 NPMK start using a fixed coi-area,remove old options,** change meaning parameters
- 2014-07-23 NPMK use calibrated values of `coi-box` and `factor`

`findBackground()`

`uvotgetspec.findBackground` (*extimg*, *background_lower*=[None, None], *background_upper*=[None, None], *yloc_spectrum*=100, *smo1*=None, *smo2*=None, *chatter*=2)

Extract the background from the image slice containing the spectrum.

Parameters `extimg` : 2D array

image containing spectrum. Dispersion approximately along x-axis.

background_lower : list

distance in pixels from `yloc_spectrum` of the limits of the lower background region.

background_upper : list

distance in pixels from `yloc_spectrum` of the limits of the upper background region.

yloc_spectrum : int

pixel *Y* location of spectrum

smo1 : float

smoothing parameter passed to smoothing spline fitting routine. *None* for default.

smo2 : float

smoothing parameter passed to smoothing spline fitting routine. *None* for default.

chatter : int

verbosity

Returns `bg` : float

mean background

bg1, bg2 : 1D arrays

`bg1` = lower background; `bg2` = upper background inherits size from `extimg.shape x-coordinate`

bgsig : float

standard deviation of background

bgimg : 2D array

image of the background constructed from `bg1` and/or `bg2`

bg_limits_used : list, length 4

limits used for the background in the following order: lower background, upper background

(**bg1_good, bg1_dis, bg1_dis_good, bg2_good, bg2_dis, bg2_dis_good, bgimg_lin**) : tuple

various other background measures

Notes

Global parameter

•**background_method** : { 'boxcar', 'splinefit' }

The two background images can be computed 2 ways:

1. **'splinefit'**: **sigma clip image, then fit a smoothing spline to each** row, then average in y for each background region
2. **'boxcar'**: **select the background from the smoothed image created** by method 1 below.
3. **'sigmaclip'**: **do sigma clipping on rows and columns to get column** profile background, then clip image and mask, interpolate over masked bits.

extimg is the image containing the spectrum in the 1-axis centered in 0-axis *ank* is the position of the anchor in the image

I create two background images:

1. split the image strip into 40 portions in x, so that the background variation is small compute the mean sigma clip (3 sigma) each area to to the local mean replace out-of-image pixels with mean of whole image (2-sigma clipped) smooth with a boxcar by the smoothing factor
2. compute the background in two regions upper and lower linearly interpolate in Y between the two regions to create a background image

bg1 = lower background; bg2 = upper background

smo1, smo2 allow one to relax the smoothing factor in computing the smoothing spline fit

get_radec ()

`uvotgetspect.get_radec (file='radec.usno', objectid=None, tool='astropy', chatter=0)`

Read the decimal ra,dec from a file or look it up using the objectid name from CDS

Parameters file: str, optional :

path, filename of ascii file with just the ra, dec position in decimal degrees

objectid : str, optional

name of object that is recognized by the (astropy.coordinates/CDS Sesame) service if not supplied a file name is required

tool : str

name tool to use; either 'astropy' or 'cdsclient'

chatter : int

verbosity

Returns ra,dec : float

Position (epoch J2000) in decimal degrees

makeXspecInput ()

`uvotgetspec.makeXspecInput (lamdasp, countrate, error, lamda_response=None, chatter=1)`

Convert the count rate spectrum per pixel into a spectrum on the given bins of the response function.

Parameters lamdasp : array

wavelengths spectrum

countrate : array

count rates at wavelengths

error : array

errors at wavelengths

kwargs : dict

- **lamda_response** : array

the wavelength for the response bins

- **chatter** : int

verbosity

Returns lambda : array

wavelengths of the bins

countrate : array

count rate in the bins

error : array

errors in the bins

Notes

errors are summed as $\sqrt{\sum (\text{errors}^{**2})}$

plan_obs_using_mags ()

`uvotgetspec.plan_obs_using_mags (S2N=3.0, lentifilter=None, mag=None, bkgrate=0.16, coi=False, obsfile=None, grism='uv')`

Tool to compute the grism exposure time needed to get a certain S/N in the filterband given observed magnitude in lentifilter.

Parameters S2N : float

signal to noise desired

lentifilter : str, {'uvw2','uvm2','uvw1','u','b','v'}, optional if *obsfile* given

lenticular filter in which a magnitude is available

mag : float, optional if *obsfile* given

measured magnitude in *lentifilter*.

bkgrate : float

the count rate in the background. This parameter determines for weak spectra to a large extent what exposure time is required.

coi : bool

apply coincidence-loss correction ? *not yet implemented*

obsfile : path, str, optional if *lentifilter*, 'mag' given

ascii filename with two columns wave, flux or a fits file with the spectrum in the second extension

grism : str, {'uv' }

Returns An estimate of the required exposure time is printed :

Notes

Lentifilter should be one of: uvw2, uvm2, uvw1, u, b, v

Assumed source is faint - no coi (can later add coi)

The exposure time will ramp up quickly once the target gets too faint. The background in the clocked uv grism varies and can be lower, like 0.06 depending of where the spectrum is put on the detector. So we could update this program at some point with the uv clocked background variation in it. Typically background values are below 0.1 c/s/arcsec. Higher backgrounds are found in crowded fields.

If obsfile is given, then calculate the magnitudes using the spectrum from the obsfile

TO DO: V grism placeholder

- 16 April 2012, initial version, Paul Kuin

spec_curvature ()

uvotgetspec.**spec_curvature** (*wheelpos, anchor, order=1*)

Find the coefficients of the polynomial for the curvature.

Parameters **wheelpos** : int, {160,200,955,1000}

grism filter position in filter wheel

anchor : list, array

anchor position in detector coordinates (pixels)

order : int

the desired spectral order

Returns Provides the polynomial coefficients for $y(x)$:

Notes

The curvature is defined with argument the pixel coordinate in the dispersion direction with reference to the the anchor coordinates in det-img coordinates. The polynomial returns the offset normal to the dispersion.

- 2011-03-07 Paul Kuin, initial version
- 2011-08-02 fixed nominal coefficients order=1

spectrumpixshift ()

uvotgetspec.**spectrumpixshift** (*w1, spec1, w2, spec2, wmin=None, wmax=None, spectrum=False, delwav=False, chatter=0*)

Accurately determine relative wavelength/pixel shift between 2 spectra.

Parameters **w1,spec1, w2,spec2** : array-like

wavelength, spectrum pairs

kwargs : dict

- **wmin,wmax**: float

limits to region to use

- **spectrum** : bool

resample 2nd spectra and return second spectrum shifted

- **delwav** : bool

- **chatter** : int

verbosity

Returns **k** : int

shift in pixels. option spectrum *False*, for option delwav *False*

delwav : float

shift in angstroms. For option spectrum *False*, option delwav *True*

k, (w1,s2) : int, tuple

pixel shift, tuple of wave, flux for second spectrum shifted and resampled on wavelength
first spectrum for option spectrum *True*

Notes

k ~ 1/6 pixel [option: resample 2nd spectra]

sum_Extimage()

`uvotgetspec.sum_Extimage(pha_file_list, sum_file_name='extracted_image_sum.fit', mode='create',
ankerlist=None, plotimage=True, correlate=True, corre-
late_wavewindow=[None, None], figno=20, shiftlist=[], clobber=False,
chatter=1)`

This routine will create/update/read a summed extracted image.

