The SWIFT UVOT Software Guide

Version 2.2
March 2008

S. Immler, M. Still, P. Boyd, S. Holland, F. Marshall, W. Landsman, G. Rohrbach, M. Tripicco, R. Wiegand

NASA/GSFC — Swift Science Center

swifthelp@athena.gsfc.nasa.gov

Contents

Document History

Table 1: Document history.

Date	Version	Author	Comment
2004-Nov-19	1.0	Martin Still	Swift 1.0 pre-launch version. Software has not been tested on flight data. Recipes use data based on simulated telemetry.
2005-Mar-10	1.1	Martin Still	Swift 1.3 release version after feedback from the Science Team.
2005-Aug-11	1.2	Martin Still	Update for HEAsoft 6.0.2 after feedback from community. New tools added.
2006-Nov-18	2.0	Stefan Immler	Update for HEAsoft 6.1 and new tools. Converted from Word->Wiki->LaTeX->HTML
2006-Jun-18	2.1	Stefan Immler	Update for HEAsoft 6.2 Changes for UVOTSOURCE, UVOTMAGHIST, New tools added: UVOTAPERCORR and UVOTCOINCIDENCE
2008-March-3	2.2	Stefan Immler	Update for HEAsoft 6.3 and 6.4 and new tools

Chapter 1

Introduction

1.1 Structure of this document

This document describes the software, both developed and adopted, that we recommend for the reduction and analysis of UVOT data. In Chapter 1 we briefly describe the software requirements for your computer, the data formats and the structure of the Swift archive, although note that these items are all discussed in more detail in other documents.

The next two chapters describe the software from differing perspectives. Chapter 2 provides a set of ABC recipes that can be followed in sequence. The end result is to produce science quality products from raw spacecraft data. These recipes are far from exhaustive. Their aim is to provide a rough guide to UVOT data analysis, and should be used as the spring board for users who want to become familiar with software.

Chapters 3 and 4 are a more detailed look at the UVOT tools developed for this mission. It is envisioned that these sections will used for reference long after the user stops following the ABC recipes of Chapter 2 rigorously. They detail the full functionality and I/O of each tool.

An important note to make is that the philosophy of UVOT software development has been a conservative one. The goal has been convert the raw data into a form readily acceptable by standard data analysis tools as quickly as possible. We use these standard tools liberally in our examples from Chapter 2 but do not reproduce the documentation for these tools here. Instead URL links are provided to existing web material, wherever appropriate.

Please forward comments, suggestions and bug reports for this document and the UVOT software to the Swift Science Center at swifthelp@athena.gsfc.nasa.gov.

1.2 Software requirements

The software suite HEASoft is required to follow the steps in this guide. It is recommended that users download this package and keep it up-to-date since it is updated on timescales of weeks. HEASoft contains general and mission-specific tools for past and current high-energy astrophysics space mission. Many of the generic tools for file viewing, imaging, photometry and spectroscopy will be used in our recipes. Download instructions can be found at http://heasarc.gsfc.nasa.gov/docs/software/lheasoft. The HEASoft package also contains Swift-specific tools, including the UVOT software described in Chapters 3 and 4.

One further package, the image viewer ds9 (http://hea-www.harvard.edu/RD/ds9) will be required to follow the recipes precisely.

1.3 Data and science modes

All data files conform to the current OGIP (Office of Guest Investigator Programs) FITS (Flexible Image Transport System) standards and conventions. Details of these conventions can be found at:

http://heasarc.gsfc.nasa.gov/docs/heasarc/ofwg/ofwg_intro.html.

The file headers are often commented, both to give a brief description of each keyword, and, occasionally, slightly more detail is provided in COMMENT entries. The processing history of these files are recorded by the HISTORY keywords. Exposure data can be fully described using five criteria, which we summarize here.

1.3.1 Data mode

There are three basic data modes: Image, Event and Image&Event. Image mode provides a two-dimensional sky map with a start and stop time for the exposure, but no arrival times for individual events. Event mode provides position and time information for each recorded event. Image&Event mode provides both types of data from the same exposure. The two windows need not have the same size, location or binning. Data mode is the most basic criteria into which data is separated within the HEASARC archive. There are separate files for Image and Event data. The Image&Event mode data are split into their constituent parts and integrated into the separate Image and Event Level I products. Science users never see this process.

1.3.2 Filter

There are 11 slots in the filter wheel in front of the UVOT camera. The full list of filters and their character codes, used within the file name scheme is provided in Table 1.3.1. Filter type provides the secondary criteria with which to separate data in the archive. There will be one Level I file for every filter used for Image mode exposures during the observing sequence, and one file for every filter used during Event mode observations in the sequence.

Table 1.3.1: UVOT filter character codes.

Filter keyword[1]	Filter code[2]
U	uu
B	bb
V	vv
UVW1	w1
UVM2	m2
UVW2	w2
WHITE	wh
VGRISM	gv
UGRISM	gu
MAGNIFIER	mg
BLOCKED	bl
UNKNOWN[3]	un

[1]From the header keywords within the file.
[2]From the filename.
[3]The UNKNOWN filter type occurs when the filter wheel has got lost.

1.3.3 On-chip binning

UVOT images can be binned in four different configurations, 1x1, 2x2, 4x4 or 8x8, although only the GeNi image, which is transmitted through the TDRSS/GCN system, is currently binned 2x2. Event mode data will always be unbinned (1x1).

1.3.4 Window size

The size of the science window, in raw pixels, can change at any time during the observing sequence.

1.3.5 Window location

The science window location can change at any point during the observing sequence.

1.4 Data archive and completeness

Publicly available Swift data can be downloaded from two separate locations. The main facility will be the HEASARC:

http://heasarc.gsfc.nasa.gov/docs/corp/data.html

An observing sequence will populate the archive once it has been fully telemetered to the ground and reprocessed at the SDC (Swift Data Center). For a brief period after a burst (∼1 week), preliminary data will be available at the SDC repository:

http://swift.gsfc.nasa.gov/sdc

This is, of course, to meet the requirement that Swift data be publicly available as soon as possible. However, when using the repository, be aware that the entire observing sequence may not yet have been telemetered. Early versions of these files may differ from those that eventually populate the HEASARC archive at a later date.

The archive is populated by three separate levels of data product. Table 1.4.1 describes the distinction. Note that there is no external way to tell which product belongs to which level, but we provide some pointers in Sec 1.5.

Table 1.4.1: Data levels contained in the HEASARC archive

Level	Description
I	Raw data. Generally the scientist need start their analysis from Level I products only if there has been improvements in the telescope calibration since the most recent data reprocessing, or if non-standard data screening is intended. Images are stored in raw detector coordinates. Event data are unscreened and recorded in raw detector coordinates.
II	Reduced data. It is envisaged that most scientists will start their analysis from the Level II products. The UVOT reduction pipeline has been performed on images and event tables. Images are stored in sky coordinates (RA2000, Dec2000), although grism data is also stored in corrected detector coordinates. Exposure maps for each individual image are available. Columns containing corrected detector and sky coordinates have been populated. Event data has been screened for standard bad times, e.g., SAA passage, Earth limb avoidance.
III	High-level data products. Level III products are intended to be quick-and-dirty representations of many of the products that the scientist would routinely create during their analysis, e.g., time-averaged images, light curves, spectra etc. Since they are created non-interactively, they are not intended as publishable products but as useful guides to the scientist. The number and type of Level III products in the archive will be determined by the confidence in any non-interactive burst detection and/or the time since the burst. More Level III products will populate the archive at an early stage if the burst has been identified during standard pipeline processing. Products will take longer to arrive if interactive analysis is required to identify the burst.

1.5 Filenames and archive structure

The UVOT instrument provides a unique set of logistical problems for file formatting. The standard observing sequences for bursts are complex, yielding a large number of exposures, with a wide variety of science modes. Formatting schemes also require the flexibility to cope with alterations in the observing sequences uploaded after the start of the sequence and alterations in the science windows during an exposure. A further complication is provided by Swift telemetry constraints. Data is telemetered down in non-sequential order and sequences are not being completely telemetered down until days after the burst. The archiving scheme minimizes the number of data files and organizes the exposures sequentially. The caveat is that many files will contain image or event data with a variety of science windows (varying in position, size, binning and exposure time). In terms of obtaining accurate images and exposure times from the data, it is crucial that the scientist is aware of this issue and analyzes their data accordingly. Table 1.5.1 lists some of the common data files and their location. These can be opened and inspected by, e.g., the FITS viewer fv, which comes with the obligatory HEASoft software distribution. Other instrument engineering and housekeeping data are stored in the /uvot/hk folder, which are not discussed here.

Table 1.5.1: Selection of UVOT archive files. The characters in square brackets flags to which data level each product belongs.

Filename	Description
Directory: uvot/image	
sw00000001001uuu_rw.img	U images in raw units (pixels) [I]
sw00000001001ubb_rw.img	B images in raw units (pixels) [I]
sw00000001001uvv_rw.img	V images in raw units (pixels) [I]
sw00000001001uvv_dt.img	V images in detector units (mm) [II]
sw00000001001uvv_sk.img	V images in sky units (RA, Dec) [II]
sw00000001001uvv_ex.img	V images exposure maps [II]
Directory: uvot/event	
sw00000001001uvvpo_uf.evt	Unscreened V filter event tables [I]
sw00000001001uvvpo_cl.evt	Screened V filter event tables [II]
Directory: uvot/product	
sw00000001001u_sk.img	Co-added sky images (all filters) [III]
sw00000001001u_ex.img	Co-added exposure maps [III]
sw00000001001uvvskim.gif	V filter co-added gif sky image [III]
	GRB light curve derived from V Event data [III]
sw00000001001u.his	Magnitude history derived from Image data [III]
sw00000001001u.cat	Catalogue of detected sources, arranged by filter [III]
Directory: uvot/auxil	
sw00000001001sat.fits	Attitude history file [I]
sw00000001001sao.fits	Orbit and attitude filter data [I]

1.6 Exposure Time Keywords

UVOT science data files include a number of special keywords that are used to calculate the total exposure time. These keywords contain information about the CCD read-out time, as well as the amount of data lost due to several possible effects. The keywords, and their role in determining the total exposure time, are described below.

This is how TIME keywords for UVOT raw images are defined as of the UVIT2FITS telemetry conversion software version V3.15+:

TSTART = The actual start time (from the exposure report packet in the telemetry)

TSTOP = The actual end time (from the exposure report packet in the telemetry)

TELAPSE = TSTOP - TSTART

FRAMTIME = The calculated frame exposure interval which is a function of the UVOT detector hardware window size. This is usually the full 2048 by 2048 but can be smaller for exposures in the White filter, to reduce telemetry volume.

BLOCLOSS = The time lost if the exposure began with the BLOCKED filter in place but was later commanded to a different filter during the exposure. Effectively a dead time. Could be in error by 0-10+ sec due to the timescale on which filter position is available in the housekeeping data. This keyword is present only if the value is nonzero, and should only rarely be present. It is nonzero for some grism observations of RS Oph.

TOSSLOSS = Time lost when the onboard shift-and-add algorithm, which corrects image positions for satellite drift, tosses event data completely off the image. Effectively a dead time. Loss of Aspect Following packets in the housekeeping stream can cause this estimated value to be incorrect.

STALLOSS = Time lost when the DPU stalls because the count rate is too high. Estimated from ((TELAPSE/FRAMTIME) - "EXP report total frames") times FRAMTIME. Set to zero if

less than 1 sec.

ONTIME = TELAPSE - TOSSLOSS - STALLOSS - BLOCLOSS (as of UVOT2FITS V3.16)

BLOCLOSS was accounted for in the DEADC value for UVOT2FITS v3.15. It was removed from the DEADC calculation and treated like all other time losses as of UVOT2FITS V3.16.

LIVETIME = (ONTIME * calculated dead fraction) - BLOCLOSS for UVOT2FITS v3.15.

LIVETIME = ONTIME * calculated dead fraction as of UVOT2FITS V3.16.

EXPOSURE = LIVETIME

DEADC = EXPOSURE / ONTIME

EXP_UNC = The uncertainty in the EXPOSURE keyword value. This attempts to account for uncertainty sources from the telemetry. The major source of uncertainty in the EXPOSURE keyword comes from lost telemetry during shift-and-add. When onboard shift-and-add is turned on and telemetry loss is indicated, EXP_UNC is 1.0 - ("Found Shift&Add Intervals" / EXPOSURE).

CNTEXP =(GIMAGE/NEVENTS)*NFRAMES*CalcFrame*CalcDEAD as of UVOT2FITS V3.16 when the image is 2048x2048. Parameters are defined by-> In sw*uct.hk, take the columns NEVENTS, NFRAMES, GIMAGE. Get values of CalcFrame, and CalcDEAD for the standard window.

NEVENTS = Number of events handled by DPU during exposure.

NFRAMES = Number of exposure frames handled by DPU during exposure. This figure automatically accounts for STALLLOSS since stalled frames are not counted in this number (we hope).

GIMAGE = Number of events found in the image on the ground. This may be in error if there were lost rows in the telemetry. This number SHOULD be identical to NEVENTS in 2048x2048 images but often is smaller by fractions of a percent. The reason for this is unknown and helps make CNTEXP a fuzzy value. One possible source is fractions of images shifted off by S&T.

CalcFrame = The time for 1 exposure frame based on the hardware window.

CalcDEAD = The 1 frame DEADC fraction which actually is 1.0 - dead fraction and thus is the live fraction (I didn't make this up, it is a standard). This value can only be applied to actual exposed to sky time. For example, TELAPSE*DEADC is meaningless unless TELAPSE = ONTIME.

Chapter 2

UVOT Data Analysis Recipes

This section provides a set of standard software recipes for reducing and analyzing UVOT science data. While not comprehensive or exhaustive, the spirit of these recipes is to:

1. Provide the software user with the ability to repeat these tasks on any of the data populating the HEASARC archive or SDC quicklook area.

2. Gain an understanding of the principles behind the software and data organization.

3. Obtain the confidence to move on to more advanced data analysis using software of the users own choice.

The Recipes can be split into two parts. The first, covered in Sections 2.1 and 2.2, contain mostly software written specifically for the reduction of UVOT software from raw (level I) data to calibrated science-grade (Level II and III) products. The majority of these tools are only ever run in the Swift pipeline before data populates the quicklook area and archive. It is anticipated that the general user need only use these tools in cases where updates to the UVOT caldb make a significant impact on the level II and III products. In these cases the user should follow these recipes in order to create new science products.

The second set of recipes, Sections 2.3-2.13, comprise imaging, temporal and spectral analysis of UVOT level II and III science products. The philosophy of the UVOT pipeline software development has been to supply products that are, for the most part, readily readable by generic analysis software. These recipes reflect that, although they are biased toward the software developed at the HEASARC and contained in the HEASoft package (http://heasarc.gsfc.nasa.gov/docs/software/lheasoft). Since the installation of this package is a prerequisite for the Swifft HEAdas tools, the majority of the tools employed in these scripts are already installed on your machine. The exception is the image viewer ds9 (http://hea-www.harvard.edu/RD/ds9).

Note that while the contents of these scripts are topical for the work of Swift Burst Advocates, there exists a Burst Advocate Cookbook containing recipes specific to the unique tasks of Burst Advocacy.

2.1 Image Mode data reduction

Synopsis

The purpose of this script is to recreate fully calibrated (level II) images from raw (level I) images. This is a necessary process if the UVOT calibration has changed significantly since the pipeline processing of the archived data. This will generally affect data collected in the first few months of

operation, during which the caldb will evolve rapidly, or when data is extracted from the archive after some significant time since it was first stored.

Processing flowchart

Figure 2.1.1 represents the flow of data through the UVOT pipeline image chain. The recipe below follows this chain from generating the bad pixel maps to exposure map calculations. The level III portion of this flow diagram is covered in various individual recipes later in this Chapter.

- The orange boxes in the flow charts represent the execution of individual FTOOLs. Some are specific to UVOT data processing, e.g. UVOTBADPIX, while others are applicable to general Swift processing, e.g., SWIFTXFORM.
- The green boxes represent the input of calibration products into the FTOOL. These products populate the Swift CALDB. The caldb filenames are provided in the recipes and tool descriptions in Chapters 3-5.
- The blue boxes represent the science products. Level I products, light-blue, are input to the tools. Level II and Level III products, medium- and dark-blue respectively, can be either input or output.

Recipe

1. Create quality maps for each image exposure. Quality maps contain a raw array of flags associated with the goodness or badness of each detector pixel and are used to help minimize modulo 8 fixed-pattern noise from the images and build exposure maps.

- Namibia> uvotbadpix infile=00072901259/uvot/image/sw00072901259ubb_rw.img.gz outfile=quality_bb.img badpixlist=caldb compress=y

  The bb in the output file name indicates the internal images were taken with the B filter. It is crucial that you keep track of the filter associated with image file. Repeat this step for all of the raw image files in the uvot/image directory, i.e., those files containing the string rw.img in their names. Ensure that each output file has a unique filename. Each output file will contain a series of images, mostly populated by zeros, except in the cases of cosmetically bad pixels, or pixels damaged by telemetry compression.

2. Reduce modulo 8 fixed-pattern noise in the images. Generally, raw image pixels are smaller than the original detector pixels by up to an integer factor of 8. This is because an on-board algorithm is able to centroid the splash of events that a single photon produces. While this process improves the resolution of the detector, artificial fixed-pattern structure is introduced into the image. The nature of this structure is a bias in the photon position rather than an introduction of bogus counts. To some extent the bias can be removed statistically without any loss in photometric accuracy using the tool uvotmodmap:

- Namibia> uvotmodmap infile=00072901259/uvot/image/sw00072901259ubb_rw.img.gz outfile=modmap_bb.img badpixfile=quality_bb.img mod8prod=no mod8file=none nsig=5 ncell=32 subimage=no

  Repeat this step for all of the raw image files in the uvot/image directory,

Figure 2.1: The UVOT imaging chain. Orange denotes a software tool, green a calibration product, light-blue a Level I science product, medium-blue a Level II product and dark-blue a Level III product.

3. Remove pixel-to-pixel variation in the image due to detector sensitivity using uvotflatfield:

- Namibia> uvotflatfield infile=modmap_bb.img
  outfile=flatfield_bb.img flatfile=caldb

  Repeat this step for all of the modulo 8 corrected image files in your working directory. Note that procedure can also be performed on the original raw data if you do not care to perform mod 8 correction on the images. However it would incorrect to perform the mod 8 correction after the flat field. Users are urged to follow these two steps in the correct order.

4. Convert the images from the raw coordinate system to a tangential projection of the sky:

- Namibia> swiftxform infile=flatfield_bb.img
  outfile=sw00072901259ubb_sk.img to=SKY attfile=misc/sw00072901259sat.fits.gz
  teldeffile=caldb ra=23.3555 dec=-41.8234 method=AREA bitpix=-32

  Repeat this step for every filter taken in image mode. Either the raw, mod 8 corrected, or flat-fielded images can be used as input to this tool, but uvotmodmap and uvotflatfield should not be used on data output from swiftxform.

5. Construct exposure maps for each image.

- Namibia> uvotexpmap infile=sw00072901259ubb_sk.img
  outfile=sw00072901259ubb_ex.img badpixfile=quality_bb.img
  teldeffile=caldb attfile=misc/sw00072901259sat.fits.gz
  method=MEANFOV attdelta=5

  Repeat this step for every file produced by swiftxform.

2.2 Event mode data reduction

Synopsis

The purpose of this script is recreate fully calibrated (level II) event files from raw (level I) files. This is a necessary process if the UVOT calibration has changed significantly since the pipeline processing of the archived data. This will generally affect data collected in the first few months of operation, during which the caldb will evolve rapidly, or when data is extracted from the archive after some significant time since it was first stored. Furthermore a reprocessing would be necessary if the pipeline event screening is deemed too stringent for a specific users needs. For example it is conceivable that raw event mode data come in a compressed format when telemetry volumes are high. In certain cases it is possible for events to lose their time tags and are consequently flagged as bad and rejected by screening. Users interested only in imaging do not care about photon arrival times and have lost a potentially valuable data. These can be reclaimed by performing the event chain on the raw data, as demonstrated in the following receipe.

Processing flowchart

Figure 2.2.1 represents the flow of data through the UVOT pipeline event chain. The recipe below follows this chain from COORDINATOR TO UVOTSCREEN. The level III portion of the chain is covered in individual recipes later in the Chapter. While the filter file constructed by the PREFILTER tool is vital component of the chain, it is generated in the Swift pipeline and archived with the data. It is unlikely that a user will need to re-run PREFILTER.

- The orange boxes in the flow charts represent the execution of individual FTOOLs. Some are specific to UVOT data processing, e.g. UVOTBADPIX, whie others are pre-existing generic FTOOLS, e.g., COORDINATOR.

- The green boxes represent the input of calibration products into the FTOOL. These products populate the Swift CALDB. The caldb filenames are provided in the recipes and tool descriptions in Chapters 3-5.

- The blue boxes represent the science products. Level I products, light-blue, are input to the tools. Level II and Level III products, medium- and dark-blue respectively, can be either input or output. Filenames are provided in.

Recipe

1. Starting from the unscreened event tables in uvot/event, repopulate the DET and SKY table columns. These columns contain the coordinates of each event in physical (mm) units and over a tangential projection of the sky (RA2000 and Dec2000).

- Namibia> coordinator eventfile=00072901259/uvot/event/sw00072901259ubbpo_uf.evt. eventext=EVENTS teldef=caldb attfile=00072901259/auxil/sw00072901259sat.fits.gz aberration=n randomize=y seed=836 ra=23.3576 dec=-41.8234

  This is a generic ftool and not a dedicated UVOT or Swift tool. Therefore we do not document it here. A detailed description of this tool and parameter use can be obtained by typing fhelp coordinator on your command line. Its generally advisable to unzip event files before running the tool. Repeat this task for every raw event table in the uvot/event directory, e.g., those files including the string po_uf.evt. Users are recommended to change the value of seed before each execution but maintain consistent values of ra and dec. While ra and dec are not critical parameters, they are used by imaging tools such as ds9 or ximage when reading these event tables to define the center of the displayed image. Note that this tool overwrites its input table. If this is undesirable, make a copy of the raw files to work upon.

2. Screen the events using time-tagged quality information both internal and external to the file.

- uvotscreen infile=uvot/event/sw00072901259ubbpo_uf.evt attorbfile=00072901259/auxil/sw00072901259sao.fits.gz outfile=sw00072901259ubbpo_cl.evt badpixfile=caldb aoexpr="ANG_DIST < 100. && ELV > 10. && SAA == 0" evexpr="QUALITY == 0"

  In order to keep events with corrupted time tags, as in our example at the top of the recipe, we should replace the evexpr argument with an OR statement exexpr=QUALITY == 0 —— QUALITY == 32. Repeat this step for every event table produced by coordinator. A full list of parameters contained in the external attitude and orbit file, sw00072901259sao.fits.gz, and their definitions can be found by opening the header keywords of the file using FITS viewer fv, e.g. on the command line: fv sw00072901259sao.fits.gz and opening the header keyword extension.

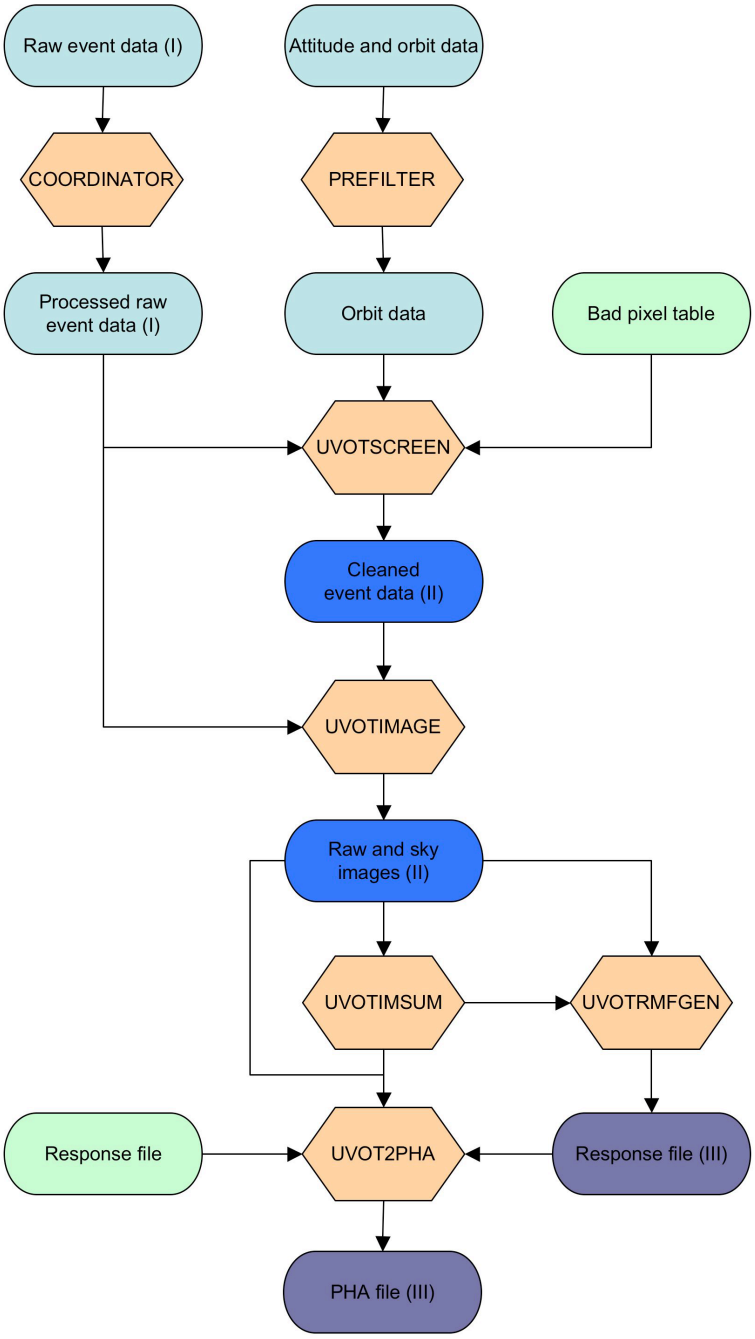

Figure 2.2: Flow of the UVOT event chain. Orange boxes represent software tools, green calibration products, light-blue Level I science products, medium-blue Level II science products and dark-blue Level III science products.

2.3 Summing a series of images

Synopsis

UVOT images generally range in exposure time from 10s to 2,000s. To obtain the deepest observation possible from a sequential or non-sequential series of images, either of the same filter or different filters, it is necessary to co-add. This is non-trivial because both the pointing and rotation of each exposure will be slightly different, requiring a procedure of shifting and rebinning. We use the HEASoft image analysis package ximage to demonstrate how this can be achieved.

Recipe I

1. Call the ximage package from the command line, co-add,e.g., three images, write out the new image to a file, total_sk.img, and quit the package:

- Namibia> ximage

  [XIMAGE> read 00072901259/uvot/image/sw00072901259ubb_sk.img+1

  [XIMAGE> save

  [XIMAGE> read 00072901259/uvot/image/sw00072901259ubb_sk.img+5

  [XIMAGE> sum

  [XIMAGE> save

  [XIMAGE> read/size=2632 00072901259/uvot/image/sw00072901259uvv_sk.img+7

  [XIMAGE> sum

  [XIMAGE> save

  [XIMAGE> write total_sk.img

  [XIMAGE> exit

  Images contained within the same file, and images contained in separate files can be co-added. In this example we combine the first image extension in the B filter file, the fifth image extension in the same file and the first image extension in the V filter file.

  Ximage makes a default assumption that all images are smaller than 1024x1024 pixels in size. Any images larger than this will be automatically truncated unless the user supplies the size of the image in the read command. A reference RA/Dec can also be chosen (e.g., "read/size=512/ra=299.7/dec=35.25"). UVOT sky images can be significantly larger than this limit. For example, an unbinned, full-frame, raw image rotated on the sky by 45 will result in a sky image of 2896x2896 pixels. In the example we supply the size of the third image, which exceeds the ximage default size.

2. Repeat the above step for the exposure maps corresponding to each of the images:

- Namibia> ximage

  [XIMAGE> read 00072901259/uvot/image/sw00072901259ubb_ex.img+1

  [XIMAGE> save

  [XIMAGE> read 00072901259/uvot/image/sw00072901259ubb_ex.img+5

  [XIMAGE> sum

[XIMAGE> save

[XIMAGE> read/size=2632 00072901259/uvot/image/sw00072901259uvv_ex.img+5

[XIMAGE> sum

[XIMAGE> save

[XIMAGE> write total_ex.img

[XIMAGE> exit

3. It is a trivial exercise to normalize the image by the exposure map, if desired, using the ftool farith:

- Namibia> farith infil1=total_sk.img+1 infil2=total_ex.img+1 outfil=total_nm.img ops=div null=y

Recipe II

Recipe I above can be a relatively typing-intensive and time consuming process, but provides the user with the freedom to co-add any number of images from a variety of different files. A faster method has been coded as part of the Swift pipeline in order to provide the deepest possible images for each filter as level III product. The increase in speed is obtained because the tool is aware that all UVOT sky images are oriented East-North, whereas ximage makes no such assumption. Scientists can employ this faster tool with the caveat that all images in a specified file will be co-added by default.

1. Co-add all image extensions for a specified file:

- uvotimsum infile=00072901259/uvot/image/sw00072901259ubb_sk.img outfile=total_sk_bb.img method=GRID

2. Perform the same operation for the appropriate set of exposure maps:

- uvotimsum infile=00072901259/uvot/image/sw00072901259ubb_ex.img outfile=total_ex_bb.img method=GRID

Figure 2.3 illustrates typical co-added images and exposure maps. The variation of window size during the observing sequence is generally apparent in these images.

Note that recipe II is valid only for summing exposures within the same sequence. Images from different sequences can be appended to image files using ftools such as fappend and ftappend (see the ftool online help; e.g. by typing fhelp fappend), although this process can be as time consuming as recipe I.

2.4 Extracting source counts from an image

Synopsis

One of the fundamental properties provided by the UVOT detector is source brightness. This recipe illustrates a simple method of extracting source counts from an image, employing region files similar to those constructed in Section 2.4. The first recipe performs count extraction from a single generic image, whereas the second recipe extracts a series of source counts from the sequence of images in a UVOT FITS image file.

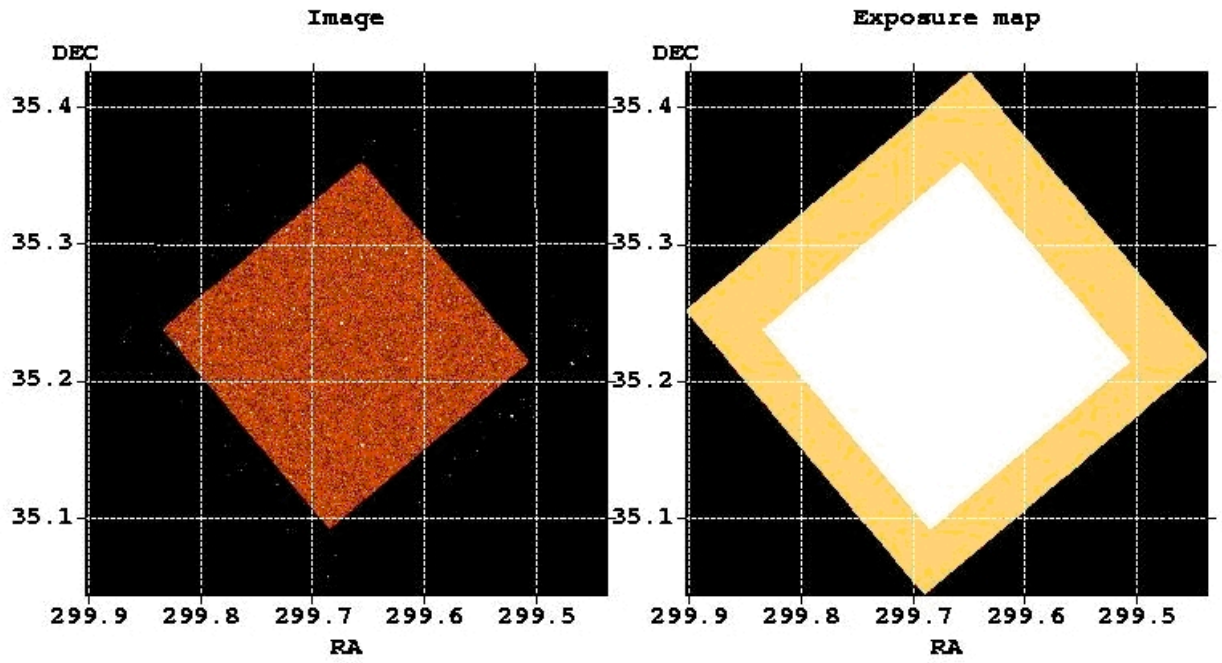

Figure 2.3: Examples of co-added images and the corresponding exposure map.