Parameters **pha_file_list** : list

list of PHA filenames written by calls of *getSpec*

kwargs : dict

- **sum_file_name** : str

file name for sum

- **mode** : str, {'create','read'}

when 'create' make the sum file; when 'read' read the sum file

- **ankerlist** : list, optional

list of anchor positions

- **plotimage** : bool, optional
make a plot of the image
- **correlate** : bool, optional
try to determine shifts by correlating the image
- **correlate_wavewindow** : list
when correlate *True* then use only the part of the spectrum within [wavemin, wavemax]
- **figno** : int, optional
figure number to use
- **shiftlist** : list, optional
list of shifts to apply
- **clobber** : bool
write over existing file
- **chatter** : int
verbosity

Returns When ‘option=read’ the following are returned: :

- **sumimg** : 2D array
summed image
- **expmap** : 2D array
exposure map for each pixel in summed image
- **exposure** : float
exposure time (maximum)
- **wheelpos** : int
grism wheel position
- **C_1, C_2** : list
dispersion coefficients
- **dist12** : float
distance in pixels between the first and second order anchors
- **anker** : list
anchor position in summed image
- **coefficients** : tuple
(coef0,coef1,coef2,coef3,sig0coef,sig1coef,sig2coef,sig3coef) curvature and sigma coefficients for the summed image
- **hdr** : fits header

Notes

The anchor point, by default, will be at point [100,500]

mode = 'create' <make new sum file>, 'read' <read sum file>

The anchor position in the pha_file will need to be passed via ankerlist or be given as keyword ANKXIMG, ANKYIMG in the header of the PHA file (it is).

when correlate_wavewindow = [none,none] nothing is done = [2300,4000] wavelength range where to do cross correlation on flux to generate corrections to ankg

shiftlist = [None, 0, -2, None] can be used to force the shifts (in pix) of the given number in the list of spectra (here assumed to be four. List length must equal pha_file_list length.

Example:

```
phafiles = ['sw00032150003ugu_1_pha.fits', 'sw00032150003ugu_2_pha.fits',
'sw00032150003ugu_3_pha.fits', 'sw00032150003ugu_4_pha.fits', 'sw00032150003ugu_5_pha.fits',
'sw00032150003ugu_6_pha.fits', 'sw00032150003ugu_7_pha.fits', 'sw00032150003ugu_8_pha.fits',
'sw00032150003ugu_9_pha.fits', 'sw00032150003ugu_10_pha.fits', 'sw00032150003ugu_11_pha.fits',
'sw00032150003ugu_12_pha.fits', 'sw00032150003ugu_13_pha.fits']
```

```
uvotgetspec.sumimage( phafiles, mode='create', chatter=1, clobber=True)
```

Paul Kuin 2011 (MSSL/UCL)

Deprecated The summing of spectra has been moved to the uvotspec module; this version is already superseded.

sum_PHAspectra ()

```
uvotgetspec.sum_PHAspectra (phafiles, wave_shifts=[], exclude_wave=[], ignore_flags=True,
use_flags=['bad'], interactive=True, outfile=None, returnout=False,
figno=[14], ylim=[-2e-15, 5e-13], chatter=1, clobber=True)
```

Read a list of phafiles. Sum the spectra after applying optional wave_shifts. The sum is weighted by the errors.

Parameters phafiles : list

list of filenames

wave_shifts : list

list of shifts to add to the wavelength scale; same length as phafiles

exclude_wave : list

list of lists of exclude regions; same length as pha files; one list per file for an individual file the the list element is like [[1600,1900],[2700,2750],]

ignore_flags : bool

do not automatically convert flagged sections of spectrum to exclude_wave regions

use_flags : list

list of flags (except - 'good') to exclude. Valid keyword values for the flags are defined in quality_flags(),

interactive : bool

if False, the program will only use the given wave_shifts, and exclude_regions

outfile : str

name for output file. If “None” then write to ‘sumpha.txt’

ylim : list

force limits of Y-axis figure

figno : int, or list

numbers for figures or (if only one) the start number of figures

Returns debug information when ‘outfile=None’. :

Notes

Two figures are shown, one with flux for all spectra after shifts, one with broad sum of counts in a region which includes the spectrum, unscaled, not even by exposure.

**** fails quietly for interactive=T, ignore_flags=F, exclude_wave=[], wave_shifts=[0,0,..], use interactive=F**

**** not yet implemented: selection on flags using use-flags**

uvotio

Major functions

fileinfo()

`uvotio.fileinfo(filestub, ext, lfilt1=None, directory='.', chatter=0, wheelpos=None, twait=40.0)`
 finds files for spectrum, matching attitude and lenticular images uncompresses gzipped files if found

Parameters filestub : str

the base of the file name, with the Swift project convention, consisting of “sw” + the OBSID, i.e., “sw00032301001”

ext : int

the number of the extension of the grism file to match

kwargs : dict

- **lfilt1** : str, optional

name of first lenticular filter used. Must be one of ‘uvw2’, ‘uvm2’, ‘uvw1’, ‘u’, ‘b’, ‘v’, ‘wh’

- **directory** : path, str

path for directory. This is the directory with the grism file.

- **chatter** : int

verbosity

- **twait** : float

The maximum time allowed between begin and end of matched exposures of grism-lenticular filter, for each match.

- **wheelpos** : int

If given, use to discriminate between UV and Visual grisms.

Returns specfile, attfile: str :

filename of spectrum, the attitude file name.

lfilt1, lfilt2 : str

lenticular filter file name before the grism exposure or None, and the file name of the lenticular filter following the grism

lfilt1_ext, lfilt2_ext : int

extension number for each lenticular filter matching exposure

rate2flux()

`uvotio.rate2flux(wave, rate, wheelpos, bkgrate=None, pixno=None, co_sprate=None, co_bgrate=None, arf1=None, arf2=None, effarea1=None, effarea2=None, spectralorder=1, anker=None, test=None, respfunc=False, swifttime=None, option=1, fudgespec=1.0, frametime=0.0110329, debug=False, chatter=1)`

Convert net count rate to flux

WARNING: dependent on the parameters passed, the old CALDB (<=2012) OR the new flux calibration will be used. Since 10SEP2012 the coi-factor is included in the calculation of the flux and the effective area. A coi-correction is still made when using the old CALDB which will be inconsistent to that calculated in the writeSpectrum() which makes the output file.

many of the parameters are needed to calculate the coi-factor

Parameters wave : float ndarray

wavelength in A

rate, bkrate : float ndarray

net and background count rate/bin in spectrum, aperture corrected

co_sprate, co_bgrate : ndarray

total of spectrum+background and background rate/bin for the coincidence area of constant width (default set to 16 pixels)

wheelpos : int

filter wheel position

pixno : ndarray

pixel coordinate (zero = anchor; + increasing wavelengths)

co_sprate, cp_bgrate : ndarray

rates for calculating the coincidence loss

arf1, arf2 : path or "CALDB", optional

effarea1, effarea2 : FITS HDU[, interpolating function]

result from a previous call to readFluxCalFile() for first or second order

spectralorder : int

the spectral order of the spectrum, usually =1

trackwidth : float

width of the spectral extraction used in units of sigma

anchor : list

anchor detector coordinate positions (pix) as a 2-element numpy array

frametime : float

the frame time for the image is required for the coi-correction

swifttime : int

swift time of observation in seconds for calculating the sensitivity loss

debug : bool

for development

chatter : int

verbosity (0..5)

respfunc : bool

return the response function (used by writeSpectrum())

Returns (flux, wave, coi_valid) : tuple

coi-corrected flux type interp1d, array wave, and matching boolean array for points not too bright for coincidence loss correction

Notes

2013-05-05 NPMKuIn - adding support for new flux calibration files; new kwarg 2014-02-28 fixed. applying fnorm now to get specrespfunc, pass earlier effective area 2014-04-30 NPMK changed coi_func parameters (option=1,fudgespec=1.322,frametime,coi_length=29)

readFluxCalFile()

`uvotio.readFluxCalFile(wheelpos, anchor=None, option='default', spectralorder=1, arf=None, chatter=0)`

Read the new flux calibration file, or return None.