Recipe I UVOTSOURCE (see Section 3.9) provides the easiest way to calcuate source counts and magnitudes in images. The example below illustrates how to obtain counts, magnitudes and fluxes from a FITS file image extension:

uvotsource image=sw00221755992uwh_sk.img+1 srcreg=source.reg bkgreg=back.reg \
sigma=3 filter=WHITE method=aperture cleanup=yes chatter=1

Recipe II UVOTMAGHIST is a script that calls all image extensions in a UVOT file, in sequence, and calculates and plots the magnitudes.

1. Call uvotmaghist, creating a FITS and GIF record of count rate over time.

- Namibia> uvotmaghist infile=00072901259/uvot/image/sw00072901259ubb_sk.img \
  outfile=maghist.fit plotfile=maghist.gif zerofile=caldb coinfile=caldb \
  ra=23.35 dec=41.823 srcas=3 bkgas=10

  The advantage of recipe II is that it also converts source count rates to instrumental magnitudes and fluxes, storing these quantities in the output FITS table.

2.5 Screening event data

Synopsis

The general tool for event screening, developed as part of the HEASoft package, is xselect. This is a workhorse applied to data from multiple missions and has a wide variety of applications including screening and the extraction of light curves, image and spectra. A thorough understanding of this tool is recommended and can be found at
http://heasarc.gsfc.nasa.gov/docs/software/ftools/xselect/xselect.html.

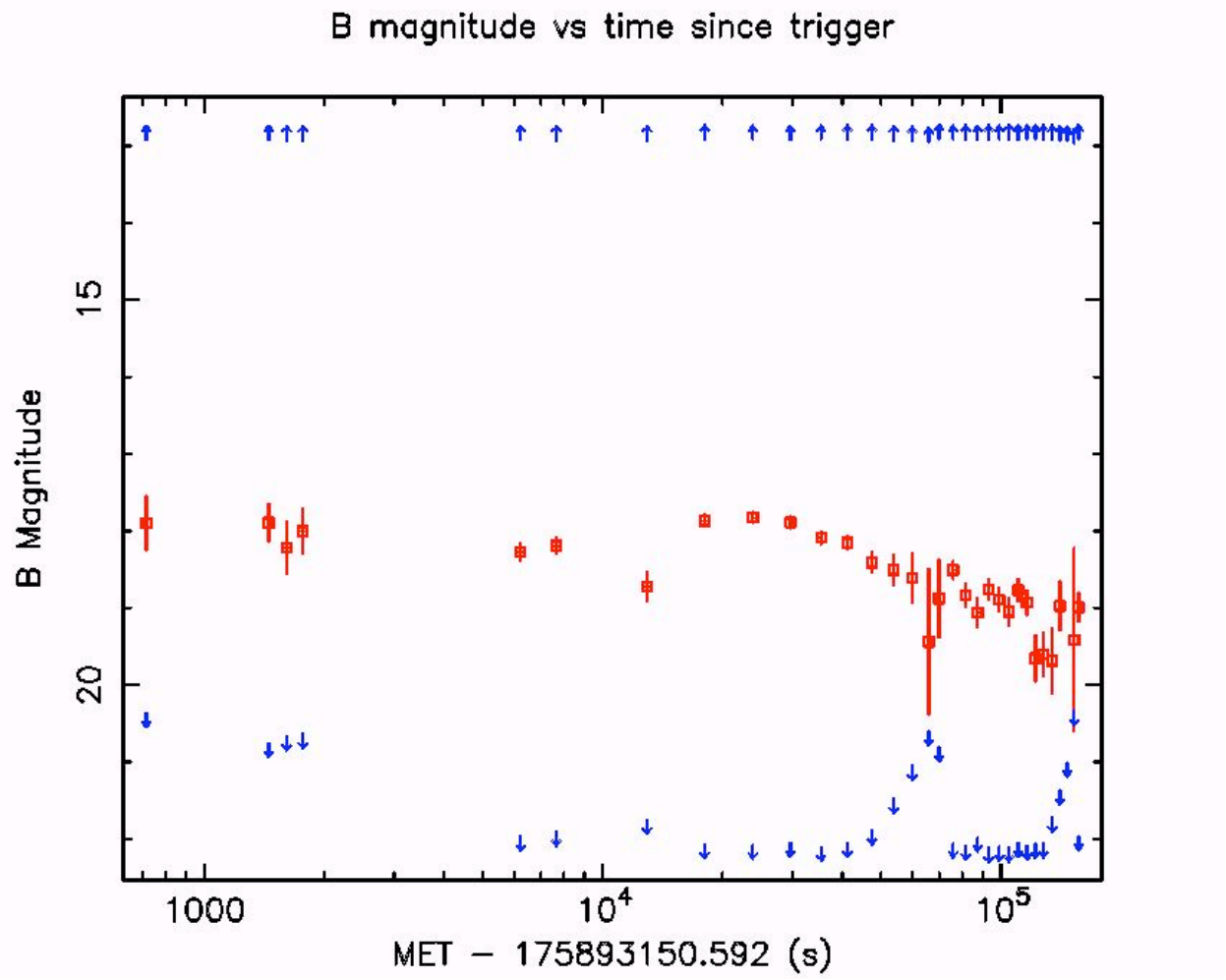

Figure 2.4: Example GIF output from uvotmaghist.

Recipes 2.6-2.8 provide only the simplest examples of xselects features. In the recipe below we will create a file of Good Time Intervals (GTIs) based on the angular distance between the telescope pointing and the bright Earth limb, e.g. to minimize Earth-glow effects in the data. Events occurring within the GTIs will be included in an output table, while those occurring outside the intervals will be thrown away.

Recipe

1. Create a GTI file based on data from the orbit filter file contained in the auxiliary directory of your data:

- Namibia> maketime infile=00072901259/auxil/sw00072901259sao.fits.gz outfile=br_earth.gti expr=BR_EARTH > 20. name=NAME value=VALUE time=TIME compact=n prefr=1 postfr=1

2. Read an event table into xselect and screen using the GTI file:

- Namibia> xselect

> Enter session name >[xsel] bertie

xsel:ASCA > read events sw00072901259ubbpo_cl.evt

> Enter the Event file dir >[./] uvot/event

Got new mission: SWIFT

> Reset the mission ? >[yes] y

xsel:SWIFT-UVOT-EVENT > filter time file br_earth.gti

xsel:SWIFT-UVOT-EVENT > extract events

xsel:SWIFT-UVOT-EVENT > save events output.evt

Wrote events list to file output.evt

> Use filtered events as input data file ? >[no] n

xsel:SWIFT-UVOT-EVENT > exit

> Save this session? >[no] n

2.6 Extracing an image from event data

Synopsis

The general tool for event extraction, developed as part of the HEASoft package, is xselect. A thorough understanding of this tool is recommended and can be found at http://heasarc.gsfc.nasa.gov/docs/software/ftools/xselect/xselect.html.

Recipes 2.6-2.8 provide only the simplest examples of xselects features. In the recipe below we will assume that the event file has been screened by the user, and extract a sky image from the event list.

Recipe

1. Read an event table into xselect, extract, plot and save an image:

- Namibia> xselect

  > Enter session name >[xsel] bertie

  xsel:ASCA > read events sw00072901259ubbpo_cl.evt

  > Enter the Event file dir >[./] uvot/event

  Got new mission: SWIFT

  > Reset the mission ? >[yes] y

  xsel:SWIFT-UVOT-EVENT > set xycenter 2000.5 2000.5

  xsel:SWIFT-UVOT-EVENT > set xysize 300 450

  xsel:SWIFT-UVOT-EVENT > set xybinsize 2

  xsel:SWIFT-UVOT-EVENT > extract image

  xsel:SWIFT-UVOT-EVENT > plot image

  xsel:SWIFT-UVOT-EVENT > save image output.img

  Wrote image to file output.img

> Use filtered events as input data file ? >[no] n

xsel:SWIFT-UVOT-EVENT > exit

> Save this session? >[no] n

By setting xycenter, xysize and xybinsize, the user has the option to specify the center, dimensions and binning of the output image.

2.7 Extracing a lightcurve from event data

Synopsis

Tool for light curve extraction, developed as part of the HEAsoft package, are uvotevtlc and xselect. A thorough understanding of the xselect tool is recommended and can be found at http://heasarc.gsfc.nasa.gov/docs/software/ftools/xselect/xselect.html. The usage of uvotevtlc is discussed in Section 5.11.

Recipes 2.6-2.8 provide only the simplest examples of xselects features. In the recipe below we will assume that the event file has been screened by the user, and extract a source light curve from the event list. This recipe requires a region file, resembling the source file from Sec. 2.4.

Recipe

1. Read an event table into xselect, extract using a region file, plot and save a light curve:

- Namibia> xselect

  > Enter session name >[xsel] bertie

  xsel:ASCA > read events sw00072901259ubbpo_cl.evt

  > Enter the Event file dir >[./] uvot/event

  Got new mission: SWIFT

  > Reset the mission ? >[yes] y

  xsel:SWIFT-UVOT-EVENT > filter region ds9.reg

  xsel:SWIFT-UVOT-EVENT > extract curve

  xsel:SWIFT-UVOT-EVENT > plot curve

  xsel:SWIFT-UVOT-EVENT > save curve output.lc

  Wrote light curve to file output.lc

  > Use filtered events as input data file ? >[no] n

  xsel:SWIFT-UVOT-EVENT > exit

  > Save this session? >[no] n

2. Read an event table into uvotevtlc, extract using source and background region files, and save the light curve:

- Namibia> uvotevtlc infile=sw00072901259ubbpo_cl.evt outfile=output.lc srcreg=src.reg bkgreg=bkg.reg

2.8 Extracing a spectrum from event data

Synopsis

Grism data obtained in Event mode is particularly useful for measuring spectral evolution over time. in the recipe below we will assume that the event file has been screened for time constraints by the user, extract an energy spectrum from the resulting event list. A pre-requisite for this recipe is to have passed the event file through the UVOT tool uvotevgrism.

Recipe

1. Create a binned spectrum over wavelength, with bin size of 20 Angstrom:

- Namibia> fhisto infile=00072901259/uvot/event/sw00072901259uvupo_cl_time1.evt+1 outfile=spectrum_time1.fits column=WAVELENGTH binsz=20

  Note that this recipe provides a spectrum of counts, calibrated over wavelength. This is, of course suitable for comparing count rates at different epochs. If you have a requirement to flux the data, or fit spectral model in e.g., xspec, the correct procedure would be to extract an image from the event table in DETX, DETY units, and then pass the image through the uvotimgrism and uvotrmfgen tools (Sec 2.11).

2.9 Lightcurve analysis

Synopsis

The xronos package (http://heasarc.gsfc.nasa.gov/docs/xanadu/xronos/xronos.html) has been installed on your machine as part of the LHEAsoft distribution. It contains applications dedicated to the analysis of timing data, such as the light curve produced by xselect in Sec. 2.7. Table 2.10.1 lists a subset of these applications. More details and applications can be found at the URL above.

Table 2.10.1: xronos applications for time series analysis.

Application	Description
autocor	Auto-correlation
crosscor	Cross-correlation of two time series
earth2sun	Barycentric correction
efold	Epoch folding
efsearch	chi2 period search
flc2ascii	FITS-to-ASCII conversion
lcmath	Arithmetic between two time series
lcstats	Time series statistics
lcurve	Creates binned light curves and colour diagrams
powspec	Power density analysis

The recipe below provides a simple example of rebinning and plotting a UVOT light curve, which was originally extracted from Event mode data using xselect.

Recipe I

1. Load the unbinned light curve into the lcurve application:

- Namibia> lcurve

  – lcurve 1.0 (xronos5.21)

  Number of time series for this task[1] 1

  Ser. 1 filename +options (or @file of filenames +options)[] sw00000001001ubb_sr.lc

  Series 1 file 1:sw00000001001ubbsr.lc

  Selected FITS extensions: 1 - RATE TABLE;

  Source ..... Safe Pointing 1 Start Time (d) 12991 15:34:19.567

  FITS Extension .... 1 - `RATE` Stop Time (d) 12991 16:17:38.559

  No. of Rows ..... 35665 Bin Time (s) 0.1100E-01

  Right Ascension ... 23.35 Internal time sys.. Converted to TJD

  Declination ..... -41.82305556 Experiment ..... SWIFT UVOTA

  Filter ..... B

  Corrections applied: Vignetting - No ; Deadtime - No ; Bkgd - No ; Clock - No

  Selected Columns: 1- Time; 2- Y-axis; 3- Y-error; 4- Fractional exposure;

  File contains binned data.

  Name of the window file ('-' for default window)[-]

  Expected Start ... 12991.64883758009 (days) 15:34:19:567 (h:m:s:ms)

  Expected Stop .... 12991.67891850347 (days) 16:17:38:559 (h:m:s:ms)

  Minimum Newbin Time 0.11000000E-01 (s)

  for Maximum Newbin No.. 236272

  Default Newbin Time is: 5.0821114 (s) (to have 1 Intv. of 512 Newbins)

  Type INDEF to accept the default value

  The tool has read the light curve and provided some statistics to your shell such as the source name and position, the instrument name and filter, type of data, start and stop times, the time system, the amount of data and its binning.

2. Choose an new bin size:

- Newbin Time or negative rebinning[100] 1.0

  Newbin Time ..... 1.0000000 (s)

  Maximum Newbin No. 2599

  Default Newbins per Interval are: 512

  (giving 6 Intervals of 512 Newbins each)

  Type INDEF to accept the default value

3. Segregate the data into a number of time intervals. This is particularly useful for plotting the evolution of colour-colour diagrams over the time interval, but has limited value in this example. The full set of data has 2599 1-s bins, so we plot them all together:

- Number of Newbins/Interval[98] 2599
  Maximum of 1 Intvs. with 2599 Newbins of 1.00000 (s)

4. Define the output filename and the output plotting device:

- Name of output file[default] output.flc
  Do you want to plot your results?[yes] yes
  Enter PGPLOT device[/XW] /XW
  2599 analysis results per interval
  100% completed
  Intv 1 Start 12991 15:34:20
  - Ser.1 Avg 9.940 Chisq 337.0 Var 14.50 Newbs. 405
    * Min 0.000 Max 40.11 expVar 12.41 Bins 35665

  PLT> q
  Writing output file: output.flc

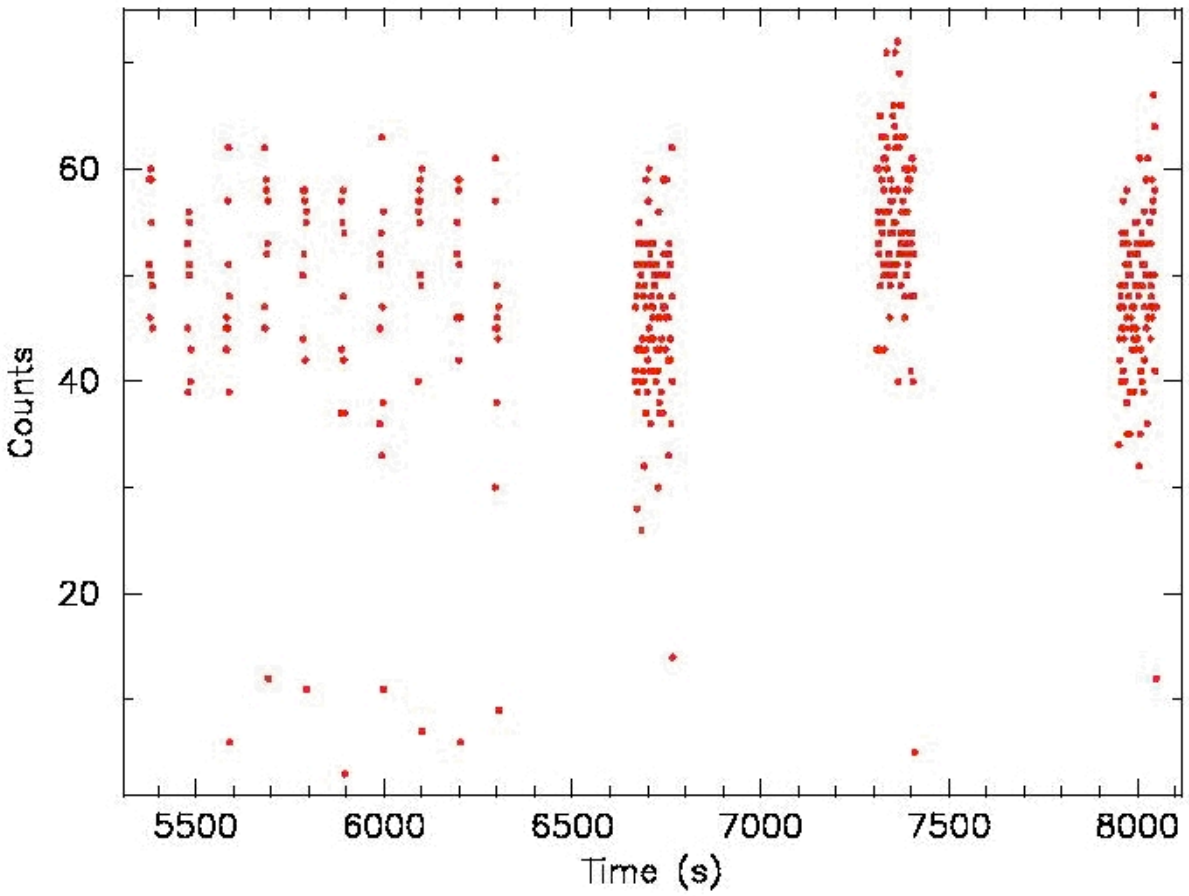

Figure 2.5: Time series output from the lcurve tool.

- Before plotting the new light curve to the xserver, the tool provides a few statistics, such as the average count rate, minimum and maximum count rate values and variability measures.

Recipe II

UVOT data from different filter can never be simultaneous, therefore plotting colour information is not practical without choosing large time bins. However much more detailed colour analysis is possible if further time series data is loaded into the tool from e.g. the XRT or ground-based observatories.

1. In these instance the first input parameter asks or the number of time series used in the analysis:

- Number of time series for this task[1]

  In the previous example we used only one time series. But we can use up to three. A minimum of two time series are required to plot a colour ratio versus time, while three are required to plot a colour-colour diagram.

2. After the plotting device has been set, the tool will ask what type of plot is required.These are designated by a numeric character whose meaning depends on the number of time series used:

- Enter PLOT style number (default=1)[]

  For the two-series case:

  1 = hardness ratio versus time

  2 = intensity of both series versus time

  3 = both intensity and harness ration versus time

  For the three-series case:

  1 = colour-colour diagram

  3 = plot all three intensities versus time

2.10 Extracting a grism spectrum

Synopsis

This recipe illustrates the method of extracting a calibrated grism spectrum from an image. The tool uvotimgrism extracts and calibrates the spectrum internally, producing a FITS table containing a fluxed spectrum in units of erg s**-1 cm**-2 Angstrom**-1, over wavelength, in units of Angstrom. We have striven to make UVOT spectral data consistent with that expected from the xspec spectral fitting package (http://heasarc.gsfc.nasa.gov/docs/xanadu/xspec). This requires the raw spectrum and calibration to be separated into distinct files. Therefore the output from uvotimgrism will have two extension of spectral data, the first containing a raw spectrum and the second containing a fully-calibrated spectrum. Users wishing to use xspec will have to perform a further task, creating a response matrix for the spectrum.

The uvotimgrism tool requires an input image in detector (DET) coordinates rather than RA and Dec (SKY). The user must supply the centroid of the zero order of the target source in detector pixels. As of this writing, the option to supply a RA and Dec position for the centroid is not yet available because the grism distortion has not been mapped, and so the astrometry is only good to about 5". Instead the centroid must be independently determined (e.g. using XIMAGE or DS9). The user also needs to specified the angle that the first order makes with the X-axis. Approximate values of this angle are 148.1 (V grism, nominal), 140.5 (V grism, clocked), 151.4 (U grism, nominal) and 144.5 (U grism, clocked).

Recipe

1. Extract a spectrum from a DET grism image:

- Namibia> uvotimgrism infile=sw00000001001ugv_dt.img.gz+1 outfile=sw00000001001ugv_1.pha backfile=sw00000001001ugv_1_bk.pha badpixfile=badpix.img+1 wavefile=caldb areafile=caldb teldeffile=caldb ra=-1 dec=-1 sourcex=1553.67 sourcey=611.200 ang=148.1 srcwid=21 bkgwid1=20 bkgwid2=20 bkgoff1=7 bkgoff2=7 wavemin=2900 wavemax=5500 nsigma=5 cleanup=y clobber=y history=y chatter=1

This call to uvotimgrism will output a region file sw00000001001ugv_dt.img.reg which can be overlayed on the detector image in DS9 to examine the source and background extraction regions. In addition, the third extension of the output spectrum file is an image containing just the 0th and 1st order light of the source, plus the background regions chosen by the input parameters above. The image is rotated so that the X-axis is parallel with the 1st order dispersion direction, as in Fig 2.6.

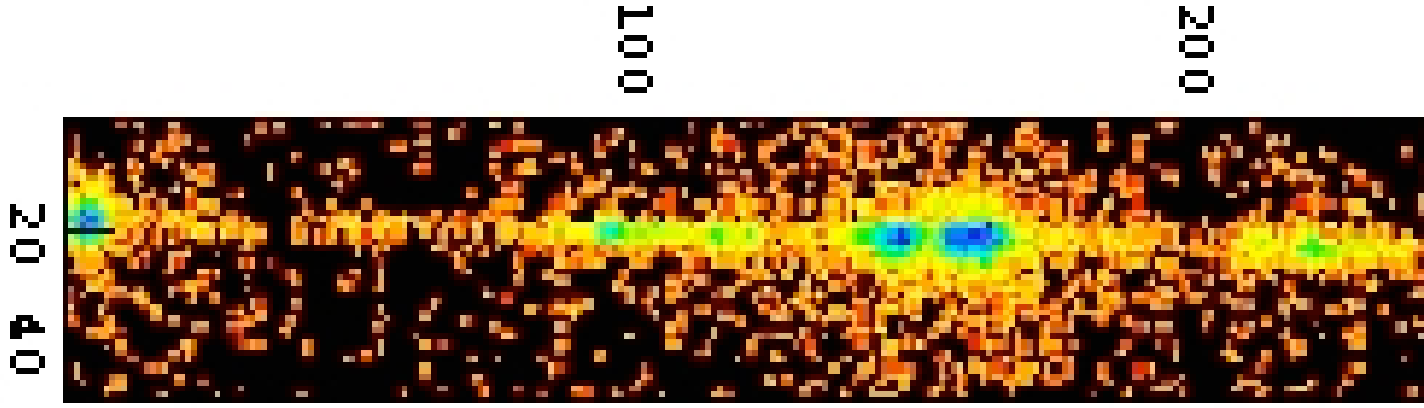

Figure 2.6: A typical image contained in the output from uvotimgrism.

- Plotting the wavelength column in the third extension against counts, using, e.g., fv, yields a spectrum similar to that in Fig. 2.7.

2. Construct a response matrix, containing wavelength and flux calibration data suitable for use in xspec using uvorfmrgen:

- Namibia> uvotrmfgen spectrum=sw00000001001ugv_1.pha outfile=sw00000001001ugv_1.rsp areafile=caldb lsfile=caldb

2.11 Fitting grism data

Synopsis

This recipe illustrates a simple fitting procedure using xspec. The xspec package has a large number of useful features that we will not discuss here, but needless to say, the user will find reading the xspec manual a worthwhile activity before ploughing too far into this topic:

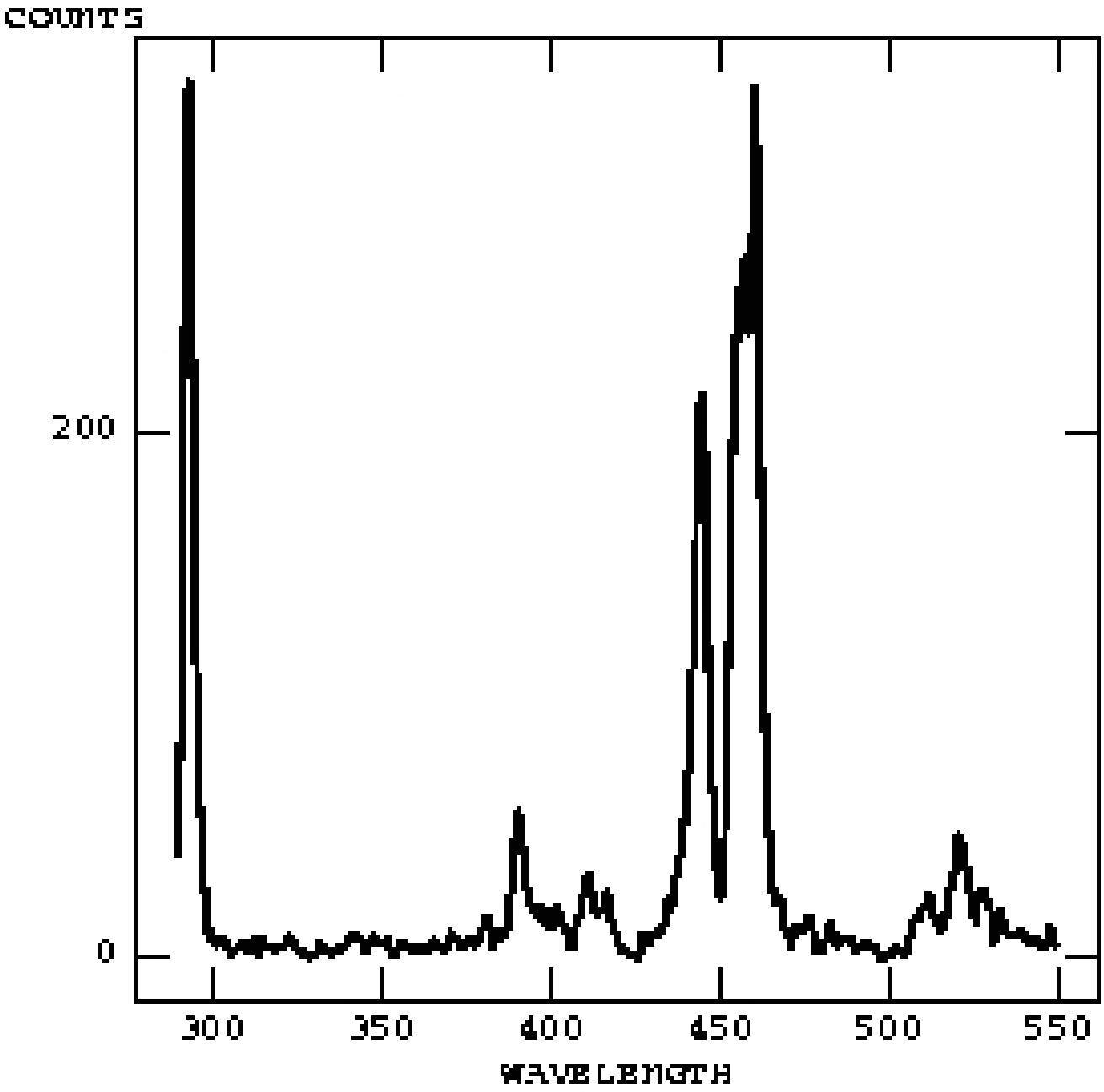

Figure 2.7: Spectrum output from uvotimgrism.

http://heasarc.gsfc.nasa.gov/docs/xanadu/xspec

We caution that the UVOT grism has many sources of systematic error (e.g. order overlap) that are not reflected in the count statistics.

Recipe

1. Start xspec and load grism source and background spectra and the response matrix:

- Namibia> xspec

  XSPEC version: 12.3.0

  Build Date/Time: Wed Aug 2 11:08:42 2006

  XSPEC>data spectrum.pha

  Net count rate (cts/s) for file 1 2.834 ± 6.3628E-02(90.7% total)

  - using background file... background.pha
    1 data set is in use

  XSPEC>resp spectrum.rmf

2. Ignore bad pixels:

- XSPEC>ignore bad

3. Define a spectral model and provide some starting parameters. In this example we will combine three multiplicative models a powerlaw (power), reddening (redden), and a neutral H edge, redshifted by z = 3 (zphabs):

- XSPEC>mo zphabs*redden*power

  - Model: zphabs<1>*redden<2>(powerlaw<3>)

  Input parameter value, delta, min, bot, top, and max values for ...

  - 1 0.001 0 0 1E+05 1E+06

  1:zphabs:nH> 1

  - 0 -0.01 0 0 10 10

  2:zphabs:redshift>1

  - 0.05 0.001 0 0 10 10

  3:redden:E(B-V)> 0.1

  - 1 0.01 -3 -2 9 10

  4:powerlaw:PhoIndex> 0.9

  - 1 0.01 0 0 1E+24 1E+24

  5:powerlaw:norm> 0.1

  -

- Model: zphabs<1>*redden<2>(powerlaw<3>)

  Model Fit Model Component Parameter Unit Value

  par par comp

  1 1 1 zphabs nH 10$\hat{\{\}}$22 1.00000 ± 0.00000

  - 2 2 1 zphabs redshift 1.00000 frozen
    3 3 2 redden E(B-V) 5.000000E-02 ± 0.00000
    4 4 3 powerlaw PhoIndex 1.00000 ± 0.00000
    5 5 3 powerlaw norm 0.100000 ± 0.00000

- Chi-Squared = 33685.34 using 522 PHA bins.

  Reduced chi-squared = 65.02961 for 518 degrees of freedom

  Null hypothesis probability = 0.00

  The information reported back to you after entering the initial fit parameters are the parameter values and chi$\hat{\{\}}$2 goodness-of-fit statistics. These will be updated every time you fit to the data.

4. Fit the data:

- XSPEC>fit 100

  Xspec now reports:

  -

- Model: zphabs<1>*redden<2>(powerlaw<3>)

  Model Fit Model Component Parameter Unit Value

  par par comp

  - 1 1 1 zphabs nH 10$\hat{\{\}}$22 1.00000 frozen
    2 2 1 zphabs redshift 3.00346 ± 0.106201E-02
    3 3 2 redden E(B-V) 2.280192E-02 ± 0.224064
    4 4 3 powerlaw PhoIndex 1.06898 ± 0.841269
    5 5 3 powerlaw norm 0.602656 ± 3.73260

- Chi-Squared = 564.6284 using 522 PHA bins.

  Reduced chi-squared = 1.0914510 for 518 degrees of freedom

  Null hypothesis probability = 1.00

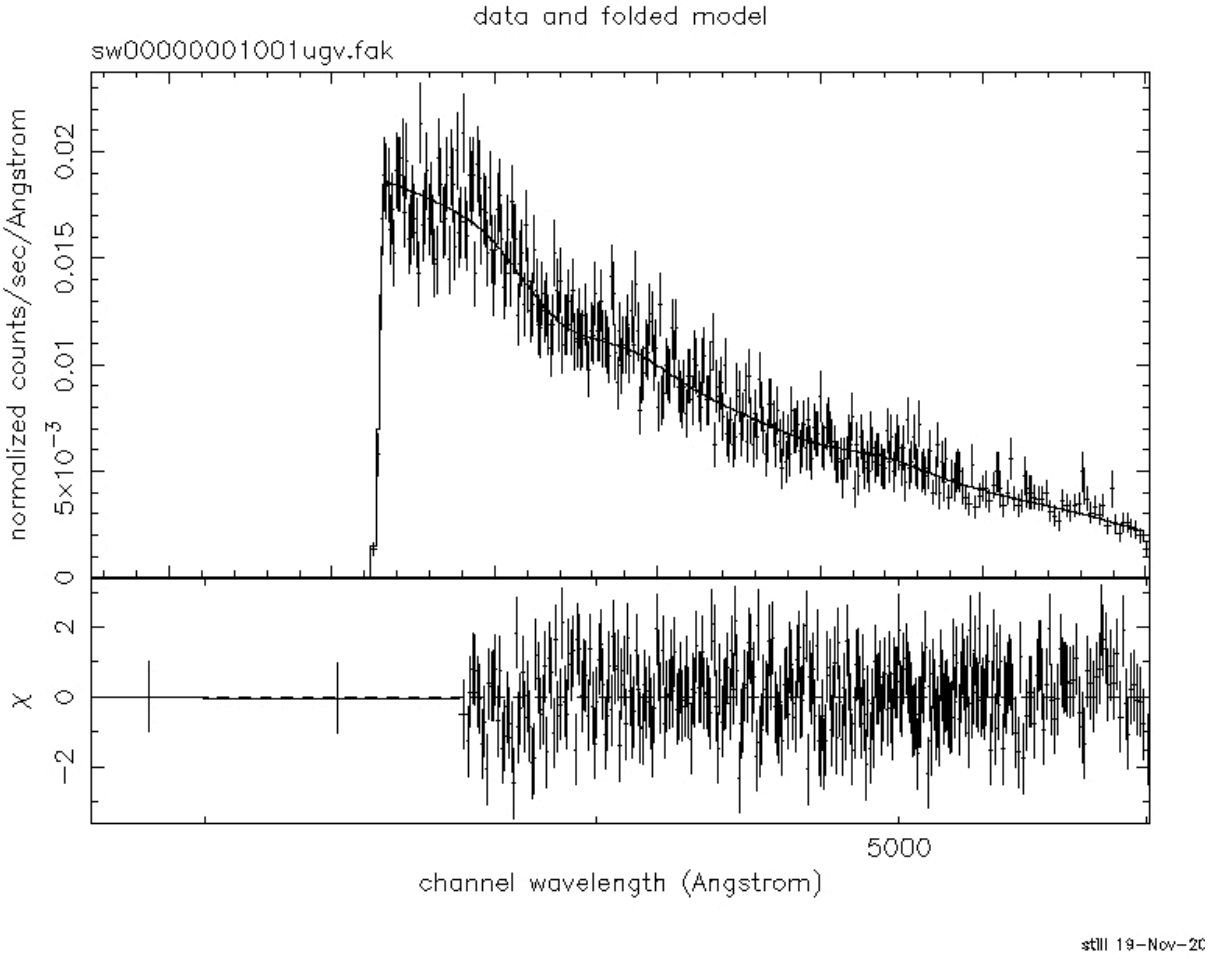

Figure 2.8: A VGRISM spectrum with fit residuals.

Which according to the chi$\hat{\{\}}$2 statistic is a good fit.

5. Uncertanties for each active parameter can be calculated thus:

- XSPEC>uncer 2-5

  Parameter Confidence Range (2.706)

  2 3.00097 3.00439 (-3.326416E-03, 9.632111E-05)

  3 0.00000 2.122159E-02 (0.00000 , 2.122159E-02)

  4 0.471290 1.25561 (-0.691941 , 9.237909E-02)

  5 0.313522 8.59642 (-4.709452E-03, 8.27819)

  The 2nd and 3rd columns contain the lower and upper 90% confidence limits.

6. Finally, set some plotting options and plot the data with fit, plus fit residials:

- XSPEC>cpd /xs

XSPEC>setplot wave

XSPEC>setplot rebin 5 100

XSPEC>plot data delchi

A typical output can be found in Figure 2.8.

2.12 Aspect Correction

Synopsis

Aspect correction is essential to obtain correct sky coordinates of UVOT sources and to ensure that individual exposures are added without offsets. This recipe gives a brief instruction on how to use the Swift FTOOLS aspect correction tool.

Recipe

Aspect corrections (i.e., shifts and rotations) can be applied to UVOT sky coordinate images using the Swift tool UVOTSKYCORR. In this recipe, we will use Swift observations of GRB050525A (sequence 00130088000) as an example.

Make sure that you have installed the WCSTools on your machine, which are needed to run the UVOT aspect correction. In particular, make sure that the WCSTool "scat" is installed.

The UVOTSKYCORR tool needs to be run twice to obtain aspect corrected UVOT images:

1. First the aspect corrections need to be computed by comparing UVOT source positions with those of catalogued sources.

2. In the next step the computed aspect corrections need to be applied to the image(s).

You can obtain a list of the parameters and default settings of the "uvotskycorr" task by typing:

$ plist uvotskycorr

Help on "uvotskycorr" is available by typing

$ fhelp uvotskycorr

The parameters are (see Section 3.16):

Parameter	Description
what	[string] (ID—SKY) Whether to find corrections (what=ID) or apply corrections (what=SKY).
skyfile	[filename] Name of input image file(s). This can be a comma-delimited list of file names or @<file> where <file> contains the names of the files to process, one per line.
corrfile	[filename]Input corrections file for what=SKY.
attfile	[filename] Input attitude file.
outfile	[filename] Output file name. For what=ID, the aspect corrections will be written to this file.
(starid = NONE)	[string] Parameters to pass to star identification.
(catspec = usnob1.spec)	[filename] Catalog descriptor file.
(cleanup = yes)	[boolean] Remove intermediate files?
(history = yes)	[boolean] Write history keywords?
(clobber = no)	[boolean] Overwrite existing files?
(chatter = 1)	[enumerated integer] Standard HEASoft chatter parameter.

Optional parameters are given in parenthesis.

The "what" parameter switches between computing ("what=ID") and applying ("what=SKY") aspect corrections. If aspect corrections are computed, the "outfile" parameter specifies the file used to write the aspect corrections. That file is read in the next step, when the aspect solutions are pplied to the image(s) ("what=SKY"). The "starid" parameters sets the values passed to a subroutine which control the star identification settings. The default parameters for "starid" are optimized to give best results.

Star catalog information is required to perform the aspect correction. The "catspec" parameter gives the catalog descriptor which describes how catalog information is to be loaded. It can point to a local catalog installed on your computer or one that is to be queried over the web. More information on how to access the catalog information is available here:

$ fhelp catspec

You can set "catspec=usnob1.spec" to read the parameter file usnob1.spec, which looks like this:

$ more usnob1.spec
# partition summary
type => StarID::SearchCat
fields => ID,RA,DEC,MAG,TYPE
packed => 0
data => GSC
catalog/type => Indexed
catalog/n => 4
envvar => UB1_PATH
# Local star catalog: on your machine:
#location => /ssdc/usnob1
USNO A server from [[BR]] #location => http://archive.eso.org/skycat/servers/usnoa-server[[BR]]
USNO B1 server from [[BR]] location => http://tdc-www.harvard.edu/cgi-bin/scat[[BR]] limit
=> 10000

In this case the USNO star catalog is retrieved over the web from Harvard.

1. Calculate the Aspect Correction:

To aspect-correct V-band UVOT images for our example (GRB050525A), the aspect solution has to be calculated first:

$ uvotskycorr what=ID \
skyfile=00130088000/uvot/image/sw00130088000uvv_sk.img \
attfile=00130088000/auxil/sw00130088000sat.fits \
outfile=00130088000/out/CORR.00035227003.ALL 'starid=mag.err=5 rot.error=60' \
chatter=5 catspec=usnob1.spec clobber=yes history=yes cleanup=yes corrfile=NONE

This command reads the input image and the spacecraft attitude file, and writes the output into a directory which needs to be created first. In addition, we opted to pass two parameters for the star identification, using matching magnitudes between UVOT and catalogued sources to within 5 mag and allow a rotation of the UVOT image of up to 60-arcmin. The "catspec" command gives the location of the star catalog as described above. You can give a comma-delimited list of UVOT images to be aspect-corrected or you can provide a file which lists all images to be aspect-corrected using "skyfile=@<file>".

2. Apply the Aspect Correction:

To apply the aspect corrections "uvotskycorr" needs to read in the aspect solution file created above.

$ uvotskycorr "what=SKY" \
skyfile=00130088000/uvot/image/sw00130088000uvv_sk.img \
attfile=00130088000/auxil/sw00130088000sat.fits \
corrfile=00130088000/out/CORR.00035227003.ALL \
chatter=5 catspec=usnob1.spec \ clobber=yes history=yes cleanup=yes

3. Check the Aspect Correction:

Check which header extensions were modified by "uvotskycorr" by inspecting the headers of the file extensions:

$ ftlist sw00130088000uvv_sk.img K include=EXTNAME,ASPCORR

Next, all individual exposures less the settling images (header extensions 1 and 2 in our example) and less the file extensions which were not corrected (none in our example) need to be added by running the command "uvotimsum":

$ uvotimsum sw00130088000uvv_sk.img uvv_sum.fits exclude=1,2 chatter=1

Open the added image "uvv_sum.fits" with DS9 or FV and visually inspect the improvement in the aspect solution.

The images 2.9 give the un-corrected (left) and corrected (right) summed V-band images of GRB 050525A. The XRT error position of GRB 050525A is marked by a circle with a radius of 6-arcsec as reported in GCNs. Note that the un-corrected image shows significant offsets between individual exposures, which have been corrected for in the aspect-corrected image. The afterglow candidate to the GRB (marked by two lines) is much fainter in the uncorrected image due to the offsets of the individual exposures. The aspect-corrected image also shows updated centroid positions of all UVOT sources, which also appear more point-like.

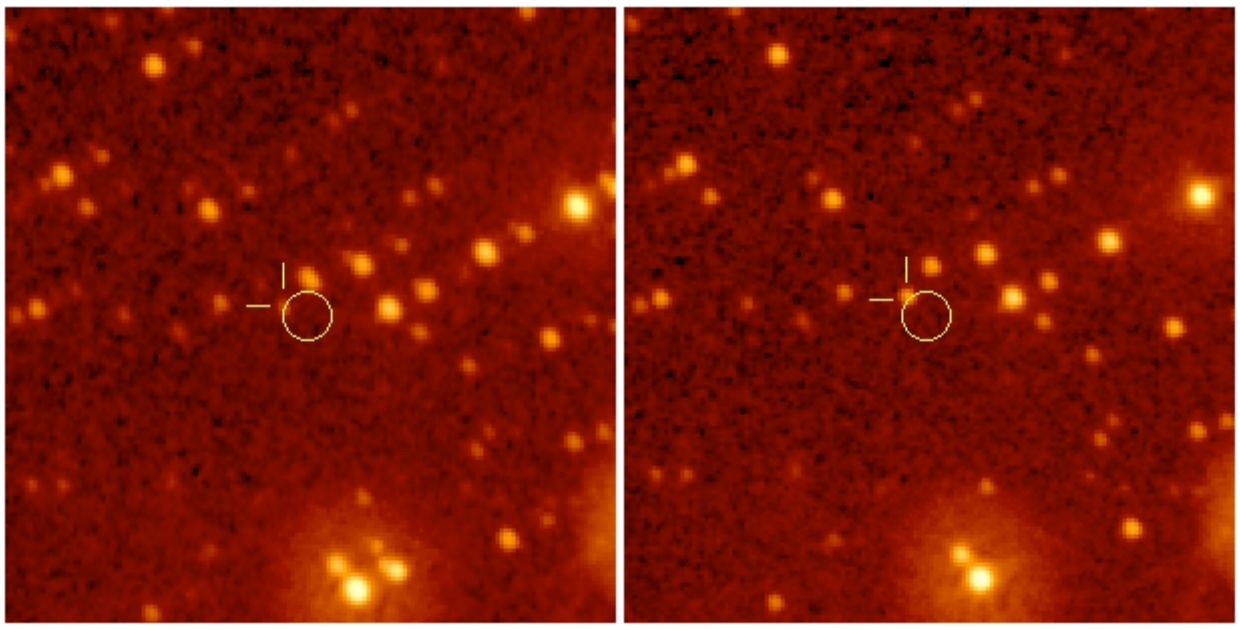

Figure 2.9: Merged UVOT B-filter images of GRB 050525A before (left) and after (right) aspect correction.

2.13 Broad-Band Spectral Fitting

Synopsis

This recipe gives a description how you can perform a broad-band (opt/UV/X-ray) spectral fitting to 6-filter UVOT and XRT data to obtain the spectral energy distribution (SED) of your source of interest. The example below uses data obtained on GRB 050525 (sequence 00130088000).

In this thread we describe how to

1) extract a spectrum from UVOT data,
2) shift the count rates of the spectrum to a common epoch,
3) extract an XRT X-ray spectrum, and
4) perform a joint spectral fitting of the UVOT and XRT data.

The example below uses data obtained on GRB 050525 (sequence 00130088000).

Recipe

1) Extract Spectrum from UVOT Data You can use individual UVOT images or co-add individual image extensions to one image per filter to increase the photon statistics:

```
uvotimsum sw00130088000uvv_sk.img uvv_sum.fits chatter=1
uvotimsum sw00130088000ubb_sk.img ubb_sum.fits chatter=1
uvotimsum sw00130088000uuu_sk.img uuu_sum.fits chatter=1
uvotimsum sw00130088000uw1_sk.img uw1_sum.fits chatter=1
uvotimsum sw00130088000um2_sk.img um2_sum.fits chatter=1
uvotimsum sw00130088000uw2_sk.img uw2_sum.fits chatter=1
```

You can prevent certain extentions from being coadded (if need) by employing the 'exclude' parameter, e.g.:

```
uvotimsum sw00130088000uw2_sk.img uw2_sum.fits chatter=1 exclude=1
```

Now load the images into DS9 and create source and background region files:

```
ds9 uvv_sum.fits &
```

The burst is located at RA = 18:32:32.6, Dec = 26:20:22.3. The source spectrum region file, 'source.reg', centered in the burst needs to be in WCS coordinates, either in degrees or in sexadesimal format:

```
fk5;circle(18:32:32.599,+26:20:22.27,6")
fk5;circle(278.13583,26.339519,6")
```

A background region file, 'back.reg', also needs to be created with DS9.

A response matrix is needed which defines the spectral properties of the data. These can be downloaded from the Swift web pages where there is one available for every lenticular UVOT filter. It is critical that the correct response matrix be used with the data (easily identified by the names of the files):

```
http://swift.gsfc.nasa.gov/docs/swift/proposals/swift_responses.html
```

Next, you can use the tool 'uvot2pha' to create a file that can be read into XSpec. Given the two region files, one containing source counts from a specific object, the other containing background

counts from around that source, UVOT2PHA will extract counts from both regions accompanied by Poisson uncertainties. These four quantities will be cast into two XSpec-compatible files.

uvot2pha infile=uvv_sum.fits srcpha=v.pha bkgpha=v_bkg.pha \
srcreg=source_v.reg bkgreg=back_v.reg respfile=v.rsp clobber=y chatter=1

uvot2pha infile=ubb_sum.fits srcpha=b.pha bkgpha=b_bkg.pha \
srcreg=source_b.reg bkgreg=back_b.reg respfile=b.rsp clobber=y chatter=1

uvot2pha infile=uuu_sum.fits srcpha=u.pha bkgpha=u_bkg.pha \
srcreg=source_u.reg bkgreg=back_u.reg respfile=u.rsp clobber=y chatter=1

uvot2pha infile=uw1_sum.fits srcpha=uvw1.pha bkgpha=uvw1_bkg.pha \
srcreg=source_uvw1.reg bkgreg=back_uvw1.reg respfile=uvw1.rsp clobber=y chatter=1

uvot2pha infile=um2_sum.fits srcpha=uvm2.pha bkgpha=uvm2_bkg.pha \
srcreg=source_uvm2.reg bkgreg=back_uvm2.reg respfile=uvm2.rsp clobber=y chatter=1

uvot2pha infile=uw2_sum.fits srcpha=uvw2.pha bkgpha=uvw2_bkg.pha \
srcreg=source_uvw2.reg bkgreg=back_uvw2.reg respfile=uvw2.rsp clobber=y chatter=1

2) Correct for a temporal variation/decay of the source: It is important to note that some astronomical objects, such as GRBs and supernovae, vary in flux on relatively short-term time scales. In order to do a correct broad-band spectral fitting, the variability of the source therefore has to be taken into account. There are two methods you could chose which are described below. We leave it to the user which method is employed (this is where the 'art' of being a scientist comes in).

1) Select data from simultaneous epochs:

Select your epoch of interest for which you want well sampled data in the X-ray and UV/optical. For X-ray data you can extract the time-interval of interest within 'xselect' and produce a spectrum that can be read into XSpec. For the UVOT you want to obtain the exposures in each filter that correspond to the epoch of intestest and co-add them into one image per filter.

2) Fit data to get an SED at an instantaneous epoch:

For this method you need to fit each light curve individually and use the fit to determine the corresponding count rate at the epoch of interest. In the case of the UVOT filters, to be accurate, you want to perform a simultaneous fit to get an accurate measure of the decay rate. You then re-fit your data filter by filter, fixing the decay slope in each case to the best-fit value determined earlier. If you want to check if your source shows evidence for a color evolution, you have to do this for multiple epochs. Once you have derived count rates, you then produce your spectral files as described below, within 'xselect' and using 'uvot2pha'. You then need to update the 'EXPOSURE' header keyword appropriately so that the count rate will be the one that you measured in your fits.

In detail, these are the steps that have to be performed to shift a UVOT .pha file to common epoch:

Input:

- source.pha file
- background.pha file

1) Scale the background counts (or count rate) so that it has the same area as the source counts.

2) Subtract the scaled background counts from the source counts.

3) Compute the counts at the time of the common epoch using the observed decay rate of the afterglow. For a power-law decay the relationship is

- counts_common_epoch = count_original * (t_common_epoch / t_original)$\hat{}${}alpha

  where t is the time since the BAT trigger.

4) Add the background to the shifted counts to get the total counts at the common epoch.

5) Propagate the errors.

- sp = statistical error in shifted counts
  s = statistical error in original counts
  sb = statistical error in background counts
  f = (t_common_epoch / t_original)$\hat{}${}alpha
  g = area of source region / area of background region

  sp = SQRT([f * s]$\hat{}${}2 + [f * g * sb]$\hat{}${}2)

3) Extract Spectrum from XRT Data: Load the cleaned XRT photon-counting events file into ds9

ds9 sw00130088000xpcw4po_cl.evt

and create a region file centered in the X-ray sources, as well as a background region file. In our case, we chose a circular region file in WCS sky coordinates, centered on the source, and an annulus around the source as background region file:

source_xrt.reg:
fk5;circle(278.13583,26.33951,47")

choosing a circle of radius 20 pixel (47 arcsec) which corresponds to the 90% encircled energy radius at 1.5 keV. The background region files has the form:

back_xrt.reg:
annulus(278.13571,26.339051,75",150")

'xselect' can be used to extract counts from the events file using a spatial filtering with the region files, and writing them to spectral files suitable for XSpec:

xselect

> Enter session name >[xsel]
xsel:SUZAKU > read events sw00130088000xpcw4po_cl.evt
> Enter the Event file dir >[./]
Got new mission: SWIFT
> Reset the mission ? >[yes]

Notes: XSELECT set up for SWIFT
Time keyword is TIME in units of s
Default timing binsize = 5.0000

Setting...
Image keywords = X Y with binning = 1
WMAP keywords = X Y with binning = 1

Energy keyword = PI with binning = 1

Getting Min and Max for Energy Column...
Got min and max for PI: 0 1023

Got the minimum time resolution of the read data: 2.5073
MJDREF = 5.1910000742870E+04 with TIMESYS = TT
Number of files read in: 1

Observation Catalogue:

Data Directory is: /namibia/00130088000/xrt/event/ HK Directory is: /namibia/GRB050525/00130088000/xr

OBJECT OBS_ID DATE-OBS DATAMODE
1 GRB050525 00130088000 2005-05-25T PHOTON

xsel:SWIFT-XRT-PHOTON > set image sky
xsel:SWIFT-XRT-PHOTON > filter region source_xrt.reg
xsel:SWIFT-XRT-PHOTON > extract spectrum

extractor v4.67 11 Jul 2006
Getting FITS WCS Keywords
Doing file: /namibia/00130088000/xrt/event/sw00130088000xpcw4po_cl.evt
100% completed
Total Good Bad: Region Time Phase Grade Cut
3269 609 2660 0 0 0 0
Grand Total Good Bad: Region Time Phase Grade Cut
3269 609 2660 0 0 0 0
in 5755.9 seconds
Spectrum has 609 counts for .1058 counts/sec
written the PHA data Extension

xsel:SWIFT-XRT-PHOTON > save spectrum xrt.pha

Wrote spectrum to xrt.pha

Next, do the same for the background region file and create a background spectrum, xrt_back.pha.

Now the response matrix and ancillary response file need to be created using xrtmkarf or downloaded from the Swift calibration site. In this example, we write the headers for the generic response files into the XRT spectrum:

xrtmkarf
Name of the input PHA FITS file[] xrt.pha
PSF correction active?(yes/no)[yes]
Name of the output ARF FITS file[] xrt.arf
Source X coordinate (SKY for PC and WT modes, DET for PD mode):[-1] 278.13583
Source Y coordinate (SKY for PC and WT modes, DET for PD mode):[-1] 26.339519

grppha
Please enter PHA filename[xrt.pha] xrt.pha
Please enter output filename[xrt.pha] xrt.pi

```
GRPPHA[] chkey RESPFILE swxpc0to12_20010101v008.rmf
GRPPHA[] chkey ANCRFILE xrt.arf
GRPPHA[] chkey BACKFILE xrt_back.pha
GRPPHA[] exit
```

```
    written the PHA data Extension
exiting, changes written to file : xrt.pi
grppha 3.0.0 completed successfully
```

4) Joint Spectral Fitting of UVOT and XRT Data Start an XSpec session, read in the UVOT and XRT data, ignore energy channels outside the energy range of the instrument, and fit the data:

```
    XSPEC>cpd /xw
XSPEC>data 1:1 v.pha 1:2 b.pha 1:3 u.pha 1:4 uvw1.pha 1:5 uvm2.pha 1:6 uvw2.pha 2:1 xrt.pha
XSPEC>setpl en
XSPEC>ignore bad
XSPEC>ignore 0.0-0.0005,7.0-**
XSPEC>plot ldata
XSPEC>mo zwabs*power+wabs*power
XSPEC> fit
XSPEC> ...
```

Make a nice plot in IPLOT:

```
XSPEC> iplot
```

```
PLT> plot
PLT> label top
PLT> label file
PLT> time off
PLT> lwidth=2
PLT> la x Channel Energy (keV)
PLT> la y Counts s\ u-1\ d keV\ u-1\ d
PLT> plot
```

and create a postscript file:

```
PLT> hardcopy GRB050525.ps/cps
```

The output of the broad-band spectral fitting is shown in Fig. 2.10.

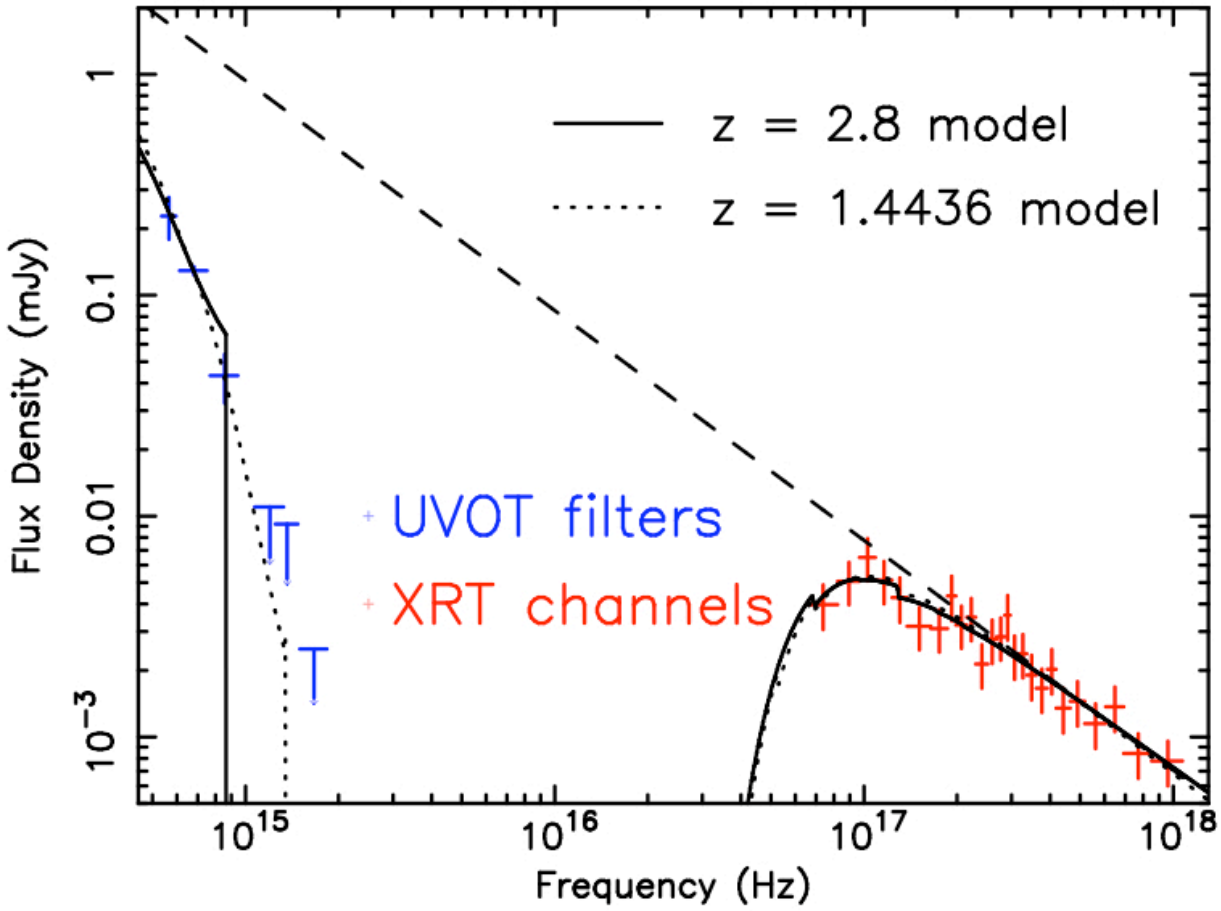

Figure 2.10: Broad-band SED using 6-filter UVOT and XRT data, and best fit spectral models.

Chapter 3

Image Tools

3.1 UVOTBADPIX

Updates

Table 3.1.1: UVOTBADPIX update history.

HEADAS Version	Date	Description of updates
8.0		Released to Swift science team
9.0	2004-10-07	Released for Burst Advocate training

Description

UVOTBADPIX creates bad pixel maps for the images. These are deposited in a level I FITS image file. A bad pixel map is an image of the same dimensions, binning and hardware window as the original data, but populated by flags that describe the quality of each pixel. The meaning of each flag is described in Table 3.1.1.

Most of the bad pixel cases result from chip defects on the instrument detector. Pixels in the Level I images are generally sub-sampled, for instance a pixel binning of 1x1 results in sub-sampling by a factor 8, so it is common to find bad pixels clustered in 8x8 arrays. Dead hardware pixels contain no charge whatsoever. Cold pixels contain charge at a reduced level, resulting in fewer counts than expected. Hot pixels contain more counts than expected, while flickering pixels can be bad during some intervals, while good at others. By the nature of the detector, hot pixels are not expected to occur. Cosmetic pixels are stored in the CALDB within a bad pixel table.

Compression damage may occur if the images were compressed on-board before being transmitted to Earth. The compression algorithm stores the difference between counts in consecutive pixels. If the difference is greater than a certain threshold, defined by flight software, then the pixel value is clipped to reduce the telemetry rate. The telemetry-to-fits software that runs in the Swift pipeline, UVOT2FITS, converts telemetry back to raw counts using reverse-compression, however the software has no means of reconstructing the correct number of counts in clipped pixels. To combat this, UVOTBADPIX compares adjacent pixels. If the difference between them equates to the on-board compression threshold then the second pixel is flagged as damaged in the bad pixel map. The user has the choice of whether to search for compression-damaged pixels or not.

NULL-valued image pixels result from either corrupted or missing telemetry. These are also flagged in the bad pixel map.

It is possible for pixels to suffer from more than one type of badness. In these cases the bad pixel map contains the sum of two or more flags. For example, a bad pixel flag of 136 indicates that a flickering pixel has suffered from compression damage (8+128).

Table 3.1.2: Values for the potential flags within a bad pixel map.

Quality value	Description
0	Good
1	Dead pixel
2	Cold pixel
4	Hot pixel
8	Flickering pixel
128	Compression damaged value
256	NULL value

Input files

UVOTBADPIX requires two input files:

1. A level I FITS image file. This may contain multiple extensions, containing one image each. Each image has raw chip coordinates, RAWX and RAWY, e.g.,

- sw00072901259.012/data/uvot/image/sw00072901259uvvrw.img.gz

2. The caldb product containing a cosmetic bad pixel list, e.g.,

- $CALDB/data/swift/uvot/bcf/badpix/swubadpix20041007v001.fits

Output files

UVOTBADPIX has a single output file:

1. A FITS image file, with an identical extension structure as the input image. Each extension contains an image in RAWX, RAWY coordinates which matches the window size, location and binning of the input image. Each output image is populated with image flags, according to Table 3.1.2. This output file is temporary in nature and not archived by HEASARC.

Parameters

Table 3.1.3 lists the input parameters for UVOTBADPIX. Parantheses indicate that parameters are not compulsory. If these parameters are not specified on the command line, the tool will look-up the current value in the parameter file. Users can do this by typing plist uvotbadpix.

Table 3.1.3: Parameter descriptions for UVOTBADPIX.

Parameter	Description
infile	Name of Level I FITS image file. These will have raw within the file name and reside in the images subdirectory
badpixlist	Name of the badpixel list that resides in the caldb. If the $CALDB environment variable is set, the path and name of the pixel list can be replaced simply with caldb
outfile	The name of the output bad pixel map file.
(compress)	Should UVOTBADPIX search for compression-damaged pixels? The default is yes. Turning this switch off will increase the speed of the tool, but should only be done if the user is certain that the data was telemetered in an uncompressed format
(clobber)	Should UVOTBADPIX overwrite a file with the same name as the output? The default is no
(history)	Should UVOTBADPIX write HISTORY keywords to the output file? This creates a record of the processing performed on the file. The default is yes
(chatter)	Verbosity of the tool (0-5). This parameter control how chatty the tool is (1 = quiet. 5 = noisy). The default is 1

Example

Below we provide a typical invocation of the UVOTBADPIX tool. The example files can be copied from ftp://heasarc.gsfc.nasa.gov/docs/swift/foo/uvotbadpix.

Namibia> uvotbadpix infile=sw00072901259ubb_rw.img badpixlist=caldb outfile=sw00072901259ubb_bd.img compress=y clobber=y chatter=1

Warnings and Errors

3.2 UVOTMODMAP

Updates

Table 3.2.1: UVOTMODMAP update history.

HEADAS Version	Date	Description of updates
8.0		Released to Swift science team
9.0	2004-10-07	Released for Burst Advocate training

Description

All images contain systematic modulo-8 fixed-pattern noise of amplitude ∼50%. This effect is the result of pixel sub-sampling on the detector. Photons passing through the UVOT aperture are amplified by a series of phospher layers and micro-channel plates, so that an individual photon results in a splash of photons (an event), on the CCD detector. Since the event is spread over approximately 3x3 CCD pixels, an on-board algorithm can fit this distribution and centroid the event to sub-pixel resolution. CCD pixels are therefore sub-sampled on-board such that a relatively coarse detector of 256x256 4x4 active pixels yields a 2048x2048 image of 0.5x0.5 pixels.