Parameters wheelpos : int, required

the position of the filterwheel

kwargs: dict :

- **anchor** : list, optional coordinate of the anchor
- **option** : str option for output selection:
 - option=="default" + anchor==None: old flux calibration option=="default" + anchor : nearest flux calibration + model extrapolation option=="nearest" : return nearest flux calibration option=="model" : model
- **spectralorder** [int] spectral order (1, or 2)
- **arf: path** fully qualified path to a selected response file

Returns None if not (yet) supported :

option == ‘model’ returns the (astropy/pyfits) fits HDU (header+data) of the model :

option == ‘nearest’ :

returns the fits HDU of the nearest calfile

option == ‘default’ and anchor == None: :

returns the fits HDU of the nearest calfile

option == ‘default’ and anchor position given (in detector coordinates) :

returns the fits HDU and an interpolating function `fnorm(wave in A)` for the flux correction

Notes

2013-05-05 NPMKuin

sensitivityCorrection()

`uvotio.sensitivityCorrection(swifftime, sens_rate=0.01)`

give the sensitivity correction factor to divide the rate/flux by

Parameters `swifftime` : float

time of observation since 2005-01-01 00:00:00 in seconds, usually TSTART

`sens_rate` : float

the yearly percentage loss in sensitivity

Notes

A 1%/year decay rate since 2005-01-01 has been assumed and the length of the mean Gregorian year was used

writeSpectrum()

`uvotio.writeSpectrum(ra, dec, filestub, ext, Y, fileoutstub=None, arf1=None, arf2=None, fit_second=True, write_rmffile=True, used_lenticular=True, fileversion=2, calibration_mode=True, history=None, chatter=1, clobber=False)`

Write a standard UVOT output file - Curved extraction only, not optimal extraction.

Parameters `ra,dec` : float, float

position in decimal degrees

`filestub` : str

“sw” + obsid

`ext` : int

extension number

`Y` : tuple

compound variable with spectral data from `uvotgetspect`

Returns Writes the output file only. :

Notes

Output file composition

For details, see the output file format description.

wheelpos, filter, orders, author

writeEffAreaFile()

```
uvotio.writeEffAreaFile(wheelpos, spectralorder, wave, specresp, specresp_err=None,
                        anker=None, dxy_anker=None, fileversion='999', todir='.', rebin=True,
                        clobber=False)
```

create an ARF file

Parameters wheelpos : int, {160,200,955,1000}

spectralorder: int, {1,2} :

wave: ndarray :

wavelengths in Angstrom

specresp: ndarray :

effective area (EA) in cm² for each wave

specresp_err: ndarray :

1-sigma EA error (random + systematic)

anker: list, ndarray[2] :

2-element array with position in det coordinates of EA

dxy_anker: list, ndarray[2] :

EA determined for box [anker[0]+/-dxy_anker[0], anker[1]+/-dxy_anker[1]]

fileversion: str :

version for this EA (spectral response) file.

todir: path :

directory to place the file into

rebin : bool

When true (old behaviour) bin 1 A in wavelength When False, make one bin for each point in array wave.

Returns the new effective area file with file name something like :

'swugu0160_ax1100ay1100_dx150dy150_o1_20041120v001.arf'

Notes

- Modified 15-SEP-2012 by Paul Kuin.

With only wheelpos, spectralorder, wave, specresp input, the output file conforms to the HEASARC approved response file. The additional keywords and error column have not been approved as of 15 September 2012.

- Modified 13 Feb 2013 by Paul Kuin

Added futher keyword COIAWARE to discriminate between the old and new effective areas and changed comments after keywords to be more descriptive.

- Modified 5 March 2013 by Paul Kuin

header edited

- Renamed 28 Dec 2013

first extension assumed 1-spaced wavelengths. Relaxed to allow variable wavelengths.

- changed to reflect use of full coi-solution 2014-08-20. Paul Kuin
- added no rebinning as option. It actually will rebin slightly by calculating the minimum value of the bin from the distance of its neighbors, and the maximum value is chosen to have no gaps between bins.

write_rmf_file()

`uvotio.write_rmf_file(rmffilename, wave, wheelpos, disp, anchor=[1000, 1000], effarea1=None, effarea2=None, chatter=1, clobber=False)`

Write the RMF file for the first order spectrum

Parameters **rmffile** : path, str

file name output file

wave [ndarray] wavelengths of the bins

wheelpos [int] filter wheel position

disp [ndarray] dispersion coefficients

- **chatter** : int verbosity
- **clobber** : bool if true overwrite output file if it already exists

Returns Writes the RMF file :

Notes

The line spread function from the uv grism at default position is currently used for all computations. Since the RMF file encodes also the effective area, this version presumes given anchor position.

2014-02-27 code cleaned up. Speed depends on number of points

Key functions

XYSpecResp ()

`uvotio.XYSpecResp(wheelpos=None, spectralorder=1, anker=[1129, 1022], test=None, chatter=0)`

the spectral response based on the position of the anchor of the spectrum. Depends on the grism mode via 'wheelpos' and the spectral order.

Parameters **wheelpos** : int

kwargs : dict

- **spectralorder** [int] order of the spectrum
- **anker** [list] position in detector coordinates (pixels)
- **test** [any] if not None then get the response at the boresight

Returns An interpolating function for the spectral response :
 based on the position (Xank,Yank) of the anchor of the spectrum. :
 Depends on the grism mode via ‘wheelpos’ and the spectral order. :

Notes

Will be superseded by *readFluxCalFile*

getZmxFlux ()

`uvotio.getZmxFlux (x, y, model, ip=1)`
 Interpolate model to get normalized flux.

Parameters *x, y* : float

anchor coordinate x,y to find an interpolated solution to the model

model : fits structure

binary table extension (header + data) fields are wave, xpix, ypix, flux

ip : int

The order of the interpolation (1=linear, 2=quadratic, 3=cubic)

Returns flux interpolated at (x,y) in (xpix, ypix) as function of wave :

kev2angstrom ()

`uvotio.kev2angstrom (E, unit='keV')`
 conversion of units

Returns The photon wavelength in angstroms :

angstrom2kev ()

`uvotio.kev2angstrom (E, unit='keV')`
 conversion of units

Returns The photon wavelength in angstroms :

updateResponseMatrix ()

`uvotio.updateResponseMatrix (rmffile, C_1, clobber=True, lsffile='zemaxlsf', chatter=0)`
 modify the response matrix lineprofiles using the zemax model prediction from zemaxlsf.fit In addition the zemax profile is broadened by the instrumental broadening of 2.7 pixels.

Parameters *rmffile* : path, str

The rmffile is updated by default

C_1: ndarray :

The dispersion C_1 is used to convert pixels to angstroms.

kwargs : dict

- *lsffile* : path

The lsffile is in the \$UVOTPY/calfiles directory

- *clobber* : bool
 overwrite output.
- *chatter* : int
 verbosity

Returns writes RMF file :

Notes

The same algorithm was implemented in the `write_rmf_file()` routine which does not need the input `rmf` file produced by the “Ftool” `rmfgen`.

uvotmisc

Convenience uvot grism fits header functions

`get_curvatureCoef()`

`uvotmisc.get_curvatureCoef(header, order=1)`
retrieve the sigma coefficients from the FITS header

`get_dispersion_from_header()`

`uvotmisc.get_dispersion_from_header(header, order=1)`
retrieve the dispersion coefficients from the FITS header

`get_keyword_from_history()`

`uvotmisc.get_keyword_from_history(hist, key)`
Utility to get the keyword from the history list.

Parameters `hist` : list

`key` : str

Returns value belonging to key or ‘None’. :

Notes

The history records are written while processing `getSpec()` and added to the FITS header of the output file. These can be read from the header by just getting *all* the history records.

`get_sigCoef()`

`uvotmisc.get_sigCoef(header, order=1)`
retrieve the sigma coefficients from the FITS header

time conversions

UT2swift ()

`uvotmisc.UT2swift (year, month, day, hour, minute, second, millisecond, chatter=0)`

Convert time in UT to swift time in seconds.