The on-board centroiding algorithm is relatively simple, whereas the actual event distribution can be variable in both CCD position (due to incident-angle effects) and time (due to variations in detector gain). It is not efficient use of time to consistently calibrate the noise pattern and up-load this regularly to flight software. Therefore Level I FITS images contain modulo-8 fixed pattern noise.

Unfortunately there is no clean method of removing the fixed pattern noise without destroying photometric accuracy. Note that within each 8x8 pixel block, photometry is conserved. Fourier filtering would degrade this accuracy. A further complication arises when sources are bright ($>$ 20 count s-1) and suffer from coincidence losses. Over these cases the fixed-pattern noise is modified and cannot be recovered without a well-calibrated Monte Carlo analysis. The fixed-pattern will be obvious in images around very bright sources as a grid-like effect.

The existing algorithm to help treat the mod-8 problem is identical to the algorithm in the XMM-Newton OM tool kit. It takes each image individually, calculating the mean structure within a sliding cell or group of n 8x8 pixels, using a sigma-clipping technique to reject outliers. In order to retain photometric accuracy, it then resizes individual pixels according to the counts within each and then rebins the entire image to yield a new linear array. An example of this algorithm working on a combined flat field image is provided in Fig. 3.1.

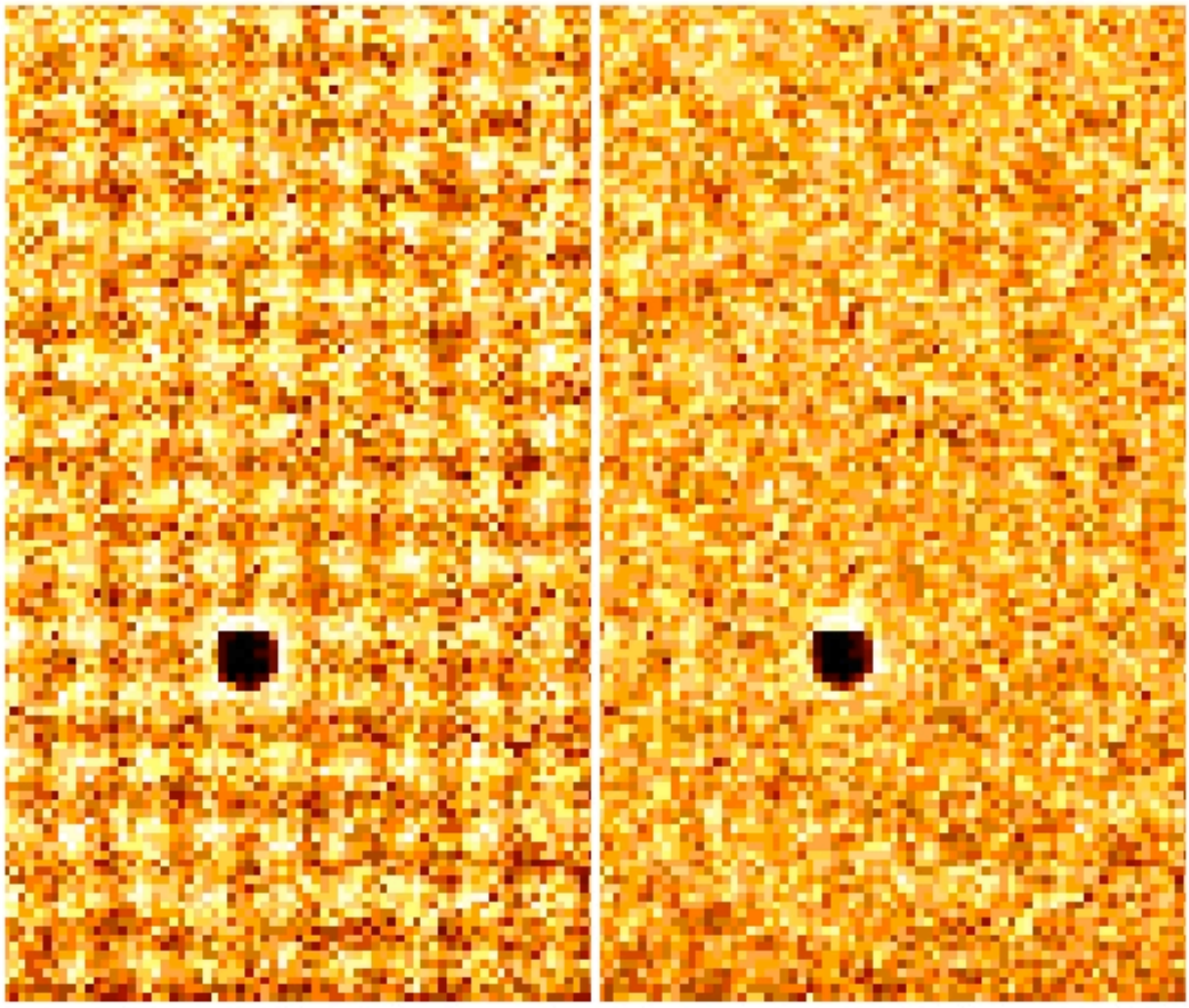

Figure 3.1: A sub-image of many flat fields combined, containing one cosmetically bad pixel, before and after mod-8 fixed pattern noise correction.

The disadvantage of this method is, of course, that it only works correctly on fields that are uniform. In individual images, this means that only the background is useful and point sources are thrown out by sigma-clipping. Since the UVOT background is small, a large amount of pixels have to be averaged in order to provide useful event statistics and this hampers the effort to remove

mod-8 noise on local scales.

Since the primary UVOT science goals are the astrometry and photometry of point sources, fixed-pattern noise will not affect this science in a major way. The one exception is grism analysis, where the mod-8 noise will be obvious in bright spectra.

Since the Swift pipeline is dedicated to processing all spacecraft, BAT, XRT and UVOT data in < 2 hr, mod-8 correction is not performed in the pipeline. UVOTMODMAP is very CPU-intensive, the overhead of running UVOTMODMAP in the pipeline is unacceptable with questionable scientific gain. It is left to the user's taste, whether or not to reprocess archived data with mod-8 correction included.

In the long-term, the correct way to approach this problem is iterative fitting to a Monte Carlo generated model, that ray-traces photons through the detector. This will treat mod-8 noise and coincidence losses in a consistent way, retaining photometric accuracy. However, it is currently not clear whether the fixed-noise pattern will remain stable enough to make this method viable.

Input files

UVOTMODMAP requires two input files:

A level I FITS image file. This may contain multiple extensions, containing one image each. Each image has raw chip coordinates, RAWX and RAWY, e.g.,
sw00072901259.012/data/uvot/image/sw00072901259uvvrw.img.gz

1. The bad pixel maps. A FITS image file, with an identical extension structure as the input image. Each extension contains an image in RAWX, RAWY coordinates which matches the window size, location and binning of the input file. Each output image is populated with image flags, according to Table 3.2.1. This file is a temporary structure generated by UVOTBADPIX.

Output files

UVOTBADPIX has one compulsory output file and one optional output file:

1. An image file, with the same extension structure as the input image file. The new coordinate system in most UVOT cases will be either SKY or DET. This is a temporary file and not archived by the HEASARC.

2. An image file, with the same extension structure as the input image file, but containing the modulo-8 fixed-pattern noise that was calculated from each input image. This is an optional output file, not created during pipeline processing and not archived by the HEASARC.

Parameters

Table 3.2.2 lists the input parameters for UVOTMODMAP. Parantheses indicate that parameters are not compulsory. If these parameters are not specified on the command line, the tool will look-up the current value in the parameter file. Users can inspect the parameter file by typing plist uvotmodmap.

Table 3.2.2: Parameter descriptions for UVOTMODMAP.

Parameter	Description
infile	Name of the input image file. These will probably have multiple image extensions
badpixfile	Name of the file containing the bad pixel maps
outfile	Name of the output image file
(subimage=no)	Process subimage?
(xmin=0)	Subimage x-min
(xmax=2047)	Subimage x-max
(ymin=0)	Subimage y-min
(xmax=2074)	Subimage y-max
(mod8prod)	Should the tool output maps of the modulo-8 fixed-pattern noise? The default is no
(mod8file)	Name of the optional output file containing modulo-8 fixed-pattern noise structure
nsig	Significance level for sigma-clipping. Those pixels with counts above or below this threshold from the mean value in a cell are discarded, and the mean recalculated
ncell	The size of the cell that slides over the image, in units of 8 pixels
(clobber)	Should UVOTMODMAP overwrite a file with the same name as the output? The default is no
(history)	Should UVOTMODMAP write HISTORY keywords to the output file? This creates a record of the processing performed on the file. The default is yes
(chatter)	Verbosity of the tool (0-5). This parameter control how chatty the tool is (1 = quiet. 5 = noisy)

Example

Below we provide a typical invocation of the UVOTMODMAP tool. The example files can be copied from ftp://heasarc.gsfc.nasa.gov/docs/swift/foo/uvotmodmap.

Namibia> uvotmodmap infile=sw00072901259ubb_rw.img
outfile=sw00072901259ubb_md.img badpixfile=sw00072901259ubb_bd.img
mod8prod=n mod8file=foo.fits nsig=3 ncell=16 clobber=y chatter=1 history=y

Warnings and Errors

3.3 UVOTFLATFIELD

Updates

Table 3.3.1: UVOTFLATFIELD update history.

HEADAS Version	Date	Description of updates
8.0		Released to Swift science team
9.0	2004-10-07	Released for Burst Advocate training

Description

Sensitivity variations occur across the detector. Due to the nature of the photon-counting detector, large-scale variations are expected to be negligible. However pixel-to-pixel variations may be significant. These are corrected for by dividing each image through by a calibration image of a flat field. Typically this field will be accumulated from many deep pointing with the point sources removed and the image normalized. The operation of dividing this filed is a trivial one, but this tool provides a labour-saving device to correct the many images from a given FITS file semi-autonomously.

Input files

UVOTFLATFIELD requires two input files:

1. Raw image file. It is optional whether these files have been treated for modulo-8 noise using uvotmodmap or not. Either case is valid. An example file location in the archive is 00072901259/uvot/image/sw00072901259uvv_rw.img.gz.

2. The caldb product containing the flat field image, e.g.,

- $CALDB/data/swift/uvot/bcf/flats/swuppsens20041007v001.fit

Output files

UVOTFLATFIELD has a single output file:

1. An FITS file containing corrected images of the same dimensions, windowing and binning as the input images. These files are temporary and not archived by the HEASARC.

Parameters

Table 3.3.2 lists the input parameters for UVOTFLATFIELD. Parantheses indicate parameters that are not compulsory. If these parameters are not specified on the command line, the tool will look-up the current value in the parameter file. Users can inspect the parameter file by typing plist uvotflatfield.

Table 3.3.2: Parameter descriptions for UVOTFLATFIELD.

Parameter	Description
infile	Name of the input image file
outfile	Name of the output image file
flatfile	Name of the calibration file containing the flat field data. If the CALDB environment variable is set, caldb will point the tool to the most relevant version of the file
(cleanup)	Whether to delete temporary files created during the execution of this tool. The default is yes
(clobber)	Should UVOTFLATFIELD overwrite a file with the same name as the output? The default is no
(history)	Should UVOTFLATFIELD write HISTORY keywords to the output file? This creates a record of the processing performed on the file. The default is yes
(chatter)	Verbosity of the tool (0-5). This parameter control how chatty the tool is (0 = quiet. 5 = noisy)

Example

Below we provide a typical invocation of the UVOTFLATFIELD tool. Example input and output files can be copied from ftp://heasarc.gsfc.nasa.gov/docs/swift/foo/uvotflatfield.

Namibia> uvotflatfield infile=sw00072901259ubb_rw.img
outfile=sw00072901259ubb_ff.img flatfile=caldb cleanup=y clobber=y chatter=1

Warnings and Errors

3.4 UVOTFLUX

Updates

Table 3.8.1: UVOTFLUX update history.

HEADAS Version	Date	Description of updates
8.0		Released to Swift science team
9.0	2004-10-07	Released for Burst Advocate training
HEAsoft 6.0.5	2006-04-26	Added parameters for frame time, source aperture and whether dead time correction has already been performed Write saturated column
HEAsoft 6.3	2007-07-01	Renamed from UVOTMAG to UVOTFLUX

Description

From a column of count rate data contained in a FITS table (and optional error column), UVOTFLUX calculates instrumental magnitudes and fluxes, in units of erg s**-1 cm**-2 Angstrom**1, using filter-specific zero-points and flux conversion coefficients. Zero-points are defined as the magnitude at which the count rate is 1 s**-1. UVOTFLUX will work on a wide variety off FITS tables, such as source lists, light curves and broad-band filter PHA files, provided the input column has units s**-1.

The tool also corrects for dead time and coincidence losses, which occur because the detector can only recognize one photon per detector pixel within each individual frame. The CCD frame rate is 11 ms, hence sources brighter than 20 count s**-1 will suffer from coincidence loss. Obviously the magnitude of this correction is directly related to the source brightness.

Input files

UVOTFLUX requires three input files:

1. A FITS table containing a column of count rates in units of s**-1. Count rate errors are optional, e.g., 00072901259/uvot/image/sw00072901259u.cat.

2. A calibration file containing the zero-points for each filter, e.g., $CALDB/data/swift/uvot/cpf/phot/swuphot20041007v001.fits.

3. A calibration file containing the coefficients of a polynomial fit to tabulated coincidence loss data versus count rate, e.g., $CALDB/data/swift/uvot/bcf/coinc/swucntcor20041007v001.fits.

Output files

UVOTFLUX has a single output file:

1. A FITS table identical to the input file, except for additional columns for instrumental magnitude and flux. Error columns will also be generated if an input error column was provided. The original input file is over-written.

Table 3.8.2: Description of additional columns in the UVOTFLUX output table.

Parameter	Description
MAG	The instrumental magnitude of the source
MAG_ERR	Uncertainty on the instrumental magnitude
FLUX	Source flux in units of erg s**-1 cm**-2 Angstrom**-1
FLUX_ERR	Uncertainty on the source flux in units of erg s**-1 cm**-2 Angstrom**-1

Parameters

Table 3.8.3 lists the input parameters for UVOTFLUX. Parantheses indicate parameters that are not compulsory. If these parameters are not specified on the command line, the tool will look-up the current value in the parameter file. Users can inspect the parameter file by typing plist uvotmag.

Table 3.8.3: Parameter descriptions for UVOTFLUX.

Parameter	Description
infile	FITS table containing a row of count rate data in units of s-1
zerofile	A CALDB file containing filter-dependent zero-points and linear flux conversion coefficients. If the CALDB environment variable is set, caldb will point the tool to the most relevant version of the zero-point file
coinfile	A CALDB file containing coincidence loss correction data
filter	The tool will automatically look for the correct FILTER in the keywords of the image file if this parameter is set to default. However UVOTFLUX is a generic tool and user has the option of passing a filter name. The filter options are u, b, v, uvw1, uvm2, uvw2, white, magnifier, ugrism, and vgrism
(ratecol)	Name of the input table column containing the count rate data
(errcol)	Name of the input table column containing the count rate error data
(history)	Should UVOTFLUX write HISTORY keywords to the output file? This creates a record of the processing performed on the file. The default is yes
(chatter)	Verbosity of the tool (0-5). This parameter control how chatty the tool is (0 = quiet. 5 = noisy)

Example

Below we provide a typical invocation of the UVOTFLUX tool.

Namibia> uvotflux infile=sw00000001001u.cat+1 zerofile=caldb
coinfile=caldb filter=B ratecol=RATE errcol=RATE_ERR chatter=1

Warnings and Errors

3.5 SWIFTXFORM

Updates

Table 3.4.2: SWIFTXFORM update history.

HEADAS Version	Date	Description of updates
8.0		Released to Swift science team
9.0	2004-10-07	Released for Burst Advocate training

Description

SWIFTXFORM is a general Swift tool that converts images from one coordinate system to another. The input coordinate system can be one of the following:

Table 3.4.2: The coordinate systems in UVOT data files.

System	Description
RAW	A linear array, identical to the pixel array within the UVOT camera. The image can be any rectangular sub-array of the detector, with binning of 1, 2, 4 or 8. The array unit is pixels
DET	An array, in physical units of the image window. The array unit is mm. Compared to a RAW image, the array has been flipped across the RAWX axis, so that we look up at the sky, rather than down upon the detector, distortion due to the telescope optics has been corrected for, and rebinned so that the image remains linear
SKY	A tangential projection of the image in ecliptic coordinates, (RA2000, Dec2000). Compared to a DET image, the SKY image has been rotated according to the roll angle of the telescope. The array has been rebinned so that pixels are linear and orthogonal to RA and Dec

Similarly, the output coordinate system can be one of the three alternatives in Table 3.4.2. The UVOT pipeline will only perform the conversion RAW to SKY; the one exception is for grism images where analysis should be performed after a RAW to DET conversion.

By default, SWIFTXFORM calculates the smallest output array possible that will contain the input image. However, the user does have control over the center of the image when converting to SKY, which also determines the size of the output array.

Output image are always linear arrays and this requires pixel rebinning after coordinate transformations. The algorithm calculates the output coordinates of the four corners of each input pixel. Those output pixels that overlap the resulting quadrilateral are flagged. The counts within each input pixel are distributed among the output pixels, weighted according to one of the following schemes listed in Table 3.4.3., which can be chosen by the user.

Table 3.4.3: Count redistribution methods available in SWIFTXFORM

System	Description
AREA	This method treats the counts in each pixel as being spread evenly over the area of the pixel. It distributes the value of each original pixel among the transformed pixels it covers proportionally by the fraction of overlapping area. This method preserves the sum over pixels, so the transformed image can be used to calculate fluxes. The transformed pixels values will not be integers. The argument bitpix=-32 must always be used with this option
CENTER	This method assumes the counts in each pixel are concentrated at the center. For each pixel in the original image, it transforms the position of the pixel center and then adds the full pixel value to the corresponding pixel in the transformed image. This method is fast to calculate, and it preserves the sum over pixels and the integer nature of the original image. However, it produces artifacts when the original and transformed pixel grids do not coincide. So this method should only be used for e.g. translations by an integer number of pixels
EVENT	This method mimics the effect of the COORDINATOR FTOOL on an event list. It assumes that the value of each pixel gives the number of events in that pixel. It then assigns random positions for each event within the pixel and transforms those positions to the new coordinate system. Then the events are binned into the pixels of the transformed image. If the pixel values are not integers, imagexform gives a warning and converts the values to integers in an arbitrary way. This method preserves the total (integer) number of counts in the image, and guarantees that the transformed pixel values will all be integers. This method is best for "counts" images
INTERPOLATE	This method linearly interpolates the input image. For each pixel in the transformed image it calculates the position of the center of that pixel in the original image. It then linearly interpolates the original image to get the transformed pixel value. The output pixel values will not generally be integers. This is a common technique for transforming terrestrial images, but the sum of the pixels is not preserved, so fluxes derived from the transformed image will be inaccurate
DEFAULT	In the UVOT case, this method is identical to AREA

Data containing the pixel sizes, reference telescope coordinates, optical distortion parameters and rotation and flip conventions are stored in the Telescope Definition (TELDEF) file that resides in the caldb. Optical distortion is stored as an array of RAW X and Y offset parameters, sampled across the detector. Corrections for each pixel location are interpolated in two dimensions using these offsets.

Input files

SWIFTXFORM requires three input files:

1. An input image file.

- Example of archive location:
  00072901259/uvot/image/sw00072901259uvv_rw.img.gz

2. An attitude history file, which is a tabular record of the spacecrafts pointing and orientation during the observation, generally with a time resolution of 10 s.

- Example of archive location:
  00072901259/auxil/sw00072901259sat.fits.gz

3. A telescope definition file. In order to convert from one coordinate system to another, the tool requires telescope data such as boresight direction, focal length, pixel size and distortion map. These quantities are contained within a single CALDB file.

- Example of caldb location:
  $CALDB/data/swift/uvot/bcf/teldef/swugen20041007.teldef

Output files

SWIFTXFORM yields a single output file:

1. A FITS file containing one or more image extensions.

- Archive location example:
  00072901259/uvot/image/sw00072901259uvv_sk.img.gz

Parameters

Table 3.4.4 lists the input parameters for SWIFTXFORM. Parantheses indicate parameters that are not compulsory. If these parameters are not specified on the command line, the tool will look-up the current value in the parameter file. Users can inspect the parameter file by typing plist swiftxform.

Table 3.4.4: Parameter descriptions for SWIFTXFORM.

Parameter	Description
infile	Name of the input FITS image file. The input file will generally have more than one image extension. SWIFTXFORM will perform a transformation on a single extension if the extension name is specified in this parameter, e.g., sw00073422001.img[00148265E]. Alternatively, all extensions will be processed if only the filename is specified
outfile	Name of the output FITS image file. This can have multiple image extensions
attfile	Name of the input attitude history file. This will reside in the att subdirectory of archived data
method	Interpolation method during the transformation. The options are: AREA, CENTER, EVENTS, INTENSITY and DEFAULT. The recommended method is AREA. Definitions for these methods are provided in Table 3.4.3
teldeffile	Name of the input telescope definition file. Entering caldb for this parameter will allow the tool to find the most appropriate version of the alignment file in the Swift CALDB
to	The coordinate system of the output file, RAW, DET or SKY
ra	For a SKY output image, the RA(2000) coordinate of the center of the array. This parameter is redundant for other types of transformation. Units are decimal degrees
dec	For a SKY output image, the Dec(2000) coordinate of the center of the array. This parameter is redundant for other types of transformation. Units are decimal degrees
(bitpix)	The number of bits stored in the output image. The default is to write the same type of array to output as the input array. This argument should be used when this is not a desired feature. The recommended value is 32 for most applications, corresponding to an integer array of single precision. The one exception to this rule occurs when method=AREA and the output array must be real, i.e. bitpix=-32
(seed)	Random number seed for the EVENT interpolation method
(aberration)	Whether velocity aberration is included in a translation to or from SKY coordinates. The default is no since velocity aberration is compensated for by on-board software before the attitude data is transmitted
(zeronulls)	Whether to treat NULL pixel values as zero.The default is no
(copyall)	Copy all image extensions over to the output file. This is only applicable when a specific image extension is requested as input. The default is no
(clobber)	Should SWIFTXFORM overwrite a file with the same name as the output? The default is no
(history)	Should SWIFTXFORM write HISTORY keywords to the output file? This creates a record of the processing performed on the file. The default is yes
(cleanup)	SWIFTXFORM is a perl script that stitches several base tools together. The default, yes, indicates that the tool will delete temporary files created during execution
(chatter)	Verbosity of the tool (0-5). This parameter control how chatty the tool is (0 = quiet. 5 = noisy)

Example

Below we provide a typical invocation of the SWIFTXFORM tool. Example input and output files can be copied from ftp://heasarc.gsfc.nasa.gov/docs/swift/foo/swiftxform.

Namibia> swiftxform infile=sw00072901259uvv_rw.img \
outfile=sw00072901259uvv_sk.img attfile=sw00072901259sat.fits \
method=AREA to=SKY ra=299.67 dec=35.235 teldeffile=caldb bitpix=-32 \
clobber=y chatter=1 cleanup=y history=y

Warnings and Errors

3.6 UVOTEXPMAP

Updates

Table 3.5.1: UVOTEXPMAP update history.

HEADAS Version	Date	Description of updates
8.0		Released to Swift science team
9.0	2004-10-07	Released for Burst Advocate training
HEAsoft 6.0.5	2006-04-26	Updated how RAW to SKY transform is determined
HEAsoft 6.4	2008-03-01	"Method" parameter revised

Description

UVOTEXPMAP creates exposure maps for each UVOT image in SKY coordinates. Telescope pointing will vary by some degree during, and between, exposures. Exposure maps are arrays of the same size as SKY images populated with effective exposure times smaller or equal than the exposure integration time. They are therefore useful when analyzing sources near bad pixels or window edges where telescope drift will reduce the effective exposure times of individual pixels. Note that UVOT does not suffer from vignetting, so count rates will, in general, be linear across the detector.

Given the exposure times of individual images, at its most basic, UVOTEXMAP takes a bad pixel map, sets good pixels to 1 and bad pixels to 0, multiplies the image by the exposure time and finally performs a transformation from RAW to SKY coordinates. Each pixel contains the interval, in units of seconds, that it was effective during the exposure.

If an image has been constructed from event-like data, it is likely that photon positions have been corrected to their true sky locations using attitude history data or similar diagnostics of the drift in spacecraft pointing. In these cases an identical correction must be made to the exposure map. A large number of intermediate exposure maps are created for each individual exposure, one map for each time quanta used to correct photon positions. The position (and roll if available) of each map is defined by the mean telescope pointing during that time interval. The final exposure map is constructed by co-adding the series of maps and rebinning so that the final product contains the same pointing and binning as the parent image. If consecutive pointing entries in the drift data differ by a user-defined threshold, the tool will linearly interpolate new points in the table so that the threshold is never exceeded.

The UVOT approach to exposure map creation will depend upon the data format. For standard imaging mode the process is relatively simple and fast. In this case, each image pixel has been exposed for equal amounts of time so no delicate calculations of aspect drift corrections around window edges and bad pixels are required. However, if spacecraft pointing stability is found to be poor, a shift-and-add algorithm will be implemented by on-board software, where photon arrival locations will be shifted in the RAWX and RAWY dimensions according to the attitude data at 10s intervals before the image is summed. In this case UVOTEXPMAP must recreate these on-board

shifts when creating the exposure map. In this mode, the spacecraft will send down an aspect report packet containing the shifts employed during this procedure. UVOTEXPMAP requires this data in order to recreate the correction in the exposure maps. Finally exposure maps for images constructed from event data can also be constructed using position corrections from attitude history data provided the same data was used to calculate the sky position of each photon. These last two cases are time consuming and CPU intensive tasks. Since the pipeline processing must complete within 2 hours, it remains to be seen whether these steps can be used by the pipeline.

Input files

UVOTEXPMAP requires four input files:

1. A Level II FITS input file, containing SKY coordinate images, e.g., 00072901259/uvot/image/sw00072901259uvv_sk.img.gz

2. A bad pixel file containing image extensions that correspond to the input images. This is a FITS image file, with an identical extension structure as the Level II sky image. Each extension contains an image in RAWX, RAWY coordinates. This file is a temporary structure generated by UVOTBADPIX.

3. Am aspect report packet or attitude history file which contains a tabular record of the spacecrafts pointing and orientation during the observation. , e.g.,

- 00072901259/auxil/image/sw00072901259sat.fits.gz

4. A telescope definition file. In order to convert from one coordinate system to another, the tool requires telescope data such as boresight direction, focal length, pixel size and distortion map. These quantities are contained within a single CALDB file.

- Example of caldb location: $CALDB/data/swift/uvot/bcf/teldef/swugen20041007.teldef

Output files

UVOTEXPMAP has a single output file:

An output image array, containing exposure map image extensions in SKY coordinates. Image sizes, positions and file extension structure are identical to the input image.

Parameters

Table 3.5.2 lists the input parameters for UVOTEXPMAP. Parantheses indicate parameters that are not compulsory. If these parameters are not specified on the command line, the tool will look-up the current value in the parameter file. Users can inspect the parameter file by typing plist uvotexpmap.

Table 3.5.2: Parameter descriptions for UVOTEXPMAP.

Parameter	Description
Infile	Name of the Level II FITS image file, containing images in SKY coordinates. Files may contain more than one image extension. This file is produced by SWIFTXFORM
badpixfile	Name of the file containing the bad pixel maps that correspond to the input images. This file is produced by UVOTBADPIX
attfile	Name of the attitude history file, contained in the att directory in archived data
teldeffile	Name of the telescope definition file contained in the CALDB. If the CALDB environment variable is set, caldb will point the tool to the most relevant version of the teldef file. This file contains telescope data such as an optical distortion map, focal length, boresight and pixel size
outfile	The name of the FITS image file containing exposure maps corresponding to the input images. This file will have an identical structure to the input image file
method	This parameter defines the method of map construction used by the tool. MEANFOV should be used for standard images, SHIFTADD for images resulting from the onboard shift-and-add process.
attdelta	If consecutive position records in the attitude history file differ by more than this parameter, virtual records are interpolated linearly between them so that the difference between adjacent records is no more than attdelta. The unit is arcsec
(aberration)	Account for relativistic velocity aberration of source positions. This correction is made to attitude data on-board, so should never be required. The default is no
(cleanup)	UVOTEXPMAP is a perl script that wraps several base tools. This parameter tells the tool to delete all intermediate files in the working directory. The default is yes
(clobber)	Should UVOTEXPMAP overwrite a file with the same name as the output? The default is no
(history)	Should UVOTEXPMAP write HISTORY keywords to the output file? This creates a record of the processing performed on the file. The default is yes
(chatter)	Verbosity of the tool (0-5). This parameter control how chatty the tool is (0 = quiet. 5 = noisy)

Example

Below we provide a typical invocation of the uvotexpmap tool. Example input and output files can be copied from ftp://heasarc.gsfc.nasa.gov/docs/swift/foo/uvotexpmap.

Namibia> uvotexpmap infile=sw00072901259uvv_sk.img
outfile=sw00072901259uvv_ex.img badpixfile=sw00072901259uvv_bd.img
teldeffile=caldb attfile=sw00072901259sat.fits method=MEANFOV clobber=y
chatter=1 cleanup=y history=y clobber=y

Warnings and Errors

3.7 UVOTDETECT

Updates

Table 3.6.1: UVOTDETECT update history.

HEADAS Version	Date	Description of updates
8.0		Released to Swift science team
9.0	2004-10-07	Released for Burst Advocate training
HEAsoft 6.0.5	2006-04-26	Updates to SExtractor parameters. Corrected background mean and sigma estimates. Fixed RATE_ERRs when an exposure map is provided

Description

UVOTDETECT detects sources within a single UVOT SKY image and constructs a source table. This tool is a script wrapped around the public package sextractor. A homepage for this software, containing a description of the algorithm, may be found at:

http://terapix.iap.fr/rubrique.php?id_rubrique=91

The choice of this tool was made because of sextractors capability to detect and measure extended sources. This script records a subset of the sextractor output in the resulting FITS table, including source positions in both epoch 2000 sky coordinates and pixel coordinates, the faint detection threshold, the size, shape and orientation of extended sources (assuming elliptical distributions), and warning flags from the tool.

Input files

UVOTDETECT requires one mandatory input file and one optional one:

1. A level II input FITS file, containing one or more SKY coordinate images. Only the specified image extension will be operated upon.

- Archive location, e.g.:
  00072901259/uvot/image/sw00072901259uvv_sk.img.gz

2. An optional image file containing an array of weights for each pixel in the input image. This is used to determine more accurate detection thresholds for each source. The exposure map appropriate for the input image is a viable weight map.

- Archive location, e,g,:
  00072901259/uvot/image/sw00072901259uvv_ex.img.gz

Output files

UVOTDETECT has a single output file:

1. A FITS table containing one row for every source detected in the input image. Table 3.6.2 describes the content of each row.

Table 3.6.2: Description of the output table from UVOTDETECT.

Column	Description
REFID	Internal reference number of each source, 1-n
RA	Epoch 2000 right ascension of the source in decimal degrees
DEC	Epoch 2000 declination of the source in decimal degrees
RA_ERR	1-sigma uncertainty in RA
DEC_ERR	1-sigma uncertainty in Dec
THRESHOLD	Detection threshold in magnitude. This parameter is related to the threshold input parameter
X_IMAGE	x-position of the source in the input image in pixels
Y_IMAGE	y-position of the source in the input image in pixels
X_ERR	1-sigma uncertainty in X_IMAGE
Y_ERR	1-sigma uncertainty in Y_IMAGE
PROF_MAJOR	Size of semi-major axis of source in arcsec
PROF_MINOR	Size of semi-minor axis of source in arcsec
PROF_THETA	Angle subtending the major axis and equator in degrees counter-clockwize
FLAGS	Warning flags generated by sextractor
ORIGIN	Origin of source list, e.g., image, TDRSS packet

Quoting directly from Sec 8.1 of the sextractor users guide, the origin of warning flags will be one of the following:

Table 3.6.3: Key to warning flags in the UVOTDETECT source table.

Flag	Description
1	The object has neighbours, bright and close enough to significantly bias photometry, or bad pixels where more than 10% of the integrated area is affected
2	The object was originally blended with another one
4	At least one pixel of the object is saturated, or very close it
8	The object is truncated, i.e. too close to the image boundary
16	Data in the extraction aperture is incomplete or corrupted
32	Objects isophotal data is corrupted or incomplete
64	A memory overflow occurred during deblending
128	A memory overflow error occurred during extraction

Note that more than one warning could be given for a source, hence FLAGS = 8+16+32 = 56 is a possible value.

Parameters

Table 3.6.4 lists the input parameters for UVOTDETECT. Parantheses indicate parameters that are not compulsory. If these parameters are not specified on the command line, the tool will look-up the current value in the parameter file. Users can inspect the parameter file by typing plist uvotdetect.

Table 3.6.4: Parameter descriptions for UVOTDETECT.

Parameter	Description
infile	Input level II image FITS file. Only one image is accepted, so if the file has multiple image extensions, the correct extension should be provided
outfile	Output FITS source table
weightfile	An image extension containing exposure weights for each pixel. none is the default, but an accurate exposure map should be supplied here, if the input image is a mosaic or sum of several individual exposures
threshold	Detection threshold above the background in terms of signal-to-noise. Potential sources detected below this threshold will not be written to the output table
(cleanup)	UVOTDETECT is a wrapper script. This parameter determines whether intermediate files are deleted from the working directory. The default is yes
(clobber)	Should UVOTDETECT overwrite a file with the same name as the output? The default is no
(history)	Should UVOTDETECT write HISTORY keywords to the output file? This creates a record of the processing performed on the file. The default is yes
(cleanup)	UVOTDETECT is a perl script that wraps an external program. In doing so, several parameter control files need to be created before executing sextractor. This parameter tells the tool to delete the control files in the working directory. The default is yes
(chatter)	Verbosity of the tool (0-5). This parameter control how chatty the tool is (0 = quiet. 5 = noisy)

Example

Below we provide a typical invocation of the UVOTDETECT tool. Example input and output files can be copied from ftp://heasarc.gsfc.nasa.gov/docs/swift/foo/uvotdetect.

Namibia> uvotdetect infile=sw00000001001u_sk.img+1 outfile=sources.fit weightfile=sw00000001001u_ex.img+1 threshold=2 history=y cleanup=y clobber=y chatter=1

Warnings and Errors

3.8 UVOT2PHA

Updates

Table 3.7.1: UVOT2PHA update history.

HEADAS Version	Date	Description of updates
8.0		Released to Swift science team
9.0	2004-10-07	Released for Burst Advocate training

Description

Swift is a panchromatic instrument consisting of three detectors, BAT, XRT and UVOT. It will often be critical to analyze data from two or three of these instruments simultaneously and consistently. We therefore provide this tool that constructs data files from the UVOT images that are compatible with XSPEC, the X-ray spectral analysis package which both XRT and BAT data conform to. It then becomes possible to perform panchromatic spectral fitting across the optical, UV, soft X-ray

and hard X-ray bands simultaneously. XSPEC is incorporated in the XANADU subpackage of the HEASoft distribution which is required before attempting any Swift data analysis, therefore it should already exist on your machine. A description of the XSPEC package can be found at:

http://heasarc.gsfc.nasa.gov/docs/xanadu/xspec/index.html

UVOT2PHA is only applicable to images derived from the UVOTs lenticular (broad-band) filters, i.e., U, B, V, UVW1, UVM2, UVW2, WHITE and MAGNIFIER. The tasks for converting grism image data to XSPEC compatible spectra are contained in UVOTIMGRISM and UVOTRMFGEN. Of course, other than a central wavelength, data from an individual lenticular filter contains no spectral information. However, by combining two or more lenticular filter data files, or one or more lenticular filter file with XRT or BAT files, spectral analysis may be performed in XSPEC such as spectral slope and redshift measurements. These can also be fit simultaneously with any XRT or BAT data available. Note this one word of caution, UVOT exposures are taken in series, not parallel, so the user must understand the potential of time-variable phenomena biasing the results of spectral fitting.

The core of this tool is inherently simple. Given two region files, one containing source counts from a specific object, the other containing background counts from around that source, UVOT2PHA will extract counts from both regions accompanied by Poisson uncertainties. These four quantities will be cast into two XSPEC-compatible files. The specific file format is documented at the HEASARC:

http://heasarc.gsfc.nasa.gov/docs/heasarc/ofwg/docs/summary/ogip_92_007_summary.html

The extraction regions should be stored in two external ascii files prior to executing this tool. These can be created by hand, or more conveniently, using standard image analysis tools such as DS9:

http://hea-www.harvard.edu/RD/ds9

or XIMAGE:

http://heasarc.gsfc.nasa.gov/docs/xanadu/ximage/ximage.html

XIMAGE is also part of the XANADU package and will be installed on your machine at the same time as the rest of the HEASoft distribution. Note that when using DS9 to create these extraction region files the user must ensure that they are written in image pixel coordinates rather than RA and Dec coordinates. This is the tool default, but it can be verified by clicking on the region button on the upmost tool bar above the image. The data extraction itself is executed by an internal call to XIMAGE, using the command counts/regionfile.

While the output from UVOT2PHA is XSPEC-ready, it is not scientifically useful without a suitable response matrix, which defines the spectral properties of the data. These can be downloaded from the Swift web pages, where there is one for every lenticular filter. It is critical that the correct response matrix be used with the data:

http://swift.gsfc.nasa.gov/docs/swift/proposals/swift_responses.html

UVOT2PHA propagates required keywords from the image header extension to the spectrum files. However if the user attempts to execute this tool using image files that do not contain these mandatory keywords then he or she is required to supply suitable keywords values manually using the optional parameters ra, dec, date-obs, time-obs, date-end and time-end described in Table 3.7.2.

UVOT2PHA is a value-added tool and not employed in the automated Swift data reduction pipeline.

Input files

UVOT2PHA requires three input files:

1. A specific extension within a level II UVOT FITS image file, e.g., 00072901259/uvot/image/sw00072901259uvv_sk.img+1.

2. An ascii file containing the source extraction region using WCS sky coordinates. A simple example would be:

- fk5;circle(273.7256,-23.9964,12)

  the coordinates are relative to the lower-left pixel in the image.

3. An ascii file containing the background extraction region, e.g.:

- fk5;circle(273.7256,-23.9964,24)

  fk5;-circle(273.7256,-23.9964,12)

Output files

UVOT2PHA has two output files:

1. An XSPEC-compatible FITS file containing source counts.
2. An XSPEC-compatible FITS file containing background counts.

Parameters

Table 3.7.2 lists the input parameters for UVOT2PHA. Parantheses indicate parameters that are not compulsory. If these parameters are not specified on the command line, the tool will look-up the current value in the parameter file. Users can inspect the parameter file by typing plist uvot2pha.

Table 3.7.2: Parameter descriptions for UVOT2PHA.

Parameter	Description
Infile	Input image file or extension
Srcpha	Output pha file containing the source + background count rate
bkgpha	Output pha file containing the background count rate
srcreg	Region file defining the source extraction area
bkgreg	Region file defining the background extraction area
(phatype)	Type of output data. The options are counts, providing an output table containing counts, or rate, providing an output table containing data in units of count s-1. The rate option is essential if the user intends to run UVOTMAG on the output from this tool
(ra)	The Right Ascension (epoch 2000) of the source in decimal degrees
(dec)	The Declination (epoch 2000) of the source in decimal degrees
(date_obs)	Date at the start of the observation
(time_obs)	The observation start time in MET
(date_end)	Date at the end of the observation
(time_end)	The observation end time in MET
(tmpdir)	Name of the directory to store temporary files. The default is .
(clobber)	Should UVOT2PHA overwrite a file with the same name as the output? The default is no
(history)	Should UVOT2PHA write HISTORY keywords to the output file? This creates a record of the processing performed on the file. The default is yes
(chatter)	Verbosity of the tool (0-5). This parameter control how chatty the tool is (0 = quiet. 5 = noisy)

Example

Below we provide a typical invocation of the UVOT2PHA tool.

Namibia> uvot2pha infile=sw00072901259uvv_sk.img+1 srcpha=v.pha bkgpha=b_bkg.pha srcreg=v.reg bkgreg=v_bkg.reg clobber=y chatter=1

Warnings and Errors

3.9 UVOTMAG

Updates

Table 3.8.1: UVOTMAG update history.

HEADAS Version	Date	Description of updates
8.0		Released to Swift science team
9.0	2004-10-07	Released for Burst Advocate training
HEAsoft 6.0.5	2006-04-26	Added parameters for frame time, source aperture and whether dead time correction has already been performed Write saturated column
HEAsoft 6.3	2007-07-01	Renamed from UVOTMAG to UVOTFLUX

Description

This tool was renamed UVOTFLUX and is obsolete.

3.10 UVOTSOURCE

Updates

Table 3.9.1: UVOTSOURCE update history.

HEAsoft Version	Date	Description of updates
6.0.1	2005-04-01	Released to public
6.0.2	2005-08-11	Flux and count rate output option added. Detection significance calculated. Temporary file handling recoded
6.0.5	2006-04-26	Added output in mJy Changed default value of output parameter to ALL
6.2.1	2007-06-18	Revision includes aperture correction and coincidence loss correction

Description

This tool performs aperture photometry on a single source in a UVOT SKY exposure (sw*.img+extension). It returns information about the count rate from the source, the source's magnitude, and flux density information, corrected for the employed aperture and coincidence losses.

The user specifies the source extraction region and background region using region files that are in the standard ftool or DS9 format. If "srcreg" is set to NONE then the tool will compute the "sigma"-sigma limiting magnitude for the exposure. The "sigma" parameter tells the tool what level of significance to use to compute the background limit. Photometric and coincidence loss calibration data are read from the files specifed by "zerofile" and "coinfile" respectively. The special value CALDB tells the tool to obtain this data from the Swift/UVOT calibration database.

There are two methods for doing photometry. APERTURE does simple aperture photometry. All of the counts in the "srcreg" region are summed and divided by the exposure time to produce a count rate. The background count rate is subtracted and the magnitude is computed from the coincidence-corrected net count rate. Photometry is done as follows.

1. Extract the raw counts in three apertures.

   The user-supplied source region, "srcreg".

   The source aperture may be any size or shape that can be described in a valid region file. The source region should be selected to maximize the science return. In general faint point sources should use circular apertures with a radius that maximizes the signal-to-noise ratio in the aperture. This is typically about 3 arcsec. For bright sources the standard photometric aperture (defined in "zerofile") is usually prefered. The source aperture should be chosen to minimize contamination from other sources.

   The user-supplied background region, "bkgreg".
   The background region may be any size or shape that can be described in a valid region file. It should be chosen to have the same background properties as the source region. It should be free of contaminating sources and large enough to that the mean pixel value is not biased by Poisson statistics. The background value is computed by taking the mean of the pixel values in the background region.

   A coincidence-loss correction aperture, defined in "coinfile"

This region is a circular aperture with an radius defined by the COIAPT column in the COINCIDENCE extension of the "coinfile" file. It is centred on the centre of the "srcreg" region. This is the region that is used to compute the coincidence loss correction factor. Counts are extracted from the input exposure by the XImage "counts" command. See the XImage User Manual for details on how counts are extracted.

2. Calculate the coincidence loss correction factor from the count rate in the coincidence-loss correction aperture. See the fhelp for uvotcoincidence for details about coincidence-loss corrections.

3. Apply the coincidence-loss factor to the raw count rate in the source aperture.

4. Scale the background count rate to the area of the coincidence-loss aperture and then apply the coincidence loss factor.

5. Scale the coincidence-corrected background rate to the area of the user-supplied source aperture and subtract this from the coincidence-corrected rate in the source aperture to get the coincidence-corrected net count rate from the source.

6. Apply the photometric calibrations from "zerofile" to the coincidence-corrected net count rate from the source to obtain the magnitude and flux density information described below. Note that the photometric calibrations assume that the source region is the same as the standard photometric aperture, so the values returned using the APERTURE method are not on the standard UVOT photometric system. See the description of the CURVEOFGROWTH method for details on how to correct for this.

The CURVEOFGROWTH method does aperture photometry as for the APERTURE method with one additional step. The APERTURE method will only return magnitudes and flux densities that are on the standard UVOT photometric system if the source region is the same as the standard photometric aperture defined in "zerofile". To bring these magnitudes onto the standard system the CURVEOFGROWTH method computes an aperture correction to the coincidence-loss corrected net count rate from the source. See the fhelp for uvotapercorr for details on aperture corrections and how they are applied. The CURVEOFGROWTH method assumes that the source is a point source.

The aperture correction applied by the CURVEOFGROWTH method is approximate and intended for preliminary data analysis. It does not take into account changes in the PSF due to voltage or count rate variations (see the uvotapercorr fhelp). For high-precision photometry the APERTURE method should be used, and aperture corrections that take these factors into account should be performed by the user.

Input files

UVOTSOURCE requires three input files:

1. A FITS file containing at least one image extension, e.g., sw00072901259/uvot/image/sw00072901259uvv_sk.img.

2. An ascii file defining the source extraction region. Region files created using either ds9 or ximage are compatible with this tool.

3. An ascii file defining the background region.

Output

Uvotsource returns the following information:

Source Information: The position is the position specified in the "srcreg" region file. The exposure time is the value of the EXPOSURE keyword.

Magnitude Information: The magnitude of the source, its one-sigma statistical error, and the significance of the detection. Background is the magnitude of the sky. Background-limit is the "sigma"-sigma limiting magnitude of the exposure. Coincidence-limit is the magnitude corresponding to a count rate of one count per frame time.

Flux Information: The flux density information is given in cgs units. Flux densities are computed assuming a mean spectrum taken "zerofile". They do not reflect the actual spectrum of the source.

Coincidence Corrected Rate Information: This is the count rate of the source after all corrections have been applied. This includes aperture corrections if the CURVEOFGROWTH method was used.

Raw Rate Information: This is the raw count rate from the source after subtracting the background count rate, but before applying coincidence or aperture corrections.

Flux mJy: The flux density information is given in milliJansky. Flux densities are computed assuming a mean spectrum taken from "zerofile". They do not reflect the actual spectrum of the source.

The errors in the computed quantities are the 1-sigma statistical errors based on Poisson statistics in the count rates. The computed quantities are appended to the FITS file specified by "outfile&quot. If "syserr=YES" then the systematic errors in the calibration are added in quadrature to the statistical errors.

Parameters

Table 3.9.2 lists the input parameters for UVOTSOURCE. Parantheses indicate parameters that are not compulsory. If these parameters are not specified on the command line, the tool will look-up the current value in the parameter file. Users can inspect the parameter file by typing plist uvotsource.

Table 3.9.2: Parameter descriptions for UVOTSOURCE.

Parameter	Description
image	FITS file containing at least one image extension
srcreg	ds9 or ximage ascii region file defining the source extraction region
bkreg	ds9 or ximage ascii region file defining the background extraction region
filter	The tool will automatically look for the correct FILTER in the keywords of the image file if this parameter is set to default. However UVOTSOURCE is a generic tool and a user has the option of passing a filter name. The filter options are u, b, v, uvw1, uvm2, uvw2, white, magnifier, ugrism, and vgrism
sigma	The number of sigma above image noise that corresponds to a secure detection
outfile	Output FITS file to append results to.
(coinfile)	Coincidence loss correction file. The special value CALDB indicates to read from the calibration data base.
(framtime)	If uvotsource is invoked with frametime=default and FRAMTIME is not present in the header, uvotsource assumes frametime = 0.0110322 [s].
(syserr)	Are systematic errors in the photometric calibration to be used in the error calculations. If set to YES then systematic errors are added in quadrature to the statistical errors. If set to NO then only statistical errors are returned.
(method)	Magnitude calculation method. APERTURE or CURVEOFGROWTH. CURVEOFGROWN assumes a point source.
(output)	Output units. The options are instrumental filter magnitude, flux density in erg/s/cm$\hat{}$2/Angstrom, corrected count rate in counts/s, raw count rate in counts/s, flux density in milliJanskys. ALL indicate that all sets of units are output.
(cleanup)	Whether to delete temporary files. Valid answers are y or n
(chatter)	Verbosity of the tool (0-5). This parameter control how chatty the tool is (0 = quiet. 5 = noisy)

Example

Below we provide a typical invocation of the UVOTSOURCE tool.

The following examples illustrate running uvotsource

1. run uvotsource prompting for all mandatory options:
uvotsource

2. run uvotsource specifying all arguments:
uvotsource image=sky.img.gz+1 srcreg=src.reg bkgreg=bkg.reg sigma=5 \
zerofile=CALDB coinfile=CALDB syserr=NO method=APERTURE output=ALL outfile=sources.fits
cleanup=yes clobber=no chatter=1

3. run uvotsource with the curve of growth method and include systematic errors:
uvotsource image=sky.img.gz+1 srcreg=src.reg bkgreg=bkg.reg sigma=5 \
syserr=YES method=CURVEOFGROWTH outfile=sources.fits

Warnings and Errors

3.11 UVOTMAGHIST

Updates

Table 3.10.1: UVOTMAGHIST update history.

HEADAS Version	Date	Description of updates
8.0		Released to Swift science team
9.0	2004-10-07	Released for Burst Advocate training
HEAsoft 6.0.5	2006-04-26	Added frametime parameter
HEAsoft 6.2.1	2007-06-16	Revision

Description

This tool is a wrapper that calls UVOTSOURCE once per image extansion, makes a FITS table and generates light curve plot(s). The fact that UVOTMAGHIST is now a wrapper that calls UVOTSOURCE is a CHANGE compared to previous versions. The purpose of this change is to remove duplicate functionality in the two tools. It is intended as a labour-saving script for the analysis of UVOT image mode data.

In general, provided a user has suitable region files for a specific object, the extraction of count rates from an image is a trivial exercise using, e.g. the counts/regionfile command in the HEASoft XIMAGE, while the conversion of these count rates to instrumental magnitude and flux is provided by a single call to UVOTMAG. With the large number of images produced by the UVOT, this task becomes both arduous and repetitive. UVOTMAGHIST is a script that performs an identical region extraction over all of the images contained in either a Level I, II or II image file and converts each count rate to magnitude and flux. Output files are a FITS table containing the magnitude history of the source, and an optional GIF file containing the filter magnitude against time.

Input files

UVOTMAGHIST has three input files:

1. An FITS file containing a series of image extensions, e.g.,
sw00072901259/uvot/image/sw00072901259uvv_sk.img.gz

2. A calibration file containing the zero-points for each filter, e.g.,
$CALDB/data/swift/uvot/cpf/phot/swuphot20041007v001.fits.

3. A calibration file containing the coefficients of a polynomial fit to tabulated coincidence loss data versus count rate, e.g.,
$CALDB/data/swift/uvot/bcf/coinc/swucntcor20041007v001.fits.

Output files

UVOTMAGHIST has two output files:

1. An optional GIF file plotting instrumental magnitude against time.

2. A FITS table containing information on time, count rate, magnitude, flux, etc., for each of the images in the input file. Definitions for the table columns are provided in Table 3.10.2.

Table 3.10.2: Column definitions for UVOTMAGHIST output.

Col	Name	Format [Units]	Definition
1	MET	D [seconds]	Mission time
2	EXTNAME	12A	Image identifier
3	TSTART	D [seconds]	Image start time
4	TSTOP	D [seconds]	Image stop time
5	EXPOSURE	E [seconds]	Corrected exposure time
6	TELAPSE	E [seconds]	TSTOP - TSTART
7	TIME	D [seconds]	Offset from TIMEZERO
8	SRC_AREA	E [arcsec2]	Area of source extraction region
9	BKG_AREA	E [arcsec2]	Area of background region
10	STD_AREA	E [arcsec2]	Area of CoI aperture
11	PLATE_SCALE	E [arcsec/pix]	Plate scale
12	RAW_TOT_CNTS	E [count]	Total counts in source region
13	RAW_TOT_CNTS_ERR	E [count]	Error in RAW_TOT_CNTS
14	RAW_BKG_CNTS	E [count]	Total counts in background region
15	RAW_BKG_CNTS_ERR	E [count]	Error in RAW_BKG_CNTS
16	RAW_STD_CNTS	E [count]	Total counts in standard region
17	RAW_STD_CNTS_ERR	E [count]	Error in RAW_STD_CNTS
18	RAW_TOT_RATE	E [count/s]	Measured count rate in source region
19	RAW_TOT_RATE_ERR	E [count/s]	Error in RATE_TOT_RATE
20	RAW_BKG_RATE	E [count/s/arcsec2]	Measured count rate in bkg region
21	RAW_BKG_RATE_ERR	E [count/s/arcsec2]	Error in RAW_BKG_RATE
22	RAW_STD_RATE	E [count/s]	Measured count rate in standard region
23	RAW_STD_RATE_ERR	E [count/s]	Error in RATE_STD_RATE
24	COI_STD_FACTOR	E	CoI factor for standard region
25	COI_STD_FACTOR_ERR	E	Error in COI_STD_FACTOR
26	COI_BKG_FACTOR	E	CoI factor for background region
27	COI_BKG_FACTOR_ERR	E	Error in COI_BKG_FACTOR
28	COI_TOT_RATE	E [count/s]	CoI corrected rate in source region
29	COI_TOT_RATE_ERR	E [count/s]	Error in COI_TOT_RATE
30	COI_BKG_RATE	E [count/s/arcsec2]	CoI corrected rate in background region
31	COI_BKG_RATE_ERR	E [count/s/arcsec2]	Error in COI_BKG_RATE
32	COI_SRC_RATE	E [count/s]	Bkg subtracted, CoI corrected source rate
33	COI_SRC_RATE_ERR	E [count/s]	Error in COI_SRC_RATE
34	AP_FACTOR	E	Aperture correction factor
35	AP_FACTOR_ERR	E	Error in AP_FACTOR
36	AP_COI_SRC_RATE	E [count/s]	Aperture corrected, CoI corrected source rate
37	AP_COI_SRC_RATE_ERR	E [count/s]	Error in AP_COI_SRC_RATE
38	MAG	E [mag]	Magnitude from AP_COI_SRC_RATE
39	MAG_ERR	E [mag]	Error in MAG

Table 3.10.2 (continued): Column definitions for UVOTMAGHIST output.

Col	Name	Format [Units]	Definition
40	MAG_BKG	E [mag/arcsec2]	Sky magnitude
41	MAG_BKG_ERR	E [mag/arcsec2]	Error in MAG_BKG
42	MAG_LIM	E [mag]	Limiting magnitude
43	MAG_LIM_SIG	E [sigma]	Sigma for limiting magnitude
44	MAG_COI_LIM	E [mag]	Coincidence limit magnitude
45	FLUX_AA	E [erg/s/cm2/A]	Flux density
46	FLUX_AA_ERR	E [erg/s/cm2/A]	Error in FLUX_AA
47	FLUX_AA_BKG	E [erg/s/cm2/A/arcsec2]	Sky flux density
48	FLUX_AA_BKG_ERR	E [erg/s/cm2/A/arcsec2]	Error in FLUX_AA_BKG
49	FLUX_AA_LIM	E [erg/s/cm2/A]	Flux density corresponding to MAG_LIM
50	FLUX_AA_COI_LIM	E [erg/s/cm2/A]	Coincidence limit flux density
51	FLUX_HZ	E [mjy]	Flux density in mJy
52	FLUX_HZ_ERR	E [mjy]	Error in FLUX_HZ
53	FLUX_HZ_BKG	E [mjy/arcsec2]	Sky flux density
54	FLUX_HZ_BKG_ERR	E [mjy/arcsec2]	Error in FLUX_HZ_BKG
55	FLUX_HZ_LIM	E [mjy]	Flux density corresponding to MAG_LIM
56	FLUX_HZ_COI_LIM	E [mjy]	Coincidence limit flux density in mJy
57	COI_RATE_LIMIT	E [count/s]	CoI corrected rate limit

Parameters

Table 3.10.3 lists the input parameters for UVOTMAGHIST. Parantheses indicate parameters that are not compulsory. If these parameters are not specified on the command line, the tool will look-up the current value in the parameter file. Users can inspect the parameter file by typing plist uvotmaghist.

Table 3.10.3: Parameter descriptions for UVOTMAGHIST.

Parameter	Description
Infile	Input FITS image file containing a series of image extensions
ouffile	Output FITS table containing exposure times, exposure durations, count rates, instrumental magnitudes, fluxes and detection limits
plotfile	Optional GIF file presenting instrumental magnitude over time. NONE produces no GIF output
zerofile	A CALDB file containing filter-dependent zero-points and linear flux conversion coefficients. If the CALDB environment variable is set, caldb will point the tool to the most relevant version of the zero-point file
coinfile	A CALDB file containing coincidence loss correction data
syserr	Include systematic errors?
timezero	Plot start time or 0 to determine from input data [seconds]
ra	Epoch 2000 Right Ascension of the source in decimal degrees
dec	Epoch 2000 Declination of the source in decimal degrees
(srcas)	Radius of a circular extraction region for the source in units of arcsec
(bkgas)	Radius of a circular extraction region for the background in units of arcsec
(srcreg)	Source region file or NONE
(skgreg)	Background region file or NONE
(exclude)	List of extensions to exclude or NONE
(frametime)	Frame time [s]
(nsigma)	No sigma for calculating faintest detection
(logtime)	Plot time using log scale?
(cleanup)	UVOTMAGHIST creates a number of intermediate files in your working directory. These are only useful for software developers. This option removes all intermediate files at the end of the routine. The default is yes
(clobber)	Should UVOTMAGHIST overwrite a file with the same name as the output? The default is no
(history)	Should UVOTMAGHIST write HISTORY keywords to the output file? This creates a record of the processing performed on the file. The default is yes
(chatter)	Verbosity of the tool (0-5). This parameter control how chatty the tool is (0 = quiet. 5 = noisy)

Example

Below we provide a typical invocation of the UVOTMAGHIST tool.

Namibia> uvotmaghist infile=sw00072901259uvv_sk.img.gz outfile=maghist.fit
plotfile=maghist.gif zerofile=caldb coinfile=caldb ra=23.35 dec=41.823
srcas=3 bkgas=10 cleanup=y history=y clobber=y chatter=1

Warnings and Errors

3.12 UVOTCENTROID

Updates

Table 3.11.1: UVOTCENTROID update history.

HEAsoft Version	Date	Description of updates
6.2.1	2007-06-18	Released to the public

Description

UVOTCENTROID locates the centroid positions of a source on a UVOT image.

One of the primary goals of the UVOT instrument is to provide accurate celestial positions for gamma ray burst afterglows. The problem can be separated into three parts i) the conversion from detector pixels to sky coordinates (see the ftool swiftxform), ii) an astrometric adjustment to correct systematic pointing uncertainties in the spacecraft attitude (see uvotaspcorr), and iii) within the statistical limits of the data, locate the centroid of a particular source on the image. It is this last item which is performed by uvotcentroid. Sky coordinate conversion must have been performed before running uvotcentroid. This tool can be used before an astrometric correction, but only to determine a statistical confidence limit. An astrometric correction is required to determine an accurate position. But note well that both items i) and ii) above can only be achieved to a certain accuracy and this systematic uncertainty - *which can often exceed the statistical position error provided by uvotcentroid* - is not propagated through the centroiding calculation. The statistical and systematic errors must be combined before reporting a formal position ucertainty.

This tool employs uvotdetect to determine the position of sources. uvotdetect is a wrapper for the sextractor tool which does the fundamental work of detecting and providing source positions; see http://terapix.iap.fr/rubrique.php?id_rubrique=91/. Both point- and extended sources can be centroided and relatively crowded fields can be accommodated. 1-sigma uncertainties are reported by uvotdetect but the method by which sextractor obtains these errors is somewhat nebulous. In order to achieve some confidence in the statistical error, uvotdetect performs a user-defined number of trials. Before each trial a new, small image is created from the original with the source of interest close to the center. The value of each individual pixel is ajusted by a value chosen at random from the normal cummulative distribution function defined to have a sigma width of the square root of the pixel value. This provides multiple images with identical noise distributions but pixel count distributions varying within the limits defined by the noise. A centoid position for the source in each trial image is recorded and the variance used to estimate confidence limits on the result.

The tool reports back the most-likely RA and Dec of the source, a confidence limit on the position and the number of trials in which the source could not be detected (useful for faint soures). Optionally a plot may be generated to inspect the trial distribution.

If more than one source is located within the subimage, the tool will default to the source nearest to the center of a user-provided region file. The user has the ability to change the size of the sub-field over which uvotcentroid regenerates source images. Small fields have the advantage of less field sources and faster runtimes, at the expense of smaller background statistics. A reasonable subimage size is 20x20 arcsec for optical filter observations and 40x40 arcsec for UV observations.

Input files

UVOTCENTROID requires three input files:

1. A FITS file containing at least one image extension, e.g., sw00072901259/uvot/image/sw00072901259uvv_sk.img.

2. An ascii file defining the source extraction region. Region files created using either ds9 or ximage are compatible with this tool.

Output

UVOTCENTROID returns the centroid position of the source on the screen.

Parameters

Table 3.11.2 lists the input parameters for UVOTCENTROID. Parantheses indicate parameters that are not compulsory. If these parameters are not specified on the command line, the tool will look-up the current value in the parameter file. Users can inspect the parameter file by typing plist uvotcentroid.

Table 3.11.2: Parameter descriptions for UVOTCENTROID.

Parameter	Description
image	The name and correct path to a standard UVOT FITS sky image. if either are incorrect, the tool will complain that it has found no such image. The name should normally contain the FITS extension hosting the image, either by number, e.g. example.fits+1, or name, e.g. example.fits[EXTNAME]. If neither are supplied the tool will look for an image in the primary extension and use it. If none is found, it will default to the first extension and look again. If none is then found the tall will complain.
srcreg	The name and correct path to a standard ds9 region file. These regions can often be complex, uvotcentroid will read the first line which begins with 'fk5;' (ignoring any exclusion regions 'fk5:-') and use these coordinates as the center of each subimages it creates.
confidence	This is the level of confidence (percent) with which you would like the source position reported. A typical value is 90, but the range is $0 <$ confidence < 100, exclusive.
niter	The number of trials to be performed. The more trials, the more accurate the result. A minimum of 100 successful trials are required to provide a confidence estimate. If less than 100 trails could detect the source,a position will be reported but no error estimate. In these cases, re-run the tool with a larger number of trials. uvotcentroid can be run with just 1 trial which will provide a position with no confidence. Provided an astrometic correction has been peformed, most likely this position will be good to at least 1 arcsec $> 99.9\%$ confidence.
threshold	uvotdetect will only recognise sources if they are detected above this threshold. The units are sigma, assuming a Gaussian noise distribution. This argument must be positive and larger than zero. For weak, marginal source a threshold of 2 is suitable.
subdimsiz	The dimensions of the subimage are square and this argument provides the length of one side in arcsec units. The subimage must be large enough to incorporate the source and some background in order for the source to be located.
ra	This is not an input argument. It is used to store the output right ascension in decimal degrees so that users and scripts can access the results easily, using e.g. from a shell: 'pget uvotcentroid ra'
dec	This is not an input argument. It is used to store the output declination in decimal degrees so that users and scripts can access the results easily, using e.g. from a shell: 'pget uvotcentroid dec'

Table 3.11.2 (continued): Parameter descriptions for UVOTCENTROID.