Parameters `year` : int

e.g., 2012

month : str or int

e.g., 'JAN'

day : int

e.g., 21

hour : int

minute : int

second : int

millisecond : int

Returns `swifttime` : float

in seconds (see Heasarc for more conversions)

swtime2JD ()

`uvotmisc.swtime2JD (TSTART)`

Time converter to JD from swift time

Returns `JD` : float

Julian Date

MJD : float

Modified Julian Date

gregorian : str

normal date and time

outdate : datetime

python datetime object

Notes

example (input TSTART as a string) for 2001-01-01T00:00:00.000

TSTART=0.000 MJD=51910.00000000 JD=2451910.5

reading ascii tables**rdList ()**

`uvotmisc.rdList (file, symb=' ', chatter=0, line1=None, line2=None, skip='#')`

Put data in list: chatter>4 gives detailed output restrict lines in file with line1, line2 skip lines with the skip char in first position

Parameters `file` : str

file name ascii table

symb : str

character used to split out the columns

line1,line2 : int

sub-select records[line1:line2]

chatter : int

Returns table : ndarray

a table of values

Notes

The table must have equal length columns and the same number of fields on each row/record.

Use rdTab to read a table with numerical only data

rdTab ()

uvotmisc.**rdTab** (*file, symb=' ', commentsymb='#', get_comments=False*)

RdTab will read in a table of numerical values provided every record has the same number of fields. Comment lines start by default with a hash mark, but that can be changed by passing another symbol in commentsymb comments in data records are not supported.

Parameters file : str

file name ascii table

symb : str

character used to separate the columns

commentsymb : str

character used in first position of line for comments

get_comments : bool

if True, return comments only

Returns table : ndarray

a table of values

Notes

The table must have equal length columns with only numbers.

Use rdList to read a table with character data

NPMK (MSSL) 2010

other functions

uniq()

`uvotmisc.uniq(list)`
 preserves order

uvotrotvec()

`uvotmisc.uvotrotvec(X, Y, theta)`
 rotate vectors X, Y over angle theta (deg) with origen [0,0]

Parameters X, Y : arrays

coordinates

theta : float

angle in degrees

Returns rx,ry : arrays

rotated coordinates

bilinear()

encircled_energy()

`uvotmisc.encircled_energy(uvotfilter, areapix)`

Compute the encircled energy in a uvotfilter as compared that in the default 5" radius.

Parameters uvotfilter : one of ["wh", "v", "b", "u", "uvw1", "uvm2", "uvw2"]

filer name

areapix : float

constant describing the number of sub-pixels for computing the cps rate

Notes

This applies solely for point sources.

polyfit_with_fixed_points()

`uvotmisc.polyfit_with_fixed_points(n, x, y, xf, yf)`

compute a polynomial fit with fixed points

Parameters n : int

order of polynomial

x,y : array like

data point coordinates

xf,yf : array like

fixed data point coordinates

uvotplot

helper functions

Ellipse ()

`uvotplot.Ellipse ((x, y), (rx, ry), angle=0.0, resolution=200, **kwargs)`

plot an ellipse using an N-sided polygon

Parameters (x,y) : float

centre ellipse

(rx,ry) : float

half axis ellipse

angle : float

angle in units of degrees

resolution : int

determines number of points to use

and additional kwargs for pyplot.plot() :

binplot ()

`uvotplot.binplot (*args, **kwargs)`

Bin up the arrays with the keyword `bin=<number>` Same parameters as used by plot (pyplot)

contourpk ()

`uvotplot.contourpk (x, y, f, levels=None, xb=None, xe=None, yb=None, ye=None, s=60, kx=1, ky=1, dolabels=True, **kwargs)`

Make a contour plot with 1-D array inputs for X, Y and F. This is a wrapper to convert lists of points (X,Y,Z) in 2-D arrays, then calls contour()

Parameters X, Y: ndarrays[:], 1D on a 2D plane :

coordinates X, Y

Z : ndarray[:], 1D function on X,Y

maskEllipse ()

`uvotplot.maskEllipse (masking, x, y, a, b, theta, test=0, chatter=1)`

update a mask excluding ellipse region

Parameters masking : ndarray, 2D, bool

boolean array to apply mask to (i.e., `numpy.ones(array([200,400]),dtype=bool)`)

x,y : int, float

ellipse center coordinate x,y

a,b : float

ellipse major axis a; minor axis b;

theta : float

rotation angle theta counterclockwise in deg.

Returns `masking` with all pixels inside the ellipse are set to `False` :

plot_ellipsoid_regions ()

uvotplot.plot_ellipsoid_regions (*Xim, Yim, Xa, Yb, Thet, b2mag, matched, ondetector, img_pivot, img_pivot_ori, img_size, limitMag, img_angle=0.0, lmap=False, makeplot=True, color='k', annulusmag=13.0, ax=None, chatter=1*)

This routine is to plot ellipsoid regions on the grism image/graph, which may be a rotated, cropped part of the detector image

Returns None or boolean map image, plots an ellipse on the current figure :

waveAccPlot ()

uvotplot.waveAccPlot (*wave_obs, pix_obs, wave_zmx, pix_zmx, disp_coef, obsid=None, acc=None, order=None, wheelpos=200, figureno=1, legloc=[1, 2]*)

Plots of the accuracy of the wavelength solution from zemax compared to the observed wavelengths.

Parameters wave_obs, pix_obs : ndarray

observed wavelengths points (green circles)

wave_zmx, pix_zmx : ndarray

calculated zemax points (or the interpolated solution (red crosses))

disp_coef : ndarray

dispersion coefficients

disp_coef : list

coefficients in reverse order: if p is of length N, this the polynomial

is as follows for coeff named p:

$$y(x) = p[0]*(x**N-1) + p[1]*(x**N-2) + \dots + p[N-2]*x + p[N-1]$$

kwargs : dict

- **acc :** accuracy in wavelength
- **order :** order of polynomial disp_coef (default len(coef))
- **obsid :** if given, append to title

Notes

Figure description

x-axis : pix - pixel number referenced to [260nm in first order]

Top panel only

y-axis: lambda - lambda_linear

linear term in the dispersion a linear term is fit to the wavelengths

$$\lambda_{lin} = coef[0] + coef[1]*pix$$

Bottom panel only

y-axis: residuals

wave_obs, pix_obs - wave(pix_obs) (green circles) wave_zmx, pix_zmx - wave(pix_zmx) (red crosses)

uvotspec

Major functions

adjust_wavelength_manually()

`uvotspec.adjust_wavelength_manually` (*file=None, openfile=None, openplot=None, ylim=[None, None], ions=['HI', 'HeII'], reference_spectrum=None, recalculate=True*)

manually adjust the wavelength scale

Parameters **file** : path

extracted spectral data (i.e., after running `uvotgetspec.getSpec()`)

fileopen : filehandle

opened spectral data file

openplot : axis

axis instance to use

ylim : list(2)

list of length 2 with limits of the Y-axis or None

ions : list

list of ions to use for annotation valid ions are `sdata.keys()`

reference_spectrum : `astropy.table.table.Table`

column 1: wavelength, column 2: flux

recalculate : bool

when set, use wavelength shift to determine shift of `pixno` array and use dispersion to recalculate the wavelengths

Notes

The header will be updated with the value of the wavelength shift The wavelengths in the second extension `lambda` column will be shifted. The response file will need to be recreated separately.