Parameter	Description
conflimit	This is not an input argument. It is used to store the output location uncertainty in arcseconds so that users and scripts can access the results easily, using e.g. from a shell: 'pget uvotcentroid conflimit'. Conflimit is only useful if you get the confidence level associated with it: 'pget uvotcentroid confidence'
plot	If yes, the tool will plot a summary of the result and trials. Two windows are plotted. The left one contains a histogram of the distance in arcsec between the most-likely source position' and all of the trials. The confidence range requested by the user is cast as a region colored yellow behind the histogram. The most-likely RA and Dec, the confidence limit and the number of positive source detections are provided in a dialog box in the top right of this window. The right hand plot shows the distribution of trial positions in RA and Dec space, relative to the most-likely position. The confidence limit is represented as a yellow circle in the background. Each source position is marked by a blue cross symbol. The size of each cross is arbitrary and has no relation to a measured error.
plotdev	The device through which you want to make the plot. Available options will depend on your PGPLOT installation but common usages are: /XS - xserver on your monitor /GIF - gif file /PS - postscript document
(chatter)	Verbosity of the tool (0-5). This parameter control how chatty the tool is (0 = quiet. 5 = noisy)

Example

The following examples illustrate running UVOTCENTROID:
1. run uvotcentroid prompting for all mandatory options:
uvotcentroid

2. run uvotcentroid specifying all required control arguments on the command line:
uvotcentroid image=sw00035934002uuu_sk.img+1 srcreg=srcreg confidence=90 \
niter=1000 threshold=3 subdimsiz=20 chatter=5

3. run uvotcentroid specifying all control arguments on the command line, plotting output to a gif file and retaining all intermediate files:
uvotcentroid image=sw00035934002uuu_sk.img+1 srcreg=srcreg confidence=90 \
niter=1000 threshold=3 subdimsiz=20 plot=y plotdev=/gif cleanup=n chatter=5

Limitations

The subimage and region file must always be fully contained within the sky region sampled by the input image. Cases where the source lies very close to RA = 0h or the celestial poles have not been tested.

3.13 UVOTIMGRISM

Updates

Table 3.12.1: UVOTIMGRISM update history.

HEADAS Version	Date	Description of updates
8.0		Released to Swift science team
9.0	2004-10-07	Released for Burst Advocate training
HEAsoft 6.0.5	2006-04-26	Changed units of sourcex/y to input image pixels More verbose description of positions in various coord systems Generate a region file ZERODET[XY] now give physical pixels

Description

UVOTIMGRISM extracts 1-D FITS spectrum tables from 2-D grism images. The input image must be in DET coordinates so that the dispersion direction is at a fixed angle in the image array. Eventually the user should be able to supply position of a source in RA2000 and Dec2000 coordinates but this capability is not yet available because the grism-specific distortion has not yet been mapped. Instead the user must currently use an external program (e.g. DS9 or XIMAGE) to determine the centroid in pixels of the target zero order. Although the astrometry accuracy (about 5") is not sufficient for centroiding, it should be adequate to identify the zeroth order. UVOTIMGRISM adds appropriate offsets to the source position and rotates this image about the 0th order source so that the 1st order dispersion axis is parallel to the x-direction. The image is rebinned so that square pixels are linear in size, oriented in the dispersion and cross-dispersion directions.

The user also supplies extraction limits for the tool. The dispersion limits are in units of Angstroms (Angstrom) and define the wavelength range of the output spectrum. The cross-dispersion limits are in pixel units and some thought should be taken to optimize the source signal over the background using these parameters. The source spectrum, with background, is generated by summing each column of pixels between the limits in the cross-dispersion direction. The uncertainty in each spectrum bin is calculated assuming a Poisson photon distribution. Note that the detector, while containing a CCD device, behaves an event detector. Consequently there is no effective readout noise.

The background is estimated by determining the mean spectrum in two user-defined regions on either side of the source in the cross-dispersion direction. The mean is calculated iteratively with pixel outliers rejected from the sum. Once a solution has been converged upon, the background spectrum under the source extraction region is determined by interpolation. Errors are, again, treated as Poisson.

Wavelength bins that contain bad pixels are flagged in the QUALITY column of the output spectrum tables. Spectrum bins are converted to wavelength units using a multi-termed polynomial defined in the CALDB. Count rates in each wavelength bin are converted to flux (erg s-1 cm-2 -1) using effective area curves stored in the CALDB.

Input files

UVOTIMGRISM requires five input files:

1. A FITS grism image file. The file may contain multiple extensions but the tool will work on only one image at a time, so the extension of an individual image must be specified. The image must be in DET coordinates, e.g.,
sw00072901259/uvot/image/sw00072901259ugu_dt.img+1

2. The corresponding bad pixel image file. This is created by UVOTBADPIX. Note that this file is not archived by HEASARC or quicklook.

3. A CALDB file containing polynomial coefficients that describe the grating equation. This equation defines the mapping between DET pixel and wavelength e.g.,
$CALDB/data/swift/uvot/bcf/grism/swuvgrism20041120v101.fits

4. A CALDB file containing the wavelength-dependent 1st order effective area curve for the grism filters, e.g.,
$CALDB/data/swift/uvot/cpf/arf/swugv_20010101v013.arf

5. A telescope definition file. In order to convert from SKY to DET coordinates. The data required to perform this operation contained within a single CALDB file, e.g.,
$CALDB/data/swift/uvot/bcf/teldef/swugv1000_20041120v101.teldef

Output files

UVOTIMGRISM puts two spectral extensions in the output file:

1 The first (SPECTRUM) extension, is constructed to be XSPEC (http://heasarc.gsfc.nasa.gov/docs/xanadu/xspec) compatible. This allows the grism spectra to be analyzed consistently and simultaneously with the XRT data. Conventions for high-energy spectral analysis differ from the usual optical approach. Raw, uncalibrated spectra are stored separately from calibration files. Spectral models are fit iteratively to the data by folding the model through the calibrations and then minimized by comparison with the raw data. Fluxes are determined generally from the best-fit model, not the data. XSPEC-compatible data are provided in terms of raw counts, pixels (CHANNEL) and bad pixel flags (QUALITY). The user must supply a suitable calibration file before XSPEC analysis, and this is constructed using UVOTRMFGEN. The counts in this extension have not been background-subtracted. This is done within XSPEC software and requires a separate background spectrum file.

The second (CALSPEC) extension contains the fully-calibrated and background subtracted source spectrum, provided in both net counts and flux units (erg s-1 cm-2 -1). 1-sigma uncertainties are provided in both count and flux units. Wavelength is provided in . This spectrum is intended as a standalone product for inspection and analysis with generic software.

A third extension, IMAGE, contains the 0th and 1st order source in the DET image, rotated to the dispersion axis.

- An example of this FITS file can be found in the archive typically at a location resembling, e.g., sw00072901259.012/data/uvot/product/sw00072901259ugu_1.pha

2. The background spectrum FITS table for use during XSPEC analysis, e.g.,
sw00072901259.012/data/uvot/product/sw00072901259ugu_1_back.pha.

Parameters

Table 3.12.2 lists the parameters for UVOTIMGRISM. Parantheses indicate parameters that are not compulsory. If these parameters are not specified on the command line, the tool will look-up the current value in the parameter file. Users can inspect the parameter file by typing plist uvotimgrism.

Table 3.12.2: Parameter descriptions for UVOTIMGRISM.

Parameter	Description
Infile	Name of the input grism image file or extension
Outfile	Name of the output source spectrum file
backfile	Name of the output background spectrum file
wavefile	Name of the calibration file containing the pixel-to-wavelength conversion factors. If the CALDB environment variable is set, caldb will point the tool to the most relevant version of the file
areafile	Name of the calibration file containing the filter effective areas. If the CALDB environment variable is set, caldb will point the tool to the most relevant version of the file
teldeffile	Name of the calibration file containing the telscope definition data, such as pixel size and optical distortion map. If the CALDB environment variable is set, caldb will point the tool to the most relevant version of the file
badpixfile	Name of the bad pixel image file that corresponds to the input grism image. The source of this bad pixel image is UVOTBADPIX
ra	Right Ascension (epoch 2000) of the source, currently not implemented
dec	Declination (epoch 2000) of the source, currently not implemented
(sourcex)	0th order position in pixels. This position will be used instead of ra if ra = -1
(sourcey)	0th order position in pixels. This position will be used instead of dec if dec = -1
(nsigma)	For each image column, the threshold above or below the mean value over which individual pixels should be rejected from the background calculation
— (uvmin)	UV grism filter only. The lower limit of the spatial filter region in the dispersion direction. The units are Angstroms
(uvmax)	UV grism filter only. The upper limit of the spatial filter region in the dispersion direction. The units are Angstroms
(ang)	Aangle in degrees subtended by the DETX axis and the dispersion direction of the 1st order light. Approximate values are 148.1 (V grism,nominal), 140.5 (V grism, clocked), 151.4 (U grism, nominal) and 144.5 (U grism, clocked)
(srcwid)	The size of the spatial filter region of the source in cross-dispersion direction. The units are pixels
(bkgwid1)	Width of lower background region [pixels]
(bkgoff1)	Separation of lower background region from source region [pixels]
(bkgwid2)	Width of upper background region [pixels]
(bkgoff2)	Separation of upper background region from source region [pixels]
(wavemin)	The lower limit of the spatial filter region in the dispersion direction. The units are Angstroms
(wavemax)	The upper limit of the spatial filter region in the dispersion direction. The units are Angstroms
(cleanup)	UVOTIMGRISM is a wrapper script. This parameter determines whether intermediate files are deleted from the working directory. The default is yes
(clobber)	Should UVOTIMGRISM overwrite a file with the same name as the output? The default is no
(history)	Should UVOTIMGRISM write HISTORY keywords to the output file? This creates a record of the processing performed on the file. The default is yes
(chatter)	Verbosity of the tool (0-5). This parameter control how chatty the tool is (0 = quiet. 5 = noisy)

Example

Below we provide a typical invocation of the UVOTIMGRISM tool.

Namibia> uvotimgrism infile=sw00072901259ugv_dt.img+1 \
outfile=sw00072901259ugv_sr.pha backfile=sw00072901259ugv_bk.pha \
badpixfile=badpix.img+1 wavefile=caldb areafile=caldb teldeffile=caldb \
ra=-1 dec=-1 sourcex=1539.99 sourcey=657.28 vang=206.7 vsrcwid=20 vbkgwid=50 \
wavemin=2900 wavemax=5500 nsigma=5 cleanup=y clobber=y history=y chatter=1

Warnings and Errors

3.14 UVOTRMFGEN

Updates

Table 3.13.1: UVOTRMFGEN update history.

HEAsoft Version	Date	Description of updates
8.0		Released to Swift science team
9.0	2004-10-07	Released for Burst Advocate training

Description

As described in section 3.11, the output from UVOTIMGRISM is not yet ready for spectral analysis using the XSPEC package. First we must construct a specific calibration file that maps raw pixels to physical units and converts raw counts to flux energy. The term for this calibration file is Redistribution Matrix FILE or RMF. XSPEC will fold analytic or tabular spectral models through the RMF before performing a statistical comparison between data and model and iterative minimization of the parameters to produce a best-fit model.

The redistribution matrix is constructed using two factors. The first is a simple translation from pixel number to photon energy using the grism dispersion equation stored in the caldb. This takes the form of a low-order polynomial. The second is the finite possibility that a photon will land on the pixel predicted by the dispersion equation. In other words each row of the redistribution matrix is convolved by a function that defines the 1st order spectral resolution of the grism. Currently this function is assumed to be Gaussian, where the Full-Width Half-Maximum (FWHM), which varies with wavelength, is stored in the caldb.

In order to convert raw counts to flux, the area under the convolution function above for each pixel varies as a function of wavelength and is equal to the 1st order effective area of the optics.

Input files

UVOTRMFGEN requires three input files:

1. A FITS table containing an XSPEC-compatible spectrum. UVOTIMGRISM will create this file, e.g.,

- sw00072901259/uvot/product/sw00072901259uvu_sr.pha

2. A CALDB file containing the wavelength-dependent 1st order effective area curve for the grism filters, e.g.,

- $CALDB/data/swift/uvot/cpf/arf/swu20041007v001.arf

3. A CALDB file containing the line spread function of the 1st order grism spectrum, e.g.,

- $CALDB/data/swift/uvot/bcf/grism/swvgrism20041007v001.fits

Output files

UVOTRMFGEN has a single output file:

A FITS file containing an ancillary response matrix, normalized to the effective area of the 1st order grism throughput, e.g., sw00072901259/uvot/product/sw00072901259uvu_sr.rsp

Parameters

Table 3.13.2 lists the parameters for UVOTRMFGEN. Parantheses indicate parameters that are not compulsory. If these parameters are not specified on the command line, the tool will look-up the current value in the parameter file. Users can inspect the parameter file by typing plist uvotrmfgen.

Table 3.13.2: Parameter descriptions for UVOTRMFGEN.

Parameter	Description
spectrum	XSPEC-compliant FITS table containing a 1st order grism spectrum
outfile	A Redistribution Matrix File, normalized for the 1st order effective area of the grism filter
areafile	Name of the calibration file containing the filter effective areas. If the CALDB environment variable is set, caldb will point the tool to the most relevant version of the file
lsffile	Name of the calibration file containing the 1st order spectral line spread functions. If the CALDB environment variable is set, caldb will point the tool to the most relevant version of the file
(clobber)	Should UVOTRMFGEN overwrite a file with the same name as the output? The default is no
(history)	Should UVOTRMFGEN write HISTORY keywords to the output file? This creates a record of the processing performed on the file. The default is yes
(chatter)	Verbosity of the tool (0-5). This parameter control how chatty the tool is (0 = quiet. 5 = noisy)

Example

Below we provide a typical invocation of the UVOTRMFGEN tool.

```
Namibia> uvotrmfgen spectrum=sw00072901259ugv_sr.pha \
outfile=sw00072901259ugv_sr.rsp areafile=caldb lsfile=caldb \
clobber=y history=y chatter=1
```

Warnings and Errors

3.15 UVOTIMSUM

Updates

Table 3.14.1: UVOTIMSUM update history.

HEADAS Version	Date	Description of updates
8.0		Released to Swift science team
9.0	2004-10-07	Released for Burst Advocate training
HEAsoft 6.0.5	2006-04-26	Modified exclude parameter to support excluding HDUs on the basis of the presence/value of the ASPCORR keyword By default, HDUs without ASPCORR keyword are excluded from sum

Description

By co-adding a series of individual images the depth of UVOT observations can be maximized. There are many tools available to perform this operation, e.g., XIMAGE, which is supplied as part of the HEASoft distribution. During data reduction pipeline development, it became clear that existing software was too slow to meet data availability requirements, therefore this tool was developed to increase the efficiency of the pipeline. By default, UVOT sky images are all oriented in the same direction which simplifies the rebinning necessary to accurately add images from slightly different pointings. While users are welcome to employ this tool, note that it was developed for speed, not flexibility. For example, there are no options to choose a subsample of images from a UVOT file. It is recommended that post-pipeline users instead use XIMAGE for image-combining tasks:

http://heasarc.gsfc.nasa.gov/docs/xanadu/ximage/ximage.html

In order to avoid the smearing effects of spacecraft aspect drift, UVOTIMSUM must translate each image so that they have a common pointing. This requires image rebinning which results in the loss of data near the detector or window edges and the smearing of bad pixels. It is therefore critical that exposure maps are constructed for each image using UVOTEXPMAP and these maps fed through UVOTIMSIM in an entirely consistent way to the data mages. By default, the output image shares a consistent boresight position with the first exposure in the series, and has the same pixel binning as the coarsest image in the series.

There are two rebinning methods available to the user. 1) GRID assumes that all images are oriented in the same direction. This means that image rotations are ignored during image binning, speeding up the process to a suitable level for rapid pipeline processing. All UVOT level II data is oriented so that north is up and east is to the left. 2) XIMAGE is a slower process that makes no assumption concerning image rotation. Both rotation and translation are performed during this procedure.

Input files

UVOTIMSUM requires a single input file:

1. A FITS image file containing a series of image extensions, e.g.,

- sw00072901259/uvot/image/sw00072901259uvv_sk.img.gz

Output files

UVOTIMSUM has a single output file:

2. A FITS file containing a single image extension.

Parameters

Table 3.14.2 lists the input parameters for UVOTIMSUM. Parantheses indicate parameters that are not compulsory. If these parameters are not specified on the command line, the tool will look-up the current value in the parameter file. Users can inspect the parameter file by typing plist uvotimsum.

Table 3.14.2: Parameter descriptions for UVOTIMSUM.

Parameter	Description
Infile	Input FITS image file containing a series of image extensions
outfile	Output FITS image with a single image extension
method	Image rebinning method. Options are GRID or XIMAGE
(pixsize)	Pixel size for the output image. The default (or if pixsize=0) is to rebin the input images to match the coarsest image in the series. Units are degrees
(cleanup)	UVOTIMSUM creates a number of intermediate files in your working directory. These are only useful for software developers. This option removes all intermediate files at the end of the routine. The default is yes
(clobber)	Should UVOTIMSIM overwrite a file with the same name as the output? The default is no
(history)	Should UVOTIMSIM write HISTORY keywords to the output file? This creates a record of the processing performed on the file. The default is yes
(chatter)	Verbosity of the tool (0-5). This parameter control how chatty the tool is (0 = quiet. 5 = noisy)

Example

Below we provide a typical invocation of the UVOTIMSUM tool.

Namibia> uvotimsum infile=sw00072901259uvv_sk.img.gz \
outfile=v_total.img method=GRID pixsize=0 cleanup=y history=y clobber=y chatter=1

Warnings and Errors

3.16 UVOTSEQUENCE

Updates

Table 3.15.1: UVOTSEQUENCE update history.

HEADAS Version	Date	Description of updates
6.0.2	2005-08-18	Released to public

Description

Lists and visualizes the exposures and snapshots contained within a UVOT observing sequence. Given an ascii list of the full or relative paths to a series of UVOT FITS image files, this tool will determine the times of each image exposure. Start and stop times of each snapshot are inferred from a spacecraft attitude file provided and each image is mapped to a particular snapshot. Results are provided as standard output. Optionally, the results may be plotted, e.g., in the image below.

Input files

UVOTSEQUENCE requires two input files:

1. An ascii file containing full or relative paths to a series of UVOT FITS image files.
2. A spacecraft attitude file; e.g., auxil/sw00072901259sat.fits.

Output files

UVOTIMSUM has one optional output file:

1. A plot of the exposure sequence in the format of the users choice; e.g. postscript, GIF.

Parameters

Table 3.15.2 lists the input parameters for UVOTSEQUENCE. Parantheses indicate parameters that are not compulsory. If these parameters are not specified on the command line, the tool will look-up the current value in the parameter file. Users can inspect the parameter file by typing plist uvotsequence.

Table 3.15.2: Parameter descriptions for UVOTSEQUENCE.

Parameter	Description
infile	Ascii list containing the paths to one or more UVOT image files
attfile	Spacecraft attitude file
trigtime	MET of the BAT trigger
(plotseq)	Whether to display the sequence graphically. The default is yes
(chatter)	Verbosity of the tool (0-5). This parameter control how chatty the tool is (0 = quiet. 5 = noisy)

Example

Below we provide a typical invocation of the UVOTSEQUENCE tool.
uvotsequence infile=images.lis attfile=sw00072901259sat.fits trigtime=11896589

Warnings and Errors

3.17 UVOTIMAGE

Updates

Table 3.15.1: UVOTIMAGE update history.

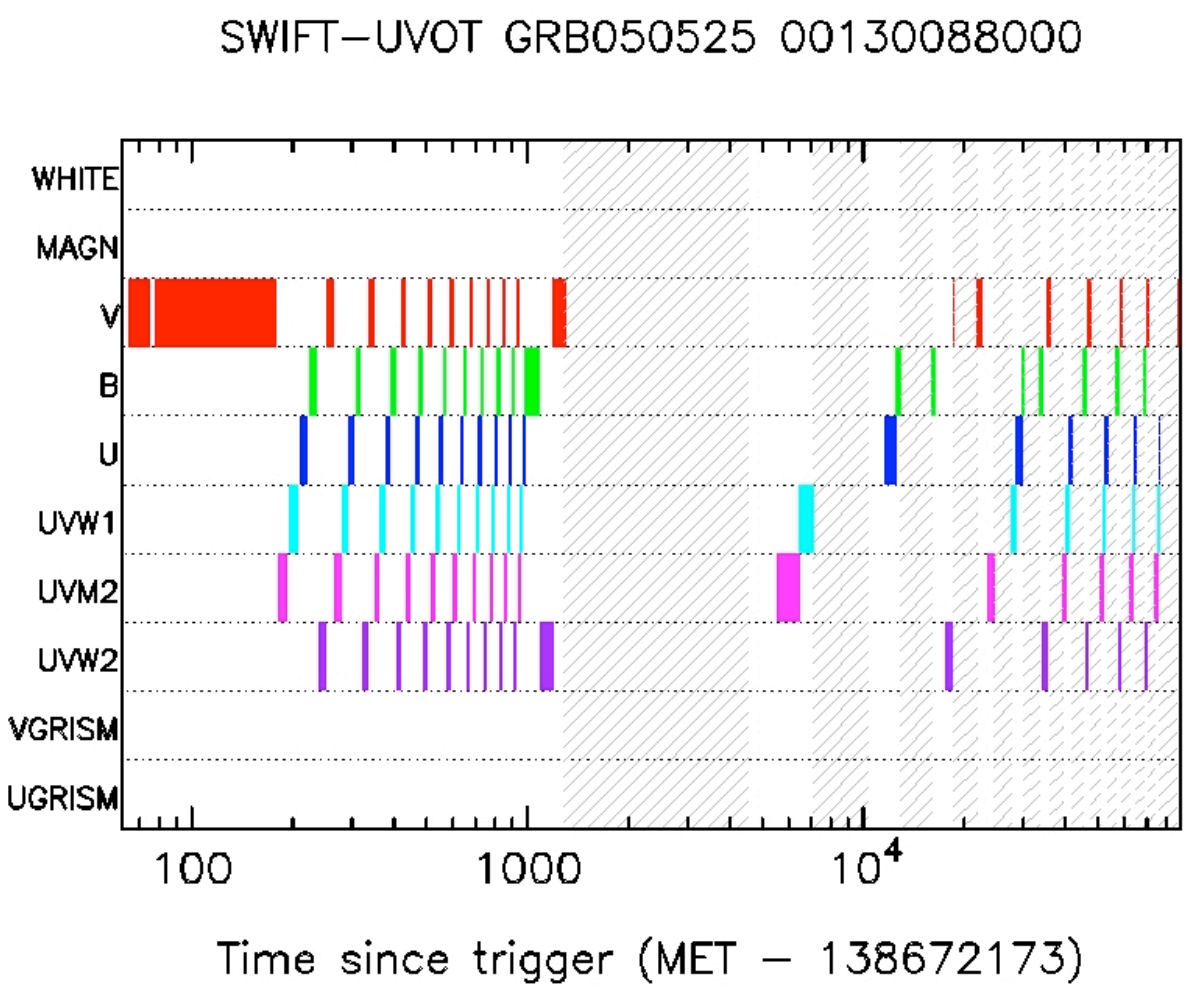

Figure 3.2: Graphical output from UVOTSEQUENCE, displaying the exposures (colours) and snapshots (in between the gray intervals) of a sequence.

HEAsoft Version	Date	Description of updates
6.0.0	2005-04-12	Released to the public
6.0.5	2006-04-26	Added output in mJy Changed default value of output parameter to ALL

Description

UVOTIMAGE iterates over Level 1 UVOT image and event files creating Level 1 and 2 image files.

Input files

UVOTIMAGE requires four input files:

1. A FITS input image file(s) or a comma-delimited list of file names or @<file> where <file> contains the names of the files to process, one per line.

2. Input attitude file.
3. Telescope definition calibration file or CALDB.
4. Alignment file or CALDB.

Output files

UVOTIMAGE creates Level 1 and 2 image files.

Parameters

Table 3.15.2 lists the input parameters for UVOTIMAGE. Parantheses indicate parameters that are not compulsory. If these parameters are not specified on the command line, the tool will look-up the current value in the parameter file. Users can inspect the parameter file by typing plist uvotimage.

Table 3.15.2: Parameter descriptions for UVOTIMAGE.

Parameter	Description
infile	[filename] Name of input image file(s). This can be a comma-delimited list of file names or @<file> where <file> contains the names of the files to process, one per line.
prefix	[string] Prefix for output files.
attfile	[filename] Input attitude file.
teldeffile	[filename]Telescope definition calibration file or CALDB.
alignfile	[filename] Alignment file or CALDB.
ra	[real] Nominal R.A.
dec	[real] Nominal Dec.
roll	[real] Nominal roll.
(catfile = NONE)	[filename] UVOT exposure catalog or NONE.
(flatfield = no)	[boolean] Perform flat fielding? Not implemented.
(mod8corr = no)	[boolean] Perform modulo 8 noise correction? Not implemented.
(cleanup = yes)	[boolean] Remove intermediate files?
(history = yes)	[boolean] Write history keywords?
(clobber = no)	[boolean] Overwrite existing files?
(chatter = 1)	[enumerated integer] Standard HEASoft chatter parameter.

Example

Below we provide a typical invocation of the UVOTIMAGE tool.

Namibia> uvotimage infile=@files prefix=Q teldeffile=CALDB alignfile=CALDB

Warnings and Errors

3.18 UVOTSKYCORR

Updates

Table 3.17.1: UVOTSKYCORR update history.

HEAsoft Version	Date	Description of updates
6.0.3	2005-10-07	Released to the public
6.0.4	2005-11-28	Renamed 'partition' parameter to 'catspec', which now gives the path of the catalog descriptor

Description

UVOTSKYCORR iterates over UVOT image files attempting to determine or applying aspect corrections.

Input files

UVOTSKYCORR requires four input files:

1. A FITS input image file(s) or a comma-delimited list of file names or @<file> where <file> contains the names of the files to process, one per line.
2. Input attitude file.
3. Input correction file.

Output files

UVOTSKYCORR creates Level 2 image files and write information about the aspect correction to this file.

Parameters

Table 3.17.2 lists the input parameters for UVOTSKYCORR. Parantheses indicate parameters that are not compulsory. If these parameters are not specified on the command line, the tool will look-up the current value in the parameter file. Users can inspect the parameter file by typing plist uvotimage.

Table 3.17.2: Parameter descriptions for UVOTSKYCORR.

Parameter	Description
what	[string] (ID—SKY) Whether to find corrections (what=ID) or apply corrections (what=SKY).
skyfile	[filename] Name of input image file(s). This can be a comma-delimited list of file names or @<file> where <file> contains the names of the files to process, one per line.
corrfile	[filename]Input corrections file for what=SKY.
attfile	[filename] Input attitude file.
outfile	[filename] Output file name. For what=ID, the aspect corrections will be written to this file.
(starid = NONE)	[string] Parameters to pass to star identification.
(catspec = usnob1.spec)	[filename] Catalog descriptor file.
(cleanup = yes)	[boolean] Remove intermediate files?
(history = yes)	[boolean] Write history keywords?
(clobber = no)	[boolean] Overwrite existing files?
(chatter = 1)	[enumerated integer] Standard HEASoft chatter parameter.

Example

Below we provide a typical invocation of the UVOTSKYCORR tool.
uvotskycorr what=ID infile=usk.img

Warnings and Errors

3.19 UVOTASPCORR

Updates

Table 3.18.1: UVOTASPCORR update history.

HEAsoft Version	Date	Description of updates
6.0.5	2006-04-26	Released to the public

Description

UVOTASPCORR simplifies finding/applying aspect corrections to UVOT sky images.

Input files

UVOTASPCORR requires one input:

1. The name of a Swift observation directory. The structure of a Swift observation directory is
sw<obsid>/
auxil/sw*sat.fits
uvot/
mage/sw*u*_sk.img
See also the flat parameter.

Output files

UVOTASPCORR creates aspect solution log files.

Parameters

Table 3.18.2 lists the input parameters for UVOTASPCORR. Parantheses indicate parameters that are not compulsory. If these parameters are not specified on the command line, the tool will look-up the current value in the parameter file. Users can inspect the parameter file by typing plist uvotimage.

Table 3.18.2: Parameter descriptions for UVOTASPCORR.

Parameter	Description
output	[directory] Name of output directory. This directory will hold the aspect solutions and log files. The sky images are modified in place.
filter = ALL	[string] (ALL—V—U—B—UVM2—UVW1—UVW2—WHITE) Which UVOT filter(s) to process.
apply	[boolean]Apply the aspect corrections? Note that the input images files must be updatable (writable and uncompressed) to apply the corrections.
catspec = usnob1.spec	[filename] Star catalog descriptor.
(starid = NONE)	[string] Star identification parameters.
(flat = no)	[boolean] Is directory hierarchy flat? Set flat=yes if all of the input files (attitude file, sky images) are in one (the input) directory.
(chatter = 4)	[integer] <1-5> Standard HEAdas chatter parameter.

Example

The following examples illustrate running uvotaspcorr and how to run uvotaspcorr on an observation:

Namibia> uvotaspcorr input=12345678012

Warnings and Errors

Note

See Section 2.12. for a description how to apply an aspect correction to UVOT images.

Chapter 4

Event Tools

4.1 UVOTSCREEN

Updates

Table 4.1.1: UVOTSCREEN update history.

HEADAS Version	Date	Description of updates
8.0		Released to Swift science team
9.0	2004-10-07	Released for Burst Advocate training

Description

UVOTSCREEN reads a FITS event table, determines the quality of each event, grades each event as good or bad according to user-defined criteria, and rejects all bad events from the output FITS table. Screening criteria can be related to internal data, performing logical calculations on, e.g., the arrival times of events and event position, but can also be related to external data, such as the orbit and attitude data of the spacecraft during the observation.

Event quality can be deemed to be poor based on two tests. The first is to compare the pixel position with a list of known cosmetic defects over the detector that is stored in the caldb. UVOTSCREEN updates the values of the QUALITY column in the FITS table for cosmetic defects. The second test is to check pixels for compression damage. UVOT telemettry may or may not come down in a compressed format, depending on the volume of the data. The compression schemes can potentially corrupt data if events from the same readout frame are widely separated or too numerous, or events occur to infrequently in time. Compression damaged is already flagged in the QUALITY column by the telemetry-to-FITS conversion software executed in the pipeline.

Table 4.1.2: Types of event badness found in Event mode tables.

QUALITY	Description
0	Good event
1	Dead pixel
2	Cold pixel
4	Hot pixel
8	Flickering pixel
16	Compression damaged event position
32	Compression damaged event time
64	Compression has caused loss of events from frame

Data is screened by calling the FTOOLS MAKETIME and EXTRACTOR internally. Screening arguments are a string of operators passed to the tool as input parameters. For example, to screen by both time and pixel quality, an argument resembling:

evexpr=time > 123456789 && quality == 0

is appropriate. External parameters, from, e.g., tbe orbit and attitude filter file provided with yout data can also be used to screen events. For example, the following argument will EXCLUDE events from the output file that were obtained within 10 degrees of the bright Earth limb or from within the South Atlantic Anomaly:

aoexpr=BR_EARTH > 10 && SAA == 0

The full functionality and syntax of these operators is discussed in the MAKETIME and EXTRACTOR on-line helps, e.g., type fhelp extractor on your command line.

UVOTSCREEN was developed as a pipeline tool rather than a user tool. it is recommended that general Swift users screen their UVOT data using the general HEASoft tools MAKETIME and XSELECT:

http://heasarc.gsfc.nasa.gov/docs/software/lheasoft/ftools/xselect/xselect.html

The one exception to this rule is if the bad pixel table in the Swift caldb has been updated since the pipeline processing of your data. In this case, UVOTSCREEN should be run in order to populate the QUALITY column in the event tables correctly.

Input files

UVOTSCREEN requires three input files:

1. A FITS table containing UVOT events, e.g.,

- 00072901259/uvot/event/sw00072901259uvvpo_uf.evt.gz

2. A FITS table containing time-tagged spacecraft orbit and attitude data for the observation, e.g.,

- 00072901259.012/auxil/sw00072901259sao.fits.gz

3. The caldb product containing a cosmetic bad pixel list, e.g.,

- $CALDB/data/swift/uvot/bcf/badpix/swubadpix20041007v001.fits

Output files

UVOTSCREEN has a single output file:

1. A FITS table containing screened UVOT events, e.g.,

- 00072901259/uvot/event/sw00072901259uvvpo_cl.evt

Parameters

Table 4.1.3 lists the input parameters for UVOTSCREEN. Parantheses indicate parameters that are not compulsory. If these parameters are not specified on the command line, the tool will look-up the current value in the parameter file. Users can inspect the parameter file by typing plist uvotscreen.

Table 4.1.3: Parameter descriptions for UVOTSCREEN.

Parameter	Description
infile	Name of the input UVOT event file
outfile	Name of the screened output event file
attorbfile	The attitude and orbit data file. This is calculated in the Swift pipeline and supplied with each observation in the archive, but can also be generated by the user using the PREFILTER tool
badpixfile	Name of the calibration file containing a list of cosmetic defects over the detector. If the CALDB environment variable is set, caldb will point the tool to the most relevant version of the file
aoexpr	The filtering expression, based on the attitude and orbit data, with which to screen the events
evexpr	The filtering expression, based on internal data within the event file, with which to screen the events
(cleanup)	UVOTSCREEN creates a number of intermediate files in your working directory. These are only useful for software developers. This option removes all intermediate files at the end of the routine. The default is yes
(clobber)	Should UVOTSCREEN overwrite a file with the same name as the output? The default is no
(history)	Should UVOTSCREEN write HISTORY keywords to the output file? This creates a record of the processing performed on the file. The default is yes
(chatter)	Verbosity of the tool (0-5). This parameter control how chatty the tool is (0 = quiet. 5 = noisy)

Example

Below we provide a typical invocation of the UVOTSCREEN tool. Example input and output files can be copied from ftp://heasarc.gsfc.nasa.gov/docs/swift/foo/uvotscreen.

Namibia> uvotscreen infile=sw00072901259uvvpo_uf.evt.gz outfile=sw00072901259uvvpo_cl.evt badpixfile=caldb attorbfile=sw00072901259sao.fits.gz
aoexpr=SUN_ANGLE > 20 evexpr=QUALITY == 0 cleanup=y history=y clobber=y chatter=1

Warnings and Errors

4.2 UVOTEVGRISM

Updates

Table 4.2.1: UVTEVGRISM update history.

HEADAS Version	Date	Description of updates
8.0		Released to Swift science team
9.0	2004-10-07	Released for Burst Advocate training

Description

UVOTEVGRISM performs two functions. Firstly it screens events according to position, keeping only those that coincide approximately with the 0th and 1st order light of a specific source. The resulting list is output to a new FITS event table. By default this table will also include a contribution from the background sky and possibly neighbouring sources that are confused with the target. Secondly this tool uses the UVOT dispersion equations to calculate wavelengths for each event. Results are written in units of Angstroms () to a column called WAVELENGTH. The dispersion equation is stored as polynomial coefficients in the caldb.

The user must supply a source position in RA2000 and Dec2000 coordinates, or alternatively a location of the 0th order source in detector coordinates (DETX,DETY) in units of mm. They must also supply the angle of the dispersion axis relative to the DETX direction, the cross-dispersion width of the screened region and the wavelength range.

In order to improve the accuracy of the wavelength calculations the tool uses the attitude history file to track spacecraft pointing during the grism exposures and correct event positions on the detector. In the absence of a series of records in the attitude history, the tool will interpolate linearly between two disparate records.

Input files

UVOTEVGRISM requires four input files:

1. A FITS event table, e.g.,

- 00072901259/uvot/ievent/sw00072901259uvupo_cl.evt.gz

2. A spacecraft attitude file covering the duration of the UVOT exposure, e.g.,

- $CALDB/data/swift/misc/sw00072901259sat.fits.gz

3. A CALDB file containing polynomial coefficients that describe the grating equation. This equation defines the mapping between DET pixel and wavelength e.g.,

- $CALDB/data/swift/uvot/bcf/grism/swuvgrism20041007v001.fits

4. A telescope definition file. In order to convert from SKY to DET coordinates. The data required to perform this operation contained within a single CALDB file, e.g.,

- $CALDB/data/swift/uvot/bcf/teldef/swu20041007.teldef

Output files

UVOTEVGRISM has a single output file:

1. A FITS event table, screened by region to contain 0th and 1st order events from a specific source, and with an additional column containing the wavelength of each event in units of .

Parameters

Table 4.2.2 lists the input parameters for UVOTEVGRISM. Parantheses indicate parameters that are not compulsory. If these parameters are not specified on the command line, the tool will look-up the current value in the parameter file. Users can inspect the parameter file by typing plist uvotevgrism.

Table 4.2.2: Parameter descriptions for UVOTEVGRISM.

Parameter	Description
infile	Name of the input grism event file
outfile	Name of the output grism event file, with wavelength column added
wavefile	Name of the calibration file containing the pixel-to-wavelength conversion factors. If the CALDB environment variable is set, caldb will point the tool to the most relevant version of the file
teldeffile	Name of the calibration file containing the telscope definition data, such as pixel size and optical distortion map. If the CALDB environment variable is set, caldb will point the tool to the most relevant version of the file
attfile	Name of the attitude history table for the observation
(attlim)	The number of seconds beyond a particular entry in the attitude history file after which the tool is allowed to interpolate pointing data. The tool will warn the user if gaps in the attitude file exceed this limit
ra	The Right Ascension (epoch 2000) of the source
dec	The Declination (epoch 2000) of the source
(uvang)	UV grism filter only. The angle in degrees subtended by the DETX axis and the dispersion direction of the 1st order light
(uvwid)	UV grism filter only. The size of the spatial filter region in cross-dispersion direction. The units are pixels
(uvmin)	UV grism filter only. The lower limit of the spatial filter region in the dispersion direction. The units are Angstroms
(uvmax)	UV grism filter only. The upper limit of the spatial filter region in the dispersion direction. The units are Angstroms
(vang)	V grism filter only. The angle in degrees subtended by the DETX axis and the dispersion direction of the 1st order light
(vwid)	V grism filter only. The size of the spatial filter region in cross-dispersion direction. The units are pixels
(vmin)	V grism filter only. The lower limit of the spatial filter region in the dispersion direction. The units are Angstroms
(vmax)	V grism filter only. The upper limit of the spatial filter region in the dispersion direction. The units are Angstroms
(clobber)	Should UVOTEVGRISM overwrite a file with the same name as the output? The default is no
(history)	Should UVOTEVGRISM write HISTORY keywords to the output file? This creates a record of the processing performed on the file. The default is yes
(chatter)	Verbosity of the tool (0-5). This parameter control how chatty the tool is (0 = quiet. 5 = noisy)

Example

Below we provide a typical invocation of the UVOTEVGRISM tool. Example input and output files can be copied from ftp://heasarc.gsfc.nasa.gov/docs/swift/foo/uvotevgrism.

Namibia> uvotevgrism infile=sw00072901259uvupo_cl.evt.gz outfile=vgrism.evt wavelfile=caldb teldeffile=caldb attfile=sw00072901259sat.fits.gz attlim=32 vang=212.5 vwid=20 vmin=2900 vmax=5500 history=y clobber=y chatter=1

Warnings and Errors

Chapter 5

FHelp

5.1 UVOT2PHA

NAME

uvotpha - create a pha file from a UVOT image and region files

USAGE

uvotpha infile=<filename> srcreg=<filename> bkgreg=<filename> srcpha=<filename> bkgpha=<filename>

DESCRIPTION

This tool uses XIMAGE to compute counts and areas from specified regions of a UVOT image and then calls ASCII2PHA to generate corresponding OGIP-compliant PHA files for the source and background regions.

PARAMETERS

infile [filename]

- UVOT FITS image file. If a particular extension isn't specified then the first extension will be used.

srcreg = "source.reg" [filename]

- Name of source region file (ascii) generated, eg, by XIMAGE or DS9.

bkgreg = "background.reg" [filename]

- Background region file (ascii) generated, eg, by XIMAGE or DS9

srcpha = "source.pha" [filename]

- Name for the output FITS source PHA file.

bkgpha = "background.pha" [filename]

- Name for the output FITS background PHA file.

respfile [filename]

- Name of the UVOT response file to be used. Specifying "CALDB" will cause the tool to look up the proper file in the Calibration Database and use it.

(phatype = "rate") [string]

- Output pha files may be expressed in COUNTS or RATE (default)

(ra = -) [string]

- Right Ascension. If provided, it will be written in the output pha file as RA_OBJ. The default is to do nothing.

(dec = -) [string]

- Declination. If provided, it will be written in the output pha file as DEC_OBJ. The default is to do nothing.

(date_obs = -) [string]

- Start date for observation. If provided, this value will be written in the output pha file as DATE-OBS. By default the value of DATE-OBS, if present, will be copied from the input file instead.

(time_obs = -) [string]

- Start time for observation. If provided, this value will be written in the output pha file as TIME-OBS. By default the value of TIME-OBS, if present, will be copied from the input file instead. The FITS standard allows for the time to be included as part of DATE-OBS so that a separate TIME-OBS keyword is not needed.

(date_end = -) [string]

- End date for observation. If provided, this value will be written in the output pha file as DATE-END. By default the value of DATE-END, if present, will be copied from the input file instead.

(time_end = -) [string]

- End time for observation. If provided, this value will be written in the output pha file as TIME-END. By default the value of TIME-END, if present, will be copied from the input file instead. The FITS standard allows for the time to be included as part of DATE-END so that a separate TIME-END keyword is not needed.

(tmpdir = .) [directory]

- Location to write two very small temporary ascii files. The default is the current working directory.

(chatter = 1) [enumerated integer]

- Standard HEAdas chatter parameter (1-5) controlling the verbosity of the task.

(clobber = no) [boolean]

- Standard HEAdas clobber parameter; controls whether the output pha files (and temporary ascii files) are permitted to overwrite existing files.

(history = yes) [boolean]

- Standard HEAdas history parameter; controls whether the runtime parameter values should be written in block of HISTORY keywords in the output pha files.

EXAMPLES

The following examples illustrate running uvot2pha

1. run uvot2pha prompting for all options:

- uvot2pha

2. run uvot2pha specifying image and region files on command line (output and response filenames will be prompted for):

- uvot2pha infile=sw00000001001ubbsky.img srcreg=source.reg bkgreg=background.reg

3. run uvot2pha specifying the RA/DEC (with required parameters prompted for):

- uvot2pha ra='23 21 01' dec='-41 48 36'

SEE ALSO

LAST MODIFIED

October 2007

5.2 UVOTAPERCORR

NAME

uvotapercorr - aperture correction for SWIFT-UVOT source count rates.

USAGE

uvotapercorr cntrate=<float> ratesig=<float> aperad=<float> fwhmsig=<float> filter=<string> psffile=<filename> chatter=<enumerated integer>

DESCRIPTION

The Swift-UVOT photometry aperture used to derive the photometric zero points is a circle of radius 5 arcsec. This aperture contains approximately 85% of the photons from a point source, and keeps background contamination to a reasonable level. It is large enough to smooth out the ubiquitous modulo-8 fixed-pattern image noise which results from the electronic oversampling of UVOT detector pixels. In cases where UVOT point sources are faint and approach the background-limited detection threshold, it is desirable to perform source detection using smaller apertures. While reducing the aperture from the size of a 5 arcsec radius circle increases signal-to-noise, and consequently the detection probability, the drawback is that counts within the wings of the source point spread function (PSF) are ignored and the brightness of the source underestimated. uvotapercorr is a tool to calculate and add the neglected photons within the PSF wings to small-aperture count rates so that magnitude conversions can be performed accurately.

The method adopted in this tool is a quick-and-dirty approach with a number of critical caveats which are discussed below. The mean PSFs for each broad band filter are characterized as encircling energy functions in tables within the Swift-UVOT CALDB; see $CALDB/data/swift/uvota/cpf/psf/ and the release note at http://heasarc.gsfc.nasa.gov/docs/heasarc/caldb/swift/docs/uvot/. The U, B, and V PSFs are all similar in form and width. The UVW1, UVM2 and UVW2 PSFs are generally narrower in the core with broader wings compared to the optical filters. The PSFs are employed to extrapolate count rates measured within a small aperture to count rates predicted from within the 5 arcsec aperture used to derive the photometric zero points. This is achieved cheaply by a linear interpolaton of the PSF table, which provides the fraction of total energy within the user-provided circular aperture radius, "aperad". The fraction is a simple scaling quantity which is applied to both the user supplied count rate, "cntrate" and count rate uncertainty, "raterr" in order to provide an aperture-corrected count rate.

This procedure is a useful approximation if the extraction aperture is circular and centered precisely on the centroid of the PSF. Any deviation from these criteria will yield unquantified systematic errors.

There are systematic problems to be aware of. The FWHM of the PSF is dependent on the temperature of the UVOT focusing rods. Their temperature varies slightly with time because the voltage on the heaters changes through the spacecraft orbit. This can cause variations in the shape of the PSF. Currently this variation has not been well-characterized. Until this occurs it remains the user's responsibility to add a systematic uncertainty to the width of the PSF which is propagated into the corrected count rate uncertainty. Current functionality assumes that the form of the PSF is constant over time, i.e., the ratio of counts in the line core and line wings remains constant, but the width of the entire profile is scaled by a variable quantity during a sequence of exposures. The PSFs characterized in the Swift-UVOT CALDB are considered to be 'typical'. To characterize the systematic uncertainty added by variable PSFs to the corrected count rate, the user provides an argument, "fwhmsig", which estimates the fractional rms variation of the FWHM of point sources. The UVOT team recommends that this number be 3%. The rms spread is then propagated through the calculation and is added in quadrature to the corrected measurement uncertainty. Consequently, be aware that the signal-to-noise in the corrected count rate is greater than the signal-to-noise in the raw count rate. This may result in some confusion when reporting results. When using apertures < 5 arcsec, detection thresholds should be reported

using raw count rates, whereas magnitudes should be calculated using aperture-corrected count rates but with systematically larger uncertanties.

The other systematic to be aware of is that the PSF is a function of count rate; significant photon coincidence losses will result in flatter PSFs. A detailed characterization of this effect is pending, but it will result in the CALDB PSFs becoming increasingly obsolete at high count rates. For most small aperture cases, coincidence loss is not an issue in the sense that if an object is bright enough to have significant losses then the 5 arcsec aperture should be used to extract count rates. The PSF is fully-sampled in a 5 arcsec aperture and no aperture correction will be necessary. However the one exception is background limited cases where the sky is bright, which often occurs when observing through the WHITE filter. Currently the aperture correction systematics attached to such ocurrances are unquantified.

It is suggested that the current tool provides a useful 1st order count rate correction for rapid-response analysis of gamma ray bursts and targets of opportunity. However a more rigorous approach is recommended for longer-term analysis activities where aperture corrections are conducted using e.g., curve of growth methods for each individual exposure in a sequence.

PARAMETERS

cntrate [float]

- The count rate (counts/sec) from a SWIFT-UVOT detected point source. Background count rates must have been subtracted prior to running this tool.

ratesig [float]

- The statistical 1-sigma measurement uncertainty attached to the source count rate (counts/sec).

aperad [float]

- The radius (in arcsec) of the CIRCULAR aperture used to extract the count rate.

(fwhmsig = 6.0) [float]

- The systematic rms uncertainty (percent) on the FWHM of the PSF caused by voltage variations during the observing sequence. This is a relatively nebulous quantity until it can be formally characterized by the UVOT team, but a value of 5% is currently recommended.

(filter = WHITE—U—B—V—UVW1—UVM2—UVW2) [string]

- The UVOT filter through which the field was observed. In the general case, the filter information is stored both in the image filename, as it is extracted from the Archive or Quick-Look database, and in the "filter" keyword within the header items of each image or event table. At present the WHITE PSF is unavailable, so the B PSF is used if WHITE is requested.

(psffile = "CALDB") [filename]

- The path+name of a FITS table containing CALDB-compliant PSF data. This is a simple two-column table containing encircled enery fraction (in column REEF) vs PSF radius in arcseconds (in column RADIUS). If a FITS HDU (extension) number is provided, the tool will open that HDU directly and use its contents. If no HDU is provided, the tool will search each extension in the FITS file until the keyword "filter" matches argument 'filter' above. However, the easiest way to use this function is to download the UVOT caldb from http://heasarc.gsfc.nasa.gov/docs/heasarc/caldb/swift. If 'CALDB' is then provided as the argument to &quote;psffile", software will select the correct file and extension for you.

aperate [float]

- The aperture corrected count rate (counts/sec). This is not an input argument. While the corrected count rate is reported to STDOUT at the end of the task, it is convenient to have this quantity stores as an argument. This can be retrieved by the user or scripts with the pget task, e.g., 'pget uvotapercorr aperate'.

apersig [float]

- The 1-sigma error on the aperture corrected count rate (counts/sec). This also is not an input argument. It can be retrieved by the user or scripts with the pget task, e.g., 'pget uvotapercorr apersig'.

(chatter = 1) [enumerated integer]

- Standard HEAsoft chatter parameter (1-5) controlling the verbosity of the task. Setting 1 is mute except for the final result reported to STDOUT, while setting 5 is the most wordy.

EXAMPLES

The following examples illustrate running uvotapercorr

1. run uvotapercorr prompting for all mandatory arguments:

- uvotapercorr

2. run uvotapercorr specifying all control arguments on the command line:

- uvotapercorr cntrate=20.3 ratesig=1.2 aperad=3.0 fwhmsig=6.0 filter='WHITE' psffile='CALDB' chatter=5

SEE ALSO

uvotsource

LAST MODIFIED

July 19, 2007

5.3 UVOTASPCORR

NAME

uvotaspcorr - find aspect corrections for UVOT sky images

USAGE

uvotaspcorr input=<directory> output=<directory>

DESCRIPTION

This tool processes an Swift observation attempting to find aspect corrections for UVOT sky images. The aspect corrections can also be applied by setting apply=yes. Note that the input image files must be updatable (writable and uncompressed) to apply the corrections.

PARAMETERS

input [directory]

- Swift observation directory. The structure of a Swift observation directory is
  - sw<obsid>/
    - auxil/sw*sat.fits uvot/
      - image/sw*u*_sk.img

  See also the flat parameter.

  output [directory]

- Name of output directory. This directory will hold the aspect solutions and log files. The sky images are modified in place.

filter = ALL [string] (ALL—V—U—B—UVM2—UVW1—UVW2—WHITE)

- Which UVOT filter(s) to process.

apply [boolean]

- Apply the aspect corrections? Note that the input images files must be updatable (writable and uncompressed) to apply the corrections.

catspec = usnob1.spec [filename]

- Star catalog descriptor.

(starid = NONE) [string]

- Star identification parameters.

(flat = no) [boolean]

- Is directory hierarchy flat? Set flat=yes if all of the input files (attitude file, sky images) are in one (the input) directory.

(chatter = 4) [integer] <1-5>

- Standard HEAdas chatter parameter.

EXAMPLES

The following examples illustrate running uvotaspcorr:

- uvotaspcorr input=12345678012

SEE ALSO

uvotskycorr aspcorr tristarid catspec

LAST MODIFIED

June 2007

5.4 UVOTATTCORR

NAME

uvotattcorr - adjust attitude file with information from UVOT aspect corrections

USAGE

uvotattcorr attfile=<sw12345678012sat.fits> outfile=<uvotatt.fits>

DESCRIPTION

The Swift spacecraft attitude is available from several sources in the telemetry. Attitude knowledge onboard is not as fine as what can be obtained from comparing positions of stars in UVOT images to catalog positions. In addition, when the spacecraft transitions between slewing and pointed observations, the source of attitude information shifts from coarse to fine, resulting in a noticeable jump in the earliest exposures found in particular in GRB AT sequences. The purpose of uvotattcorr is to take the refined position information obtained from 'uvotskycorr' and apply it to the available spacecraft attitude information (which is 'atjumpcorr'ed in the pipeline), creating a more accurate attitude file. The pipeline attempts to find an aspect solution for each image in a UVOT observing sequence. The results of these corrections can then applied to the existing attitude file. Since a sequence can contain gaps between images (and thus aspect solutions), and the exposure times

can vary from tens of seconds to thousands of seconds, the offsets are first interpolated before being applied to the existing attitude. Corrections are considered valid for the snapshot in which they occur. During a snapshot which has no aspect corrections, no chance is made to the attitude. Otherwise, the correction at a given time is found by linearly iterpolating between (or extrapolating) the nearest known correction(s).

The solutions to the aspect corrections are found in the housekeeping file sw*uac.hk. At present, the pipeline creates a new corrected attitude file, sw*uat.fits, when uvotattcorr is run.

PARAMETERS

attfile [filename]

- Swift attitude file (normally sw<obsid>sat.fits).

corrfile [filename]

- UVOT aspect corrections file (normally sw<obsid>uac.hk).

outfile [filename]

- Attitude file corrected for UVOT.

(deltafile = NONE) [filename]

- Delta attitude file. The special value NONE indicates to not create this output.

(chatter = 3) [integer] <1-5>

- Standard HEAdas chatter parameter.

EXAMPLES

The following examples illustrate running uvotattcorr:

- uvotattcorr infile=sw12345678012sat.fits corrfile=sw12345678012uac.hk \
  - outfile=uvotatt.fits

SEE ALSO

uvotskycorr, uvotaspcorr, attjumpcorr

LAST MODIFIED

June 2007

5.5 UVOTBADPIX

NAME

uvotbadpix - create pixel quality maps for images

USAGE

uvotbadpix infile=<filename> outfile=<filename>

DESCRIPTION

This program takes a caldb bad pixel list and creates a pixel quality map for use in uvotmodmap and exposure map generation. This tool checks for missing rows due to packet loss, and optionally check the input image for compression damage (overflows and underflows). uvotbadpix allows the user to

specify the caldb bad pixel list specify a fits (multi-extension) image file for which to generate matching quality maps

PARAMETERS

infile [filename]

- Input image file for which quality maps are created.

badpixlist [filename]

- Input caldb bad pixel table. Specify the name of a file, or "CALDB" to retrieve from the calibration database.

outfile [filename]

- Output quality map file name.

(compress = yes) [boolean]

- If compress = yes then check for compression over or under flows. If compress = no then don't check.

(clobber = no) [boolean]

- If outfile already exists, then "clobber = yes" will overwrite it.

(history = yes) [boolean]

- If history = yes, then a set of HISTORY keywords will be written to the header of the output file to record the value of all the ftcopy task parameters that were used to produce the output file.

(chatter = 1) [integer, 0 - 5]

- Controls the amount of informative text written to standard output. Verbosity increases with larger numbers.

(mode = ql)

- Standard PIL mode parameter.

EXAMPLES

The following example is a typical uvotbadpix run.

- uvotbadpix images.fits swubadpix.fits quality.fits

Where images is the name of the input image files, swubadpix.fits holds the CALDB bad pixel list, and quality.fits is the name of the generated quality map,.

Alternatively the user could just run uvotbadpix and be prompted for the inputs.

LAST MODIFIED

October 2004

5.6 UVOTCENTROID

NAME

uvotcentroid - Locate the centroid of a source on a UVOT image.

USAGE

uvotcentroid image=<filename> srcreg=<filename> confidence=<float> niter=<integer> threshold=<float> subdimsiz=<integer> plot=<boolean> plotdev=<string> cleanup=<boolean> chatter=<enumerated integer>

DESCRIPTION

One of the primary goals of the UVOT instrument is to provide accurate celestial positions for gamma ray burst afterglows. The problem can be separated into three parts i) the conversion from detector pixels to sky coordinates (see the ftool swiftxform), ii) an astrometric adjustment to correct systematic pointing uncertainties in the spacecraft attitude (see uvotaspcorr), and iii) within the statistical limits of the data, locate the centroid of a particular source on the image. It is this last item which is performed by uvotcentroid. Sky coordinate conversion must have been performed before running uvotcentroid. This tool can be used before an astrometric correction, but only to determine a statistical confidence limit. An astrometric correction is required to determine an accurate position. But note well that both items i) and ii) above can only be achieved to a certain accuracy and this systematic uncertainty - which can often exceed the statistical position error

provided by uvotcentroid - is not propagated through the centroiding calculation. The statistical and systematic errors must be combined before reporting a formal position ucertainty.

This tool employs uvotdetect to determine the position of sources. uvotdetect is a wrapper for the sextractor tool which does the fundamental work of detecting and providing source positions; see http://terapix.iap.fr/rubrique.php?id_rubrique=91/. Both point- and extended sources can be centroided and relatively crowded fields can be accommodated. 1-sigma uncertainties are reported by uvotdetect but the method by which sextractor obtains these errors is somewhat nebulous. In order to achieve some confidence in the statistical error, uvotcentroid performs a user-defined number of trials. Before each trial a new, small image is created from the original with the source of interest close to the center. The value of each individual pixel is ajusted by a value chosen at random from the normal cummulative distribution function defined to have a sigma width of the square root of the pixel value. This provides multiple images with identical noise distributions but pixel count distributions varying within the limits defined by the noise. A centoid position for the source in each trial image is recorded and the variance used to estimate confidence limits on the result.

The tool reports back the most-likely RA and Dec of the source, a confidence limit on the position and the number of trials in which the source could not be detected (useful for faint soures). Optionally a plot may be generated to inspect the trial distribution.

If more than one source is located within the subimage, the tool will default to the source nearest to the center of a user-provided region file. The user has the ability to change the size of the sub-field over which uvotcentroid regenerates source images. Small fields have the advantage of less field sources and faster runtimes, at the expense of smaller background statistics. A reasonable subimage size is 20x20 arcsec for optical filter observations and 40x40 arcsec for UV observations.

PARAMETERS

image [string]

- The name and correct path to a standard UVOT FITS sky image. if either are incorrect, the tool will complain that it has found no such image. The name should normally contain the FITS extension hosting the image, either by number, e.g. example.fits+1, or name, e.g. example.fits[EXTNAME]. If neither are supplied the tool will look for an image in the primary extension and use it. If none is found, it will default to the first extension and look again. If none is then found the tall will complain.

srcreg [string]

- The name and correct path to a standard ds9 region file. These regions can often be complex, uvotcentroid will read the first line which begins with 'fk5;' (ignoring any exclusion regions 'fk5:-') and use these coordinates as the center of each subimages it creates. A typical region file looks like this:

Region file format: DS9 version 4.0 # Filename: /Volumes/data/00035934002/uvot/image/sw00035934002u
fk5;circle(76.37975,-4.3054021,5")

The circle radius is ignored by uvotcentroid in this case. The onus is on the user to provide a region file as close to the source of interest as possible. As an absolute minimum, the center of the region must be closer to the interesting source than any other in the image.

confidence [float]

- This is the level of confidence (percent) with which you would like the source position reported. A typical value is 90, but the range is 0 < confidence < 100, exclusive.

niter [integer]

- The number of trials to be performed. The more trials, the more accurate the result. A minimum of 100 successful trials are required to provide a confidence estimate. If less than 100 trails could detect the source,a position will be reported but no error estimate. In these cases, re-run the tool with a larger number of trials. uvotcentroid can be run with just 1 trial which will provide a position with no confidence. Provided an astrometic correction has been peformed, most likely this position will be good to at least 1 arcsec > 99.9% confidence.

threshold [float]

- uvotdetect will only recognise sources if they are detected above this threshold. The units are sigma, assuming a Gaussian noise distribution. This argument must be positive and larger than zero. For weak, marginal source a threshold of 2 is suitable.

subdimsiz [integer]

- The dimensions of the subimage are square and this argument provides the length of one side in arcsec units. The subimage must be large enough to incorporate the source and some background in order for the source to be located.

ra [float]

- This is not an input argument. It is used to store the output right ascension in decimal degrees so that users and scripts can access the results easily, using e.g. from a shell:

'pget uvotcentroid ra'

dec [float]

- This is not an input argument. It is used to store the output declination in decimal degrees so that users and scripts can access the results easily, using e.g. from a shell:

'pget uvotcentroid dec'

conflimit [float]

- This is not an input argument. It is used to store the output location uncertainty in arcseconds so that users and scripts can access the results easily, using e.g. from a shell:

'pget uvotcentroid conflimit'conflimit is only useful if you get the confidence level associated with it:

- 'pget uvotcentroid confidence'

plot [boolean]

- If yes, the tool will plot a summary of the result and trials. Two windows are plotted. The left one contains a histogram of the distance in arcsec between the most-likely source position' and all of the trials. The confidence range requested by the user is cast as a region colored yellow behind the histogram. The most-likely RA and Dec, the confidence limit and the number of positive source detections are provided in a dialog box in the top right of this window. The right hand plot shows the distribution of trial positions in RA and Dec space, relative to the most-likely position. The confidence limit is represented as a yellow circle in the background. Each source position is marked by a blue cross symbol. The size of each cross is arbitrary and has no relation to a measured error.

plotdev[boolean]

- The device through which you want to make the plot. Available options will depend on your PGPLOT installation but common usages are: /XS - xserver on your monitor /GIF - gif file /PS - postscript document

(chatter = 1) [enumerated integer]

- Standard HEAdas chatter parameter (0-5) controlling the verbosity of the task. Setting 0 is mute except for the final result reported to STDOUT, while setting 5 is the most wordy.

EXAMPLES

The following examples illustrate running uvotcentroid

1. run uvotcentroid and get prompted for all mandatory arguments:

- uvotcentroid

2. run uvotcentroid specifying all required control arguments on the command line:

- uvotcentroid image=sw00035934002uuu_sk.img+1 srcreg=srcreg confidence=90 niter=1000 threshold=3 subdimsiz=20 chatter=5

3. run uvotcentroid specifying all control arguments on the command line, plotting output to a gif file and retaining all intermediate files:

- uvotcentroid image=sw00035934002uuu_sk.img+1 srcreg=srcreg confidence=90 niter=1000 threshold=3 subdimsiz=20 plot=y plotdev=/gif cleanup=n chatter=5

LIMITATIONS

The subimage and region file must always be fully contained within the sky region sampled by the input image. Cases where the source lies very close to RA = 0h or the celestial poles have not been tested.

SEE ALSO

uvotdetect

LAST MODIFIED

July 19, 2007

5.7 UVOTCOINCIDENCE

NAME

uvotcoincidence - performs coincidence loss correction

USAGE

uvotcoincidence infile=<filename> coinfile=<filename> ratecol=<string> errcol=<string> frametime=<real> deadtimecorr=<boolean> history=<bool> chatter=<enumerated integer>

DESCRIPTION

Given a count rate in units of counts per second, and a 1-sigma error in the same units, uvotcoincidence will find the corresponding coincidence loss correction factors and rates. Normally I/O is performed through a FITS table which must be prepared by the user. Requirements are to provide a minimum of two columns with arbitrary names containing numeric values. One column contains a count rate, the other contains the associated 1-sigma error. The table may contain any number of rows or other columns, the task will perform the conversion on all data rows. The output is written to the same FITS table. Five columns are created or updated:

- COI_FACTOR correction factor COI_FACTOR_ERR 1-sigma error associated with COI_FACTOR COI_RATE CoI corrected rate in units of count/s COI_RATE_ERR 1-sigma error associated with COI_RATE SATURATED boolean; true indicates saturation

PARAMETERS

infile [filename]

- FITS input table containing at least two numeric columns, one containing count rates in units of counts per second, the other containing 1-sigma measurement error in the same units. Output data will be appended to the same table. If infile=NONE, the values of the ratecol and errcol parameters are operated on directly.

coinfile [filename]

- Name of file containing coincidence loss correction calibration. The special value CALDB causes the CALDB to be queried for the proper file.

(ratecol = RATE) [string]

- Name of the table column in infile containing the count rates. Each row must contain a numeric value. If infile=NONE, pass a single rate through this parameter.

(errcol = RATE_ERR) [string]

- Name of the table column in infile containing the 1-sigma count rate error. Each row must contain a numeric value. If infile=NONE, pass a single rate error through this parameter.

(frametime = DEFAULT) [string]

- Frame time [s]. Used for calculating dead time correction. The special value DEFAULT results in the value of the FRAMTIME keyword being used if it is present in the source table, otherwise, the value 0.0110322 [s] is used.