Returns the figure instance

flag_bad_manually()

`uvotspec.flag_bad_manually` (*file=None, openfile=None, openplot=None, ylim=[None, None]*)

manually flag bad parts of the spectrum

Parameters **file** : path

extracted spectral data (i.e., after running `uvotgetspec.getSpec()`)

openfile : filehandle

openplot : `matplotlib.figure`

figure handle

ylim : list

limits y-axis

Notes

returns ax:axes instance, fig:figure instance, [f:fits file handle if passed with openfile]

The data quality flag of flagged pixels will be set to “bad” The header will be updated with the value of the wavelength shift

plot_spectrum()

`uvotspec.plot_spectrum(ax, phafile, errbars=False, errhaze=False, hazecolor='grey', hazel-pha=0.2, flag='all')`

make a quick plot of a pha spectrum

plotquality()

`uvotspec.plotquality(ax, w, quality, flag=['bad'], colors=['c', 'g', 'y', 'm', 'b', 'r', 'k'], alpha=0.2)`

add vertical greyscale regions in plot for each quality flag

Parameters **ax** : matplotlib.axes.Axes instance

w : array

x-axis values

quality : array

quality flags matching x-axis points

flag : list of strings

each list value must be one of the valid keys from `quality_flags()`

colors : array

color values

alpha : float

alpha value for transparency

sum_PHAspectra()

`uvotspec.sum_PHAspectra(phafiles, flag_bad_areas=True, adjust_wavelengths=True, outfile=None, returnout=False, figno=[14], ylim=[-2e-15, 5e-13], chatter=1, clobber=True, wave_shifts=[], exclude_wave=[], ignore_flags=True, use_flags=['bad'])`

Read a list of phafiles. Sum the spectra after applying optional `wave_shifts`. The sum is weighted by the errors.

Parameters **phafiles** : list

list of filenames

flag_bad_areas : bool

interactively select areas of each spectrum not to include in each spectrum

adjust_wavelengths : bool

interactively select a wavelength shift for each spectrum to apply before summing the spectra

outfile : str

name for output file. If “None” then write to ‘sumpha.txt’, if ending in ‘.fit’ a fits file will be written.

ylim : list

force limits of Y-axis figure

figno : int, or list

numbers for figures or (if only one) the start number of figures

wave_shifts : list

list of shifts to add to the wavelength scale; same length as phafiles

exclude_wave : list

list of lists of exclude regions; same length as pha files; one list per file for an individual file the the list element is like [[1600,1900],[2700,2750],]

ignore_flags : bool

do not automatically convert flagged sections of spectrum to exclude_wave regions

use_flags : list

list of flags (except - ‘good’) to exclude. Valid keyword values for the flags are defined in quality_flags(),

Returns debug information when ‘outfile=None’. :

Notes

Two figures are shown, one with flux for all spectra after shifts, one with broad sum of counts in a region which includes the spectrum, unscaled, not even by exposure.

** not yet implemented: selection on flags using use-flags ** smooth each spectrum, correlate for shift [all quality=‘good’ data points]

Key functions

apply_shift()

`uvotspec.apply_shift(file, delwav, recalculate=False)`

apply a given wavelength shift in A

check_flag()

`uvotspec.check_flag(quality, flag, chatter=0)`

return a logical array where the elements are True if flag is set

complement_of_ranges()

`uvotspec.complement_of_ranges(ranges, rangestart=0, rangeend=None)`

given a list of exclusion ranges, compute the complement

plot_line_ids()

`uvotspec.plot_line_ids` (*ax*, *ylower=None*, *ion='HI'*, *color='k'*, *dash=[0.07, 0.1]*)

add the line ids to the plot

Parameters *ax* : plot handle

ylower : float

y-level where the bottom of the line should be

ion : ['HI', 'HeI', 'HeII',]

key to the ion to be plotted

quality_flags_to_ranges()

`uvotspec.quality_flags_to_ranges` (*quality*)

given wavelength and quality flag arrays, reduce the quality to ranges of a certain quality (except for “good” = 0.)

Parameters *wave* : array

x-axis values

quality : array

quality flags matching x-axis points

Returns *quality_ranges* : dict

a dictionary of ranges for each flag except ‘good’

uvotwcs**Major functions****makewcshdr()**

`uvotwcs.makewcshdr` (*filestub*, *ext*, *attfile*, *indir='.'*, *teldef=None*, *wheelpos=None*, *continue_when_graspcorr_fails=True*, *catspec=None*, *uvotgraspcorr_on=True*, *update_pnt=True*, *chatter=1*)

make the header of a lenticular filter for a grism image to use in `uvotapplywcs` writes a file with a bogus image returns the name of the file

Parameters *filestub* : str

identifying part of filename being “sw”+‘obsid’

ext : int

extension of fits file to process

attfile : str

attitude file name. Needs to be set when `update_pnt` set.

indir : str

path, directory of data files

teldef : str

filename *teldef* file for epoch of anchor calibration

wheelpos : int, {160,200,955,1000}

filterwheel position for grism

continue_when_graspcorr_fails : float

this would supply a solution - though quite bad

uvotgraspcorr_on: bool :

if not set, then the original pointing is used after optional update when update_pnt was set

update_pnt : bool

allows updating the header RA_PNT,DEC_PNT,PA_PNT keywords using the attitude file (which is required)

catspec : path

path to catalog spec file other than default

chatter : int

verbosity

Returns creates a fake sky file with appropriate header to run ‘findInputAngle’ :

Notes

need to update the tstart and tstop of the primary header (not a showstopper)

Other functions

`uvotwcs.correct_image_distortion(x, y, header)`

This routine applies to the (x,y) position on the image the grism distortion correction input header must be from the grism image (position) to be corrected

`uvotwcs.get_pointing_from_attfile(tstart, tstop, attfile)`

`uvotwcs.get_distortion_keywords(wheelpos)`

provide the grism header with distortion keywords

- note on the original calibration with the target at the default detector position
- guide to reducing grism data using the original uvotpy ftool This method is deprecated, since the flux calibration does not correct for coincidence.

1.7 Examples of UVOT grism spectra

1.7.1 Swift UVOT Spectra of SN2012AW

Swift followed the development of the SN2012aw spectra. The spectra show the increasing absorption due to metals in the UV over time.

Figure 1.85: The *Swift* UVOT spectra

There are two processes that affect the spectra: effective temperature of the photosphere and ionisation/excitation processes. Dividing out the continuum emission of the blackbody radiation of the photosphere clarifies the development of absorption of the radiation field over time.

More details and analysis are being published:

- the exceptionally long plateaux in the UV emission.

1.7.2 GRB 130427A *Swift* UVOT uv grism spectrum: the deconvolution of smearing from attitude loss

Creation of the kernel of blurring motion

In order to do a deconvolution of the blurred image, the motion of the source over the detector during the exposure must be found. To do that, we use the zeroth order of the GRB spectrum and a zeroth order of a good observation.

A region of 80x60 pixels in the detector image has been selected, centered approximately on the peak zeroth order emission. The centers were at

```
GRB: 00554620992ugu_dt.img gu388741990I [983, 847]
COM: 00055200001ugu_dt.img gu132535224I [1394, 864]
```

The background image was made in two steps:

```
bkg = GRB
bkg[ GRB > (GRB.mean()+GRB.std()) ] = GRB.mean()
bkg[ bkg > (bkg.mean()+bkg.std()) ] = bkg.mean()
```

replacing the values higher than the mean plus one standard deviation with the mean. The same was done for the COM images. The background was subtracted and the resulting image normalised to the counts in the area visibly above the background as follows:

```
GRB = GRB - bkg
GRB = GRB / (GRB[25:78, 2:41].sum())
COM = COM - combkg
COM = COM / (COM[20:63, 13:41].sum())
```

The zeroth order images were saved in fits files:

```
filter/normalisedGRBzerothOrder.fits
filter/normalisedGoodzerothOrder.fits
```

To obtain the kernel for the blurring, the GRB zeroth order image needs to be deconvolved with the good image of the COM zeroth order. To do that I am using the “LUCY” program from the “STARLINK” software. Since this takes special formatted input files, the fits files need to be converted first using the “FITS2NDF” program.

The “STARLINK” was grabbed from the GIT repository, built and installed.

The files were converted to the “STARLINK” NDF format using:

```
convert

cp normalisedGRBzerothOrder.fits infile
fits2ndf infile outfile fmcnv=T
mv outfile.sdf grb.sdf

cp normalisedGoodzerothOrder.fits infile
fits2ndf infile outfile fmcnv=T
mv outfile.sdf good.sdf
```

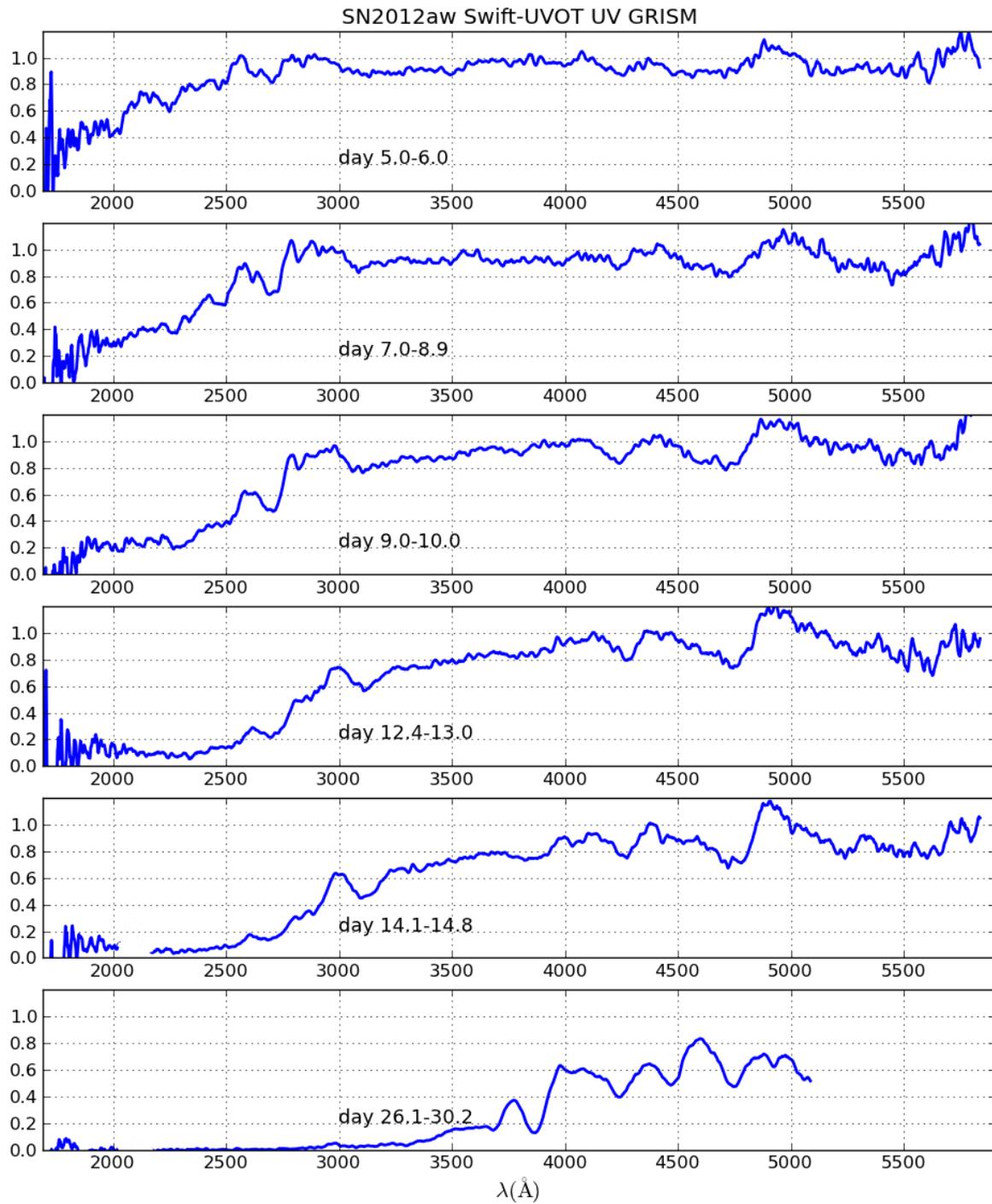


Figure 1.86: *Swift* UVOT spectra normalised by the continuum of the Blackbody photospheric emission.

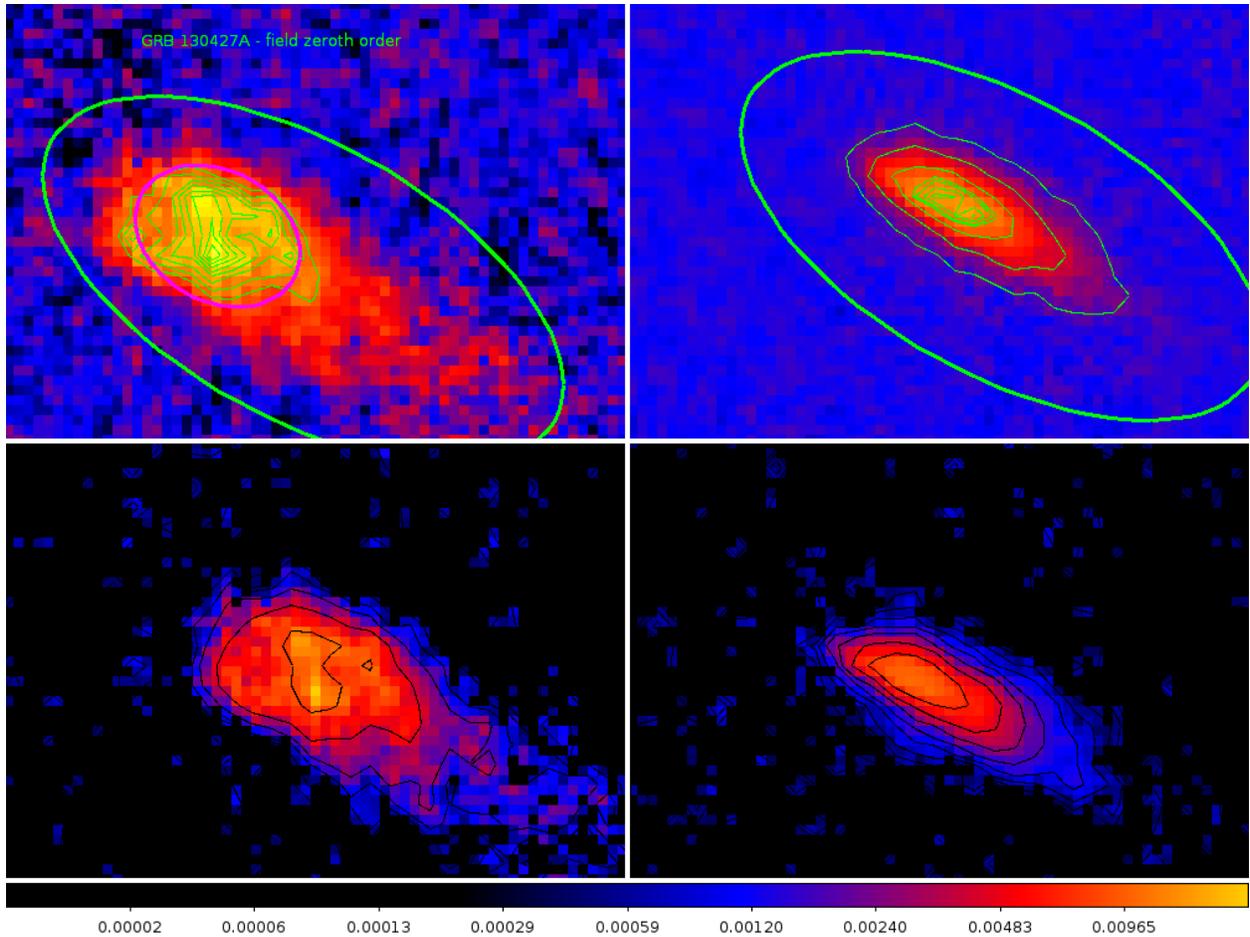


Figure 1.87: Figure 1. The top row are the zeroth order images of GRB130427A (left) and a zeroth order in the image of obsid 00055200001[2] (COM). The bottom row shows the images after background subtraction and normalisation.

```
kappa
lucy grb.sdf good.sdf first aim=0.0001 niter=60
ndf2fits first.sdf first.fit
lucy grb.sdf good.sdf second aim=0.0001 niter=60 start=first.sdf
ndf2fits second.sdf second.fit
```

Then the first result was used to make an image and the second to build a contour map. The statistics were compared using the “stats” command and showed sufficient convergence.

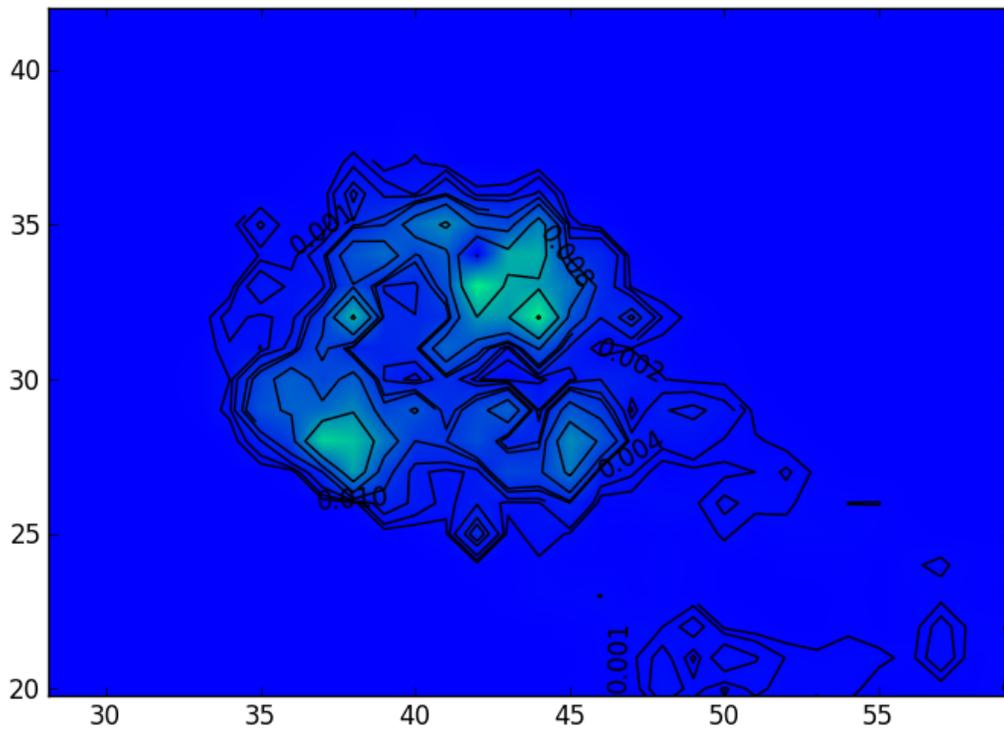


Figure 1.88: Figure 2. The resulting filter after applying the Richardson-Lucy method.

The distance between the main peaks is slightly over 7 pixels. Presumably this shows the positions that the spectrum was at over time. Obviously this would mean that the observation was taken whilst in mainly two locations. The orientation of the motion between these locations is consistent with that found from the analysis of the motion in the event data. The direction of the motion is mainly across the dispersion direction,

Correcting the grism image

Finally, the grism observation was read into Python, and the image was written to a new fits file (primary), grism.fits, followed by a conversion to NDF:

```
[paul-mssl21:uvot/image/filter] kuin% fstruct grism.fits
No. Type      EXTNAME      BITPIX Dimensions(columns)  PCOUNT  GCOUNT
    0 PRIMARY          -32      1987 2046              0      1
[paul-mssl21:uvot/image/filter] kuin% fits2ndf grism.fits grismndf fmcnv=T
```

Finally, the image was processed with the filter, and converted to fits:

```
lucy grismndf.sdf second.sdf newgrism aim=0.1 niter=5
```

PSF area is about 27 by 20 pixels.

X-axis margin is 27 pixels wide.
 Y-axis margin is 20 lines wide.
 Internal file size is 2041 by 2086.

Using a constant noise value of 0.329904.
 Initial normalised chi squared is 0.572227

```
Iteration 1
  Normalised chi squared: 0.296033
```

```
Iteration 5
  Normalised chi squared: 0.252322
```

```
[paul-mssl21:uvot/image/filter] kuin% ndf2fits newgrism.sdf newgrism.fits
```

The new fits observation was read into Python, as well as the grism observation, 'sw00554620992ugu_dt.img'. Then, in the new data array values less or equal to zero were set to 1e-3. Finally, the array was copied in place of the old one, and a new fits file was written. That file was subsequently renamed to the old name for further processing. The image was inspected using DS9 and looks much sharper.

The spectrum was extracted using the white filter images extracted from the event data for positioning. The spectrum may still be off by some pixels due to the movement of the satellite. Probably the kernel adjusted to some mean over the 50 second period, but we should allow for about 7 pixels just for that.

A better spectrum

The white event data had been processed into a file with many individual extensions, each aspect corrected. This file, sw00554620992uwh_sk.img was copied into the directory, after which the default program was run as

```
uvotgetspec.getSpec(ra,dec,obsid,1,clobber=True)
```

The calibrated anchor offset, as derived using the last white image extension (22), from the spectrum was 6 pix.

The

Spectral line identifications:

wave	identification @z=0.34
1260	Si II 2P - 2D ?
1262	C I 3P - 3P
1277	C I 3P - 3D
1280	C I 3P - 3P
1302	O I 3P - 3S
Continued on next page	

Table 1.1 – continued from previous page

wave	identification @z=0.34
1305	O I 3P - 3S
1330	C I 3P - 3P
1335	C II 2P - 2D
1368	Al I 2P - 2S
1394	Si IV 2S - 2P
1402	Si IV 2S - 2P
1450	unid
1526	Si II 2P - 2S
1548	C IV 2S - 2P
1551	C IV 2S - 2P
1640	He II
1660	C I 3P - 3P
1908	C III 1S - 3P
1964	unid
2139	N II 3P - 5S
2143	N II 3P - 5S
2177	Ni II* 4F - 4F
2185	Ni II* 4F - 4F
2206	Si III ?
2334	Fe II a6D - z6P uv3
2344	Fe II a6D - z6P uv3
2349	Fe II a6D - z6P uv3
2374	Fe II a6D - z6F uv2
2383	Fe II a6D - z6F uv2
2387	Fe II a6D - z6F uv2
2389	Fe II a6D - z6F uv2
2396	Fe II a6D - z6F uv2
2405	Fe II a6D - z6F uv2
2505	unid C II 2512?
2587	Fe II a6D - z6D uv1
2600	Fe II a6D - z6D uv1
2632	Fe II a6D - z6D uv1
2715	Fe II a4D - z4D uv63
2740	Fe II a4D - z4D uv63
2748	Fe II a4D - z4D uv63
2770	Fe II a4D - z4D uv63
2796	Mg II 2S - 2P
2803	Mg II 2S - 2P
2832	unid
2852	Mg I 1S - 1P
2929	Mg II 2P - 2S

The dispersion errors are not random, but increase towards the short and the long wavelengths, being fixed at 2600Å. Checking the calibration spectra (WR86 00057020001[1] was taken with her anchor within ~20pixels on the detector of that of the GRB spectrum, and the WR52 spectrum taken at the nearby boresight) with the wavelengths from the calibration shows that the assigned wavelength scale is ~14Å too short at 1700Å, and ~25Å too short at 5800Å, but there is no clear offset in the 2200Å-3300Å region. This is perhaps due to the adoption of a single scale factor to the zemax model dispersion.

The CIV and SiIV absorptin lines appear both *blue* shifted by about 10Å in the rest frame, which would imply a velocity of ~2500km/s relative to the lines from the neutral and singly ionised spectra lines.

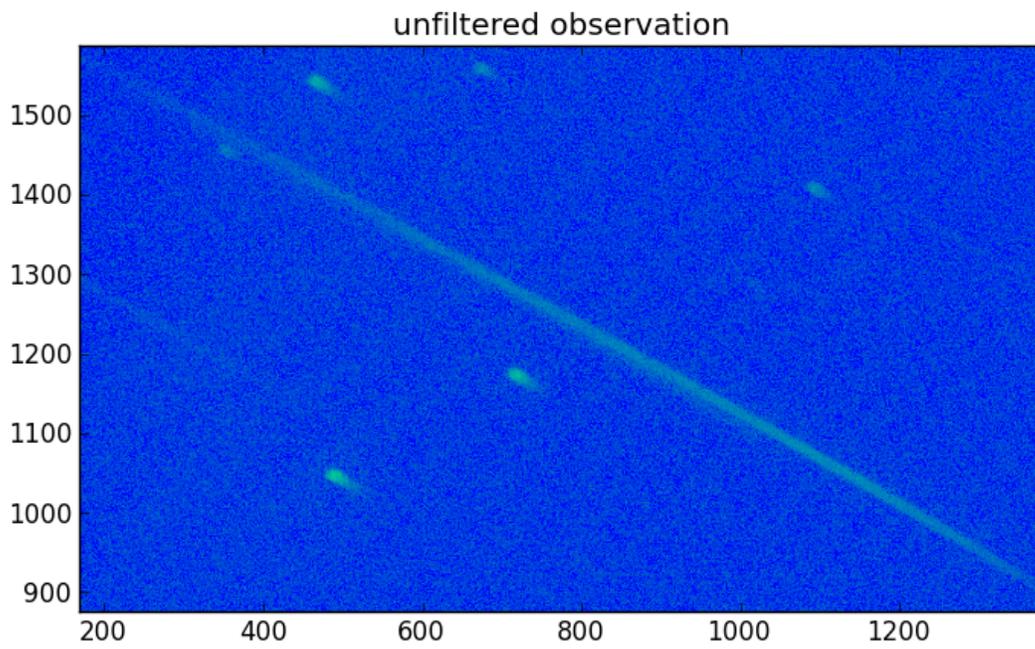


Figure 1.89: Figure 3. The unfiltered image (section).

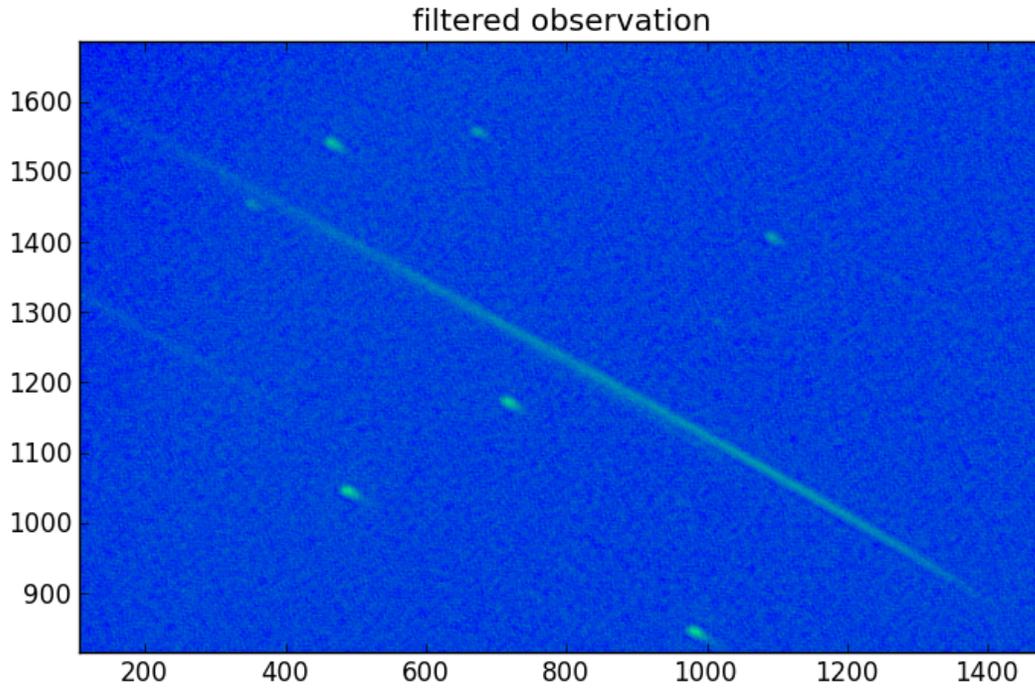


Figure 1.90: Figure 4. The filtered image (section).

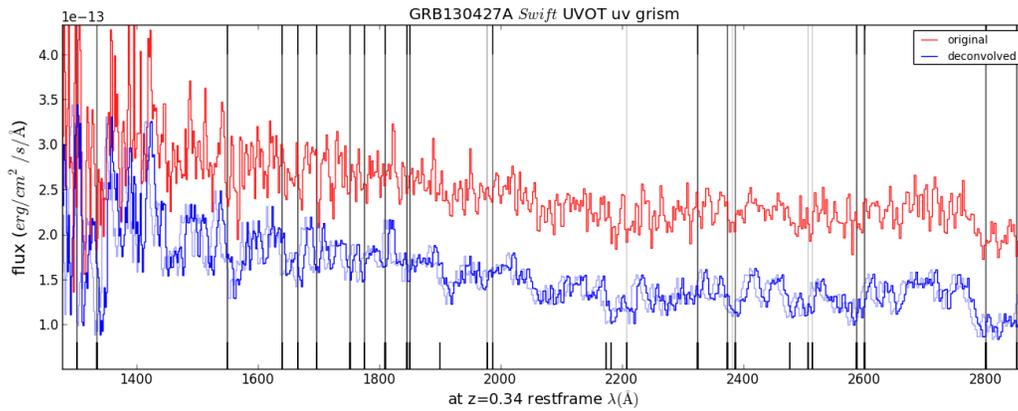


Figure 1.91: Figure 5. Comparison of the unfiltered tot the deconvolved spectrum. An idea of the uncertainty in wavelength scale can be gotten because the deconvolved spectrum was plotted with a 15Å offset also.

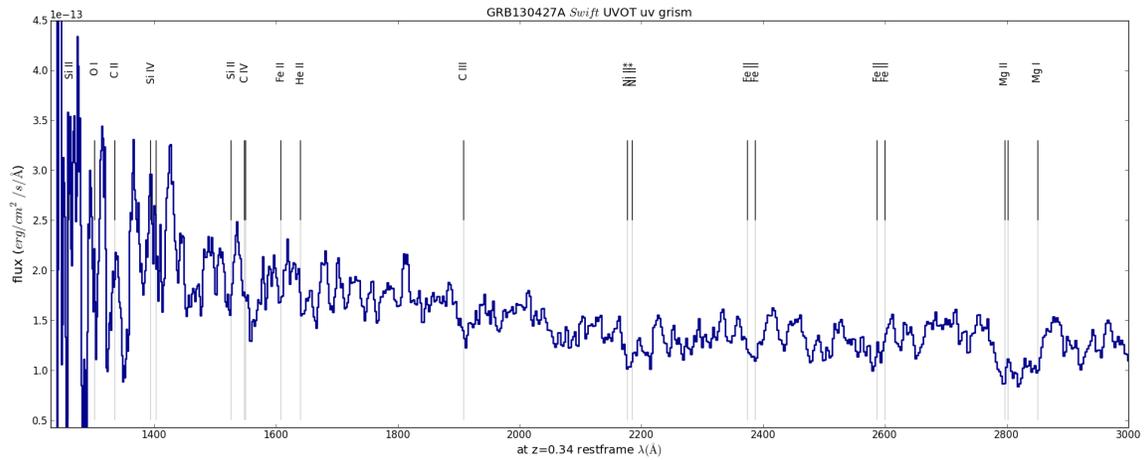


Figure 1.92: Figure 6. The deconvolved spectrum with line identifications.

1.8 Indices and tables

- *genindex*
- *modindex*
- *search*