(deadtimecorr = yes) [boolean]

- Input already dead time corrected? deadtimecorr=yes means that the dead tim e correction has already been applied in the exposure time. deadtimecorr=no mea ns that uvotcoincidence will apply the dead time correction.

(history = yes) [boolean]

- If yes, a HISTORY keyword will be written to the header of infile containing the uvotcoincidence arguments.

(chatter = 1) [enumerated integer]

- Standard HEAdas chatter parameter (1-5) controlling the verbosity of the task. Setting 1 is mute, while setting 5 is the most wordy.

EXAMPLES

The following examples illustrate running uvotcoincidence

1. run uvotcoincidence specifying source list and rate columns on the shell command line. countrates.fits is a FITS file containing a table in HDU number 1. (note that the first extension in the file, often an empty primary extension, is designated HDU number 0, by convention). The tool will look for table columns names RATEX (containing the source count rate) and RATEX_ERR (the 1-sigma error attached to RATEX) to propogate through the magnitude and flux conversion.

- uvotcoincidence infile=countrates.fits+1 ratecol=RATEX errcol=RATEX_ERR

2. run uvotcoincidence specifying a single rate and error for conversion on the command line:

- uvotcoincidence infile=countrates.fits+1 ratecol=13.4 errcol=1.13

3. run uvotcoincidence specifying the coincidence loss calibration file on the command line:

- uvotcoincidence infile=countrates.fits+1 \ coinfile=/caldb/data/swift/uvota/bcf/phot/swucntcorr20041120v

SEE ALSO

uvotdetect, uvotflux, uvotsource, uvotmaghist, uvotproduct, uvotapercorr

LAST MODIFIED

May 21, 2007

5.8 UVOTDETECT

NAME

uvotdetect - detect sources in an UVOT image

USAGE

uvotdetect infile=<filename> outfile=<filename>

DESCRIPTION

This tool runs SExtractor to detect sources in a UVOT image. By default. the output FITS table includes absolutely calibrated magnitudes (MAG) and fluxes (FLUX). The resulting columns *_ERR in the output (e.g., RA_ERR, Y_ERR, MAG_ERR, FLUX_ERR) give 1-sigma errors.

PARAMETERS

infile [filename]

- Input filename parameter (UVOT Sky FITS image). If the input file has multiple extensions, the user must specify on which extension to operate. See the HEASOFT help on filenames for specifying the extension (in section B.2).

outfile [filename]

- Output filename parameter. FITS file containing table(s) of sources corresponding to input image(s).

expfile [filename]

- Exposure map file name or NONE. This value is passed to SExtractor if the sexfile parameter is DEFAULT.

(sexfile = DEFAULT) [filename]

- SExtractor configuration file or DEFAULT. The user can control nearly every aspect of SExtractor by providing this file. There are certain parameters which uvotdetect overrides for its own purposes: CATALOG_NAME, CATALOG_TYPE.

threshold [real]

- Detection threshold. This value is passed to SExtractor if the sexfile parameter is DEFAULT.

(zerobkg = 0.03) [boolean]

- Maximum fraction of nulls in image to allow SExtractor to calculate background If the fraction of nulls in the 'middle' third of the image exceeds this limit, the background mean and sigma will be calculated externally using ftstat and passed as constants to SExtractor. This is sometimes needed because the SExtractor calculation performs poorly at low counts. Set ZEROBKG =-1 to always use the external calculation.

detected [integer]

- This output parameter gives the number of sources detected or -1 in the event of an error.

(plotsrc = no) [boolean]

- Display detected source positions in the ds9 viewer.

(expopt = BETA) [string]

- Controls processing when an exposure map is provided. The BETA method better handles the edges of single images . Specify expopt=ALPHA to use the default behavior prior to June 2007, which is generally better for co-added images.

(calibrate = Y) [boolean]

- If true, then the coincidence loss correction and absolute calibration is applied to the SExtractor output.

(clobber = no) [boolean]

- Standard HEAdas clobber parameter. If true, the output file will be overwritten.

(cleanup = yes) [boolean]

- Remove temporary files at end of run.

(chatter = 1) [integer]

- Standard HEAdas chatter parameter.

EXAMPLES

The following examples illustrate running uvotdetect

1. run uvotdetect prompting for options

- uvotdetect

2. run uvotdetect specifying input and output filenames on command line. Specify a threshold of 2 sigma above the background

- uvotdetect infile=image.fits outfile=source.fits threshold=2

SEE ALSO

SExtractor

LAST MODIFIED

June 2007

5.9 UVOTEVGRISM

NAME

uvotevgrism - filter a UVOT grism event list and determine wavelength

USAGE

uvotevgrism infile=<filename> outfile=<filename>

DESCRIPTION

This tool operates on grism UVOT level 2 event files. The output file matches the input except for the EVENT extension which is updated to include only those events which fall in the dispersion region. The dispersion region is a rectangle defined in terms of the source position (0th order right ascension and declination), the angle of the dispersion axis (measured clockwise from the +X axis), pixel width, and wavelength bounds for position on the dispersion axis.

PARAMETERS

infile [filename]

- UVOT level 2 events file (see SSC UVOT Data Handbook).

outfile [filename]

- UVOT level 2 events file (see SSC UVOT Data Handbook).

wavefile [filename]

- Pixel to wavelength conversion calibration file.

teldeffile [filename]

- Telescope definition file.

attfile [filename]

- Attitude history file.

(attlim = 32) [real]

- Attitude extrapolation limit [s]

ra [real]

- 0th order right ascension [degrees]

dec [real]

- 0th order declination [degrees]

(uvang = 207) [real]

- Angle from +X axis to UV dispersion axis [degrees]

(uvwid = 30) [real]

- Width of UV 1st order dispersion region [pixels]

(uvmin = 300) [A]

- Wavelength corresponding to lower end of UV extraction region.

(uvmax = 2000) [A]

- Wavelength corresponding to upper end of UV extraction region.

(vang = 207) [real]

- Angle from +X axis to V dispersion axis [degrees]

(vwid = 30) [real]

- Width of V 1st order dispersion region [pixels]

(vmin = 2000) [A]

- Wavelength corresponding to lower end of V extraction region.

(vmax = 6000) [A]

- Wavelength corresponding to upper end of V extraction region.

(chatter = 1) [enumerated integer]

- Standard HEAdas chatter parameter. A range of 0 to 5 is enforced by the min and max fields in the parameter file.

(clobber = no) [boolean]

- Standard HEAdas clobber parameter.

(history = yes) [boolean]

- Standard HEAdas history parameter.

EXAMPLES

The following example illustrates running uvotevgrism.

1. run uvotevgrism specifying input and output event files, attitude file, and source position (ra and dec) on the command line

- uvotevgrism infile=sw00070911011ugvb1.unf.fits outfile=evgrism.fits \
  ra=286.16 dec=08.17 attfile=sw00070911011snone.att.fits

SEE ALSO

LAST MODIFIED

May 2004

5.10 UVOTEVTLC

NAME

uvotevtlc - make a background subtracted light curve from UVOT event data

USAGE

uvotevtlc infile outfile srcreg bkgreg

DESCRIPTION

This task makes a light curve from UVOT event files. It performs region selection and background subtraction. The user can supply custom time bins using the gtifile parameter. Both source and background regions must be supplied. Complex region files are permitted.

The user can specify one file on the command line, or an "@" style batch file listing multiple files. Event files taken with different UVOT filters should never be mixed in one call to uvotevtlc.

To make sure the UVOT events have been converted to sky positions using the best available Swift spacecraft attitude, the task 'coordinator' should first be run on the event file, with an aspect-corrected attitude file that contains the ajumpapp keyword set to 'T'.

This task calls the batbinevt task to actual perform the light curve binning. The user has a choice of binning styles via the "timebinalg" parameter. Binning can be either "U"niform, indicating constant bin sizes, or "G"TI, meaning the user choses the bin edges explicitly with a good time interval file. Each separate good time interval will be turned into one output light curve time bin. The good time interval file is also used for time-based filtering of UVOT events.

Output lightcurves contain the keyword TIMEPIXR which indicates the reference point for light curve time bins. For light curves, the start, stop and center times of each bin can be calculated as:

- TIME_START = TIME - TIMEPIXR*TIMEDEL TIME_STOP = TIME + (1-TIMEPIXR)*TIMEDEL TIME_CENT = TIME + (0.5-TIMEPIXR)*TIMEDEL

Since for uvotevtlc TIMEPIXR should always be 0.5, the TIME column marks the center of each bin, and TIME_START and TIME_STOP are one-half of a bin width away. If variable size time bins are requested, then TIMEDEL will be a column instead of a keyword, but the above expressions still hold.

Region-based filtering is done using standard region files (this uses the region filtering capability built into CFITSIO). The user should supply both source and background regions. uvotevtlc computes the net rate (RATE) by subtracting the mean background level, derived from the background region, from the source rate, derived from the source region. The background rate (RATE_BKG) is reported in units of counts per second per square arcsec, and is scaled by the source region size before subtraction. The source and background region sizes are stored in the REGAREA and BREGAREA keywords, respectively (in units of square arcsec). The region areas are computed empirically by testing how many UVOT pixels in the entire field of view successfully pass the filtering criteria.

If the parameter 'uvotmag' is set to YES, then UVOT count rates are converted to magnitudes using the uvotcoincidence and uvotflux tasks.

uvotevtlc makes several scratch data files. Users should have enough free space to accomodate these files. The actual size of the scratch data varies depending on the size of the source and background region, but should never be more than the size of the original event files (when they are uncompressed).

PARAMETERS

infile [filename]

- The name of the input events file, or an @-file listing the event files to be entered.

outfile [filename]

- The name of the output light curve file

srcreg [filename]

- The name of the source region file. Complex regions are permitted.

bkgreg [filename]

- The name of the background region file. Complex regions are permitted.

(gtifile = "NONE") [string]

- The name of a GTI file. When timebinalg='g', this file should be a GTI which contains custom time bins, one bin per good time interval. No matter what the time binning algorithm, gtifile is always used for time filtering of the events.

(timedel = 0) [real]

- The time bin size in seconds if timebinalg='u'. A bin size of "0" indicates the full dataset should be summed.

(timebinalg = "g") [string]

- The time binning algorithm. Either "u" for uniform time binning (constant time bin size specified by 'timedel'); and "g" for custom time binning (specified by 'gtifile').

(tstart = "INDEF") [string]

- Optional global start time (MET seconds), or INDEF to use the input file's start time.

(tstop = "INDEF") [string]

- Optional global stop time (MET seconds), or INDEF to use the input file's stop time.

(uvotmag = "YES") [boolean]

- If YES, then run the 'uvotcoincidence' and 'uvotflux' tasks to compute calibrated magnitudes for the light curve samples. The default/CALDB parameters are used in the calls to these tasks, and deadtimecorr is set to "NO" (since the raw rates produced by uvotevtlc are not corrected for deadtime).

(frametime = "DEFAULT) [string]

- UVOT frame time, in seconds, or DEFAULT. This parameter is used by uvotcoincidence to estimate dead time and coincidence loss.

(uvotmag_params = "NONE") [string]

- Any additional parameters to be used when calling the uvotmag task. This string must specify only uvotmag hidden parameters, excluding 'deadtimecorr', which is always set to NO.

(clobber = NO) [boolean]

- If the output file already exists, then setting "clobber = yes" will cause it to be overwritten.

(chatter = 2) [integer, 0 - 5]

- Controls the amount of informative text written to standard output. Setting chatter = 1 produces a basic summary of the task actions; chatter = 2 (default) additionally prints a summary of input parameters; chatter = 5 prints debugging information.

(history = YES) [boolean]

- If history = YES, then a set of HISTORY keywords will be written to the header of the specified HDU in the output file to record the value of all the task parameters that were used to produce the output file.

EXAMPLES

1. Creates a light curve using custom time bins specified by absolute.gti

- uvotevtlc sw00176918992uvvpo_uf.evt event.lc src.reg bkg.reg gtifile=absolute.gti

SEE ALSO

batbinevt

LAST MODIFIED

July 2007

5.11 UVOTEXPCORR

NAME

uvotexpcorr – Swift/UVOT exposure time correction

USAGE

uvotexpcorr sequence=<string> dataloc=<string>

DESCRIPTION

This tools is obsolete and no longer supported.

This script corrects the EXPOSURE keyword in Swift/UVOT files based on the data in the UVOT Exposure Report. The CountRate * (1-Dataloss) from the Exposure Report (log/sw*uir.html[.gz]) is compared to the sum of the counts / EXPOSURE in each extension of each UVOT sw*sk.img[.gz] and sw*rw.img[.gz] file found in the specified directory tree (under uvot/image). Extensions not from image-mode data (ie, EXTNAME not ending in "I") will be analyzed, but NO corrections will be applied to the EXPOSURE keyword regardless of the value of the "allowcorr" parameter. If the tool encounters a previously corrected EXPOSURE keyword it will flag that in the output.

PARAMETERS

sequence [string]

- Sequence number of the data.

dataloc [string]

- Pathname to the Swift data directory (containing log/ and uvot/ subdirectories).

(threshold = 0.95) [real]

- Specifies the threshold ratio below which the EXPOSURE keyword will be corrected.

(allowcorr = no) [boolean]

- Allow the computed UVOT EXPOSURE correction (if any) to be applied. One might set this to "no" to just run the tool in an informational mode. Even if "yes", corrections will only be applied to extensions with images from image-mode data.

(chatter = 3) [integer]

- Standard HEAdas chatter parameter (1-5) controlling the verbosity of the task. A value of zero (0) supresses all screen output. A value of one (1) will generally only report on "interesting" cases. The default value of three (3) yields basic information on each extension.

(history = yes) [booleanr]

- Standard HEAdas history parameter. If "yes" then any extension which has a corrected EXPOSURE keyword will contain a block of history keywords documenting the runtime parameters for this tool.

NOTES

Any gzipped image file will be unzipped during execution and rezipped when the tool is finished with it.

SEE ALSO

LAST MODIFIED

June 2005

5.12 UVOTEXPMAP

NAME

uvotexpmap - generate exposure maps for UVOT sky images

USAGE

uvotexpmap infile=<filename> outfile=<filename>

DESCRIPTION

This tool generates exposure maps given a UVOT level 2 sky image product. An exposure map is created for each extension in the image file.

PARAMETERS

infile [filename]

- UVOT level 2 sky image file (see SSC UVOT Data Handbook).

badpixfile [filename]

- Bad pixel file. See the Data Handbook for the bad pixel file format. It has extension names corresponding to those in the sky image file. The bad pixel file is created by uvotbadpix.

method [MEANFOV—SHIFTADD]

- Exposure map method. If method is MEANFOV, then the exposure map is based on a single attitude (the attitude at the same time used when projecting the image onto the sky). If method is SHIFTADD, then the processing follows the attitude changes during the exposure as given by the aspect following packets and/or attitude data and the trackfile parameter should be set.

attfile [filename]

- Attitude history file (in coordfits/ATTFILE format).

teldeffile [filename]

- Telescope definition file (coordfits/TELDEF format).

outfile [filename]

- Exposure map output file. One extension for each image extension in the input sky image file. See Data Handbook for details.

maskfile [filename]

- Name of optional output mask file which can be used to screen out outer pixels to suppress edge-effects. This maskfile can be passed as an input parameter to uvotimsum.

(masktrim = 0) [integer]

- Mask trimming control. This parameter is only used if maskfile is not NONE. A positive value gives the number of pixels to be trimmed from the top/bottom/left/right or the mask. A negative value trims the mask based on the attitude variation during the exposure. The special value 0 is treated as -1 if method is MEANFOV, or 1 if method is SHIFTADD.

(trackfile = NONE) [filename]

- If method is SHIFTADD, this gives the name of the aspect following housekeeping file (sw*uaf.hk).

(attdelta = 100) [real]

- Maximum change in attitude exposure map increments [arcsec].

(aberration = no) [boolean]

- Account for velocity aberration?

(chatter = 1) [enumerated integer]

- Standard HEAdas chatter parameter. A range of 0 to 5 is enforced by the min and max fields in the parameter file.

(cleanup = yes) [boolean]

- Clean up temporary files?

(clobber = no) [boolean]

- Standard HEAdas clobber parameter.

(history = yes) [boolean]

- Standard HEAdas history parameter.

EXAMPLES

The following examples illustrate running uvotexpmap

1. run uvotexpmap prompting for options

- uvotexpmap

2. run uvotexpmap in command line specifying sky image input, bad pixel and attitude files, and create a mask that screens out edge-effects

- uvotexpmap infile=skyimages.fits badpixfile=badpixels.fits attfile=attitude.fits maskfile=mask.fits

SEE ALSO

expomap, uvotbadpix, uvotimsum

LAST MODIFIED

March 2008

5.13 UVOTFLATFIELD

NAME

uvotflatfield - performs flat-fielding on UVOT images

USAGE

uvotflatfield infile=<filename>

DESCRIPTION

This tool divides RAW images by the UVOT flat field calibration product.

PARAMETERS

infile [filename]

- Input image file. If no extension is specified, all extensions are processed; otherwise, only the specified extension is processed.

outfile [filename]

- Output file name. See clobber parameter.

flatfile [filename]

- Flat field calibration file or CALDB.

(clobber = no) [boolean]

- Overwrite output file if it exists?

(cleanup = yes) [boolean]

- Remove temporary files?

(history = yes) [boolean]

- Write parameter history?

(chatter = 1) [enumerated integer]

- Standard HEAdas chatter parameter.

EXAMPLES

The following example illustrates running uvotflatfield specifying the input and output file names on the command line:

- uvotflatfield infile=images.fits outfile=flat.fits

SEE ALSO

LAST MODIFIED

June 2004

5.14 UVOTFLUX

NAME

uvotflux - converts source count rates to magnitudes and flux

USAGE

uvotflux infile=<filename> zerofile=<filename> filter=<string> syserr=<bool> ratecol=<string> errcol=<string> history=<bool> chatter=<enumerated integer>

DESCRIPTION

Given a count rate in units of counts per second, and a 1-sigma error in the same units, uvotflux will convert the rate into an instrumental magnitude (based on UVOTs own filter system) and fluxes in units of erg/s/cm{}2/Angstrom. The count rate error is propagated through each calculation. Magnitudes, m, are determined from: m = ZPT - 2.5 * log_10(C)

where C is a count rate and ZPT is a zero point, appropriate to a specific filter. The zero-points have been calibrated from standard photometry fields and assume all sources have Vega-like spectra. The zero points are stored as FITS keywords within the UVOT CALDB (see http://heasarc.gsfc.nasa.gov/docs/heasarc/caldb/swift/docs/uvot/ index.html). Fluxes, F_lam, are calculated by a applying a straightforward multiplicative factor to the count rate:

F_lam = FCF * C

FCF is also filter specific and stored as CALDB keywords. It's units are erg/cm{}2/angstrom/count and, again, the conversion factors assume a Vega-like spectrum.

The user is required to provide the filter through which the observations were obtained so that the tool knows which zero point to extract from the CALDB.

I/O is performed through a FITS table which must be prepared by the user. Requirements are to provide a minimum of two columns with arbitrary names containing numeric values. One column contains a count rate, the other contains the associated 1-sigma error. The table may contain any number of rows or other columns, the task will perform the conversion on all data rows. The output is written to the same FITS table. Four new columns are written:

- MAG – instrumental magnitude
  MAG_ERR – 1-sigma error associated with MAG
  FLUX – flux in units of erg/s/cm**2/Angstrom
  FLUX_ERR – 1-sigma error associated with FLUX in the same units

Each zero point comes with an associated measurement error which is also stored as a CALDB keyword, e.g. ZPEV, ZPEWHITE and ZPEUVW1. To perform absoltute photometry the zero point error should be included in the calculation. However if one is interested in detecting relative structure in time series photometry the zero point error would add an unwanted level of error to each point along the curve. In such cases it is appropriate to ignore the zero point error in the magnitude and flux calculations. Use the syserr argument to toggle between the two cases above.

There are prerequisites before running uvotflux. For a sensible conversion, any background or other contaminants must have been removed from the source count rate. Also the extracted count rates need to have been corrected for coincidence losses (see the ftool uvotcoincidence). Without a coincidence correction, magnitudes and fluxes will be systematically underestimated. The zero-points stored in the CALDB are consistent only for a specific extraction aperture size. Count rates should either be obtained from an aperture of the same size, or corrected for aperture size using e.g. a curve-of-growth point spread function correction using e.g. uvotapercorr. The relevant aperture sizes are stored as keywords in the same CALDB product as the zero points, e.g. APTB, APTWHITE and APTUVW1. At the time of writing, all default apertures are circles with radius of 10 unbinned detector pixels (approx. 5 arcsec). Finally the tool assumes that the deadtime correction (a small fraction due to detector readout) has already been applied to the count rates.

uvotflux may be run as a stand-a-lone ftool, but was designed as a workhorse to be used internally by UVOTs photometry tools uvotsource, uvotflux and uvotproduct. It will also perform on source table created by the source detection tool uvotdetect.

PARAMETERS

infile [filename]

- FITS input table containing at least two numeric columns, one containing count rates in units of counts per second, the other containing 1-sigma measurement error in the same units. Output data will be appended to the same table.

zerofile [filename]

- Name of file containing photometric zero points. The majority of application require the most up-to-date data in the CALDB, and for these instannces simply providin zerofile=caldb will allow the tool to locate and use the caldb data.

(filter = WHITE—U—B—V—UVW1—UVM2—UVW2) [string]

- The UVOT filter through which the sources were observed. Alternatively, the user can specify filter=default which will ask the tool to search for a keyword named 'FILTER' in the header items of infile. If a valid filter is found, it will be adopted.

(syserr = no) [string]

- This refers to the systematic error associated with the zero point calibration. Choose syserr=yes if you want the absolute magnitude or flux of a source. Choose syserr=no if you are interested in analyzing relative structure within time series photometry.

(ratecol = RATE) [string]

- Name of the table column in infile containing the count rates. Each row must contain a numeric value.

(errcol = RATE_ERR) [string]

- Name of the table column in infile containing the 1-sigma count rate error. Each row must contain a numeric value.

(history = yes) [boolean]

- If yes, a HISTORY keyword will be written to the header of infile containing the uvotflux arguments.

(chatter = 1) [enumerated integer]

- Standard HEAdas chatter parameter (1-5) controlling the verbosity of the task. Setting 1 is mute, while setting 5 is the most wordy.

EXAMPLES

The following examples illustrate running uvotflux

1. run uvotflux and wait to be prompted for all arguments.

- uvotflux

2. run uvotflux specifying source list and rate columns on the shell command line. countrates.fits is a FITS file containing a table in HDU number 1. (note that the first extension in the file, often an empty primary extension, is designated HDU number 0, by convention). The tool will look for table columns names RATE (containing the source count rate) and RATE_ERR (the 1-sigma error attached to RATE) to propogate through the magnitude and flux conversion.

- uvotflux infile=countrates.fits+1 ratecol=RATE errcol=RATE_ERR

3. run uvotflux specifying all arguments on the shell command line. Old or adapted zeropoint CALDB files may be pointed to with the zerofile argument and specific filter zeropoints may be requested using the filter argument.

- uvotflux infile=countrates.fits+1 ratecol=RATE errcol=RATE_ERR zerofile=/caldb/data/swift/uvota/bcf/l
  filter=UVW1 syserr=NO history=YES chatter=5

SEE ALSO

uvotdetect, uvotcoincidence, uvotsource, uvotmaghist, uvotproduct, uvotapercorr

LAST MODIFIED

August 21, 2007

5.15 UVOTIMAGE

NAME

uvotimage - creates UVOT level 2 images

USAGE

uvotimage infile=@<filename>

DESCRIPTION

Make UVOT SKY images from RAW images and/or EVENTs. One image is generated for every UVOT exposure, regardless of the mode (image or event). For exposures taken in image mode, uvotimage transforms raw images to sky coordinates using the attitude file and the appropriate telescope definition/alignment (teldef) file. For exposures taken in event mode, sky images are made directly from the event X, Y (RA and Dec) values in the event file. UVOTIMAGE is used in the Swift pipeline to iterate over raw UVOT image and event files to create Level 2 image files.

PARAMETERS

infile [filename]

- Name of input image or event file(s). This can be a comma-delimited list of file names or @<file> where <file> contains the names of the files to process, one per line.

prefix [string]

- Prefix for output files.

attfile [filename]

- Input attitude file.

teldeffile [filename]

- Telescope definition calibration file or CALDB.

alignfile [filename]

- Alignment file or CALDB.

ra [real]

- Nominal R.A. (degrees)

dec [real]

- Nominal Dec. (degrees)

roll [real]

- Nominal roll. (degrees)

(catfile = NONE) [filename]

- UVOT exposure catalog or NONE.

(flatfield = no) [boolean]

- Perform flat fielding? Not implemented.

(mod8corr = no) [boolean]

- Perform modulo 8 noise correction?

(cleanup = yes) [boolean]

- Remove intermediate files?

(history = yes) [boolean]

- Write history keywords?

(clobber = no) [boolean]

- Overwrite existing files?

(chatter = 1) [enumerated integer]

- Standard HEAdas chatter parameter.

EXAMPLES

The following examples illustrate running uvotimage

1. run uvotimage

- uvotimage infile=@files prefix=Q teldeffile=CALDB alignfile=CALDB

SEE ALSO

LAST MODIFIED

June 2007

5.16 UVOTIMGRISM

NAME

uvotimgrism - extract UVOT grism spectra

USAGE

uvotimgrism infile=<filename> outfile=<filename>

DESCRIPTION

This tool extracts the grism dispersed region including the 0th order from a V or UV grism image and calculates a wavelength scale. The source extraction region is a rectangular box in a fixed relationship to the source position. The background regions widths and offsets are controlled by user parameters. The user must supply the sourcex and sourcey parameters giving the 0th order position on the detector in FITS input image coordinates. The use of the ra and dec parameters is not yet implemented, and these must be set to a negative number.

PARAMETERS

infile [filename]

- UVOT V or UV grism detector image file (see SSC UVOT Data Handbook).

outfile [filename]

- Output file for source+background spectrum and extracted source region.

backfile [filename]

- Output file for background spectrum.

(areafile = CALDB) [filename]

- UVOT flux calibration (effective area) file (see SSC UVOT Data Handbook).

(wavefile = CALDB) [filename]

- UVOT grating equation calibration file (see SSC UVOT Data Handbook).

badpixfile = NONE [filename]

- Bad pixel file (as created by uvotbadpix). The special value NONE causes a bad pixel image to be created with no bad pixels.

ra [real]

- Right ascension of source [degrees]. If negative, sourcex/y will be used. Currently sourcex/y MUST be used as the RA/Dec interface is not yet implemented.

dec [real]

- Declination of source [degrees]. Not yet implemented.

(sourcex = 0) [real]

- X centroid of zeroth order in input FITS image [pix]. Note that sourcex/y are only used if ra is negative.

(sourcey = 0) [real]

- Y centroid of zeroth order in input FITS image [pix]. Note that sourcex/y are only used if ra is negative.

(ang = 206.7) [real]

- Angle (CCW) from X axis to 1st order grism [degrees]

(wavemin = 1600) [real]

- Minimum wavelength of wave scale [A]

(wavemax = 1600) [real]

- Maximum wavelength of wave scale [A]

(srcwid = 30) [integer]

- Width of 1st order source extraction region [pixels]

(bkgwid1 = 10) [integer]

- Width of lower background region [pixels]

(bkgoff1 = 10) [integer]

- Separation of lower background region from source region [pixels]

(bkgwid2 = 10) [integer]

- Width of upper background region [pixels]

(bkgoff2 = 10) [integer]

- Separation of upper background region from source region [pixels]

(inreg = DEFAULT) [string]

- Name of region file to create to indicate extraction regions in infile image. The value DEFAULT causes a region file with the same base name as infile and a .reg extension to be created. The value NONE causes no region file to be created. Otherwise, the value of inreg is taken to be the name of the region file to create.

(nsigma = 5) [real]

- N sigma for determining outliers in background.

(history = yes) [boolean]

- Standard HEAdas history parameter.

(cleanup = yes) [boolean]

- Clean up temporary files?

(clobber = no) [boolean]

- Standard HEAdas clobber parameter.

(chatter = 1) [enumerated integer]

- Standard HEAdas chatter parameter.

EXAMPLES

The following example illustrates running uvotimgrism

1. run uvotimgrism on a given level 2 grism image file

- uvotimgrism infile=sw123456789ugv.fits

SEE ALSO

uvotevgrism

LAST MODIFIED

July 19, 2007

5.17 UVOTIMSUM

NAME

uvotimsum - sum UVOT sky images or exposure maps

USAGE

uvotimsum infile=<filename> outfile=<filename> method=<string> pixsize=<float> expmap=<filename> exclude=<string> maskfile=<filename> weightfile=<filename> clobber=<boolean> history=<boolean> cleanup=<boolean> chatter=<integer>

DESCRIPTION

t is sometimes desirable to sum UVOT exposures to improve the signal-to-noise, or to detect faint sources. This tool simplifies summing a multiple extension FITS image file. It determines the range on the sky of the input images, checks that the input image filters match and calculates time related keywords.

The input exposures are assumed to be in SKY coordinates. No attempt is made to align individual exposures to a common frame of reference. Use uvotskycorr to do this. The output pixel size is set to "pixsize" degrees. If "pixsize" is less than or equal to zero then the output pixel

size is equal to size of the largest input pixel. The user can exclude exposures using the "exclude" keyword.

Two rebinning methods are available. GRID assumes that all images are oriented in the same direction. This means that image rotations are ignored during image rebinning. XIMAGE makes no assumptions about image rotation. Both rotation and translation are performed during the rebinning.

This tool will sum exposures with different frame times. If this happens the UVOT photometry tools will not return accurate photometry of sources in the summed images

The primary output HDU contains keywords taken from the primary input HDU. The first extension of the output contains the summed image.

PARAMETERS

infile [filename]

- FITS input image(s). The input extensions should have UVOT image keywords.

outfile [filename]

- Name of output file.

(method = XIMAGE—GRID) [string]

- Summing tool.

(pixsize = 0) [real, deg]

- Output pixel size. If no positive number is specified (the default), the largest input pixel size is used.

(expmap = no) [boolean]

- Sum as exposure maps? Changing the size of pixels does not affect exposure times, but does affect photon counts.

(exclude = DEFAULT) [string]

- Comma delimited list of HDU names or numbers (or numbered ranges) to exclude from the sum. See the help file on filenames for a detailed description of specifying input extensions (section B.2).

The special value ASPCORR: NONE can be used to exclude extensions that do not have the ASPCORR keyword. The special value DUPEXPID can be used to exclude all but the largest of input HDUs sharing the same EXPID- this avoids including IMAGE&EVENT mode exposures twice in the sum. The special value DEFAULT is treated as exclude=ASPCORR: NONE,DUPEXPID.

(maskfile = NONE) [string]

- A FITS file containing one mask per extension matching the extensions in the "input" file. The mask file can be created by running uvotexpmap with the "maskfile" parameter. The mask uses "1"s to indicate a pixel to be included in the summed image and "0"s for pixels to be excluded.

(clobber = no) [boolean]

- Overwrite existing output file?

(clobber = no) [boolean]

- Overwrite existing output file?

(history = yes) [boolean]

- Write parameters to history keywords?

(clobber = no) [boolean]

- Overwrite existing output file?

(cleanup = yes) [boolean]

- Remove temporary files?

(chatter = 3) [enumerated integer]

- Standard HEAdas chatter parameter.

EXAMPLES

1. Run uvotimsum specifying input file and output pixel size, overwriting the output file (which will be prompted for) if necessary

- uvotimsum infile=swuv.img pixsize=0.0001 clobber=yes

2. Run uvotimsum excluding HDU 3, the HDU named RALPH, and HDUs 6-14 from the output sum

- uvotimsum exclude=3,RALPH,6-14

3. Run uvotimsum to produce an exposure map corresponding to the output of example 2

- uvotimsum infile=swuv_ex.img expmap=yes exclude=3,RALPH,6-14

4. Run uvotimsum with a weightfile Say we have an image file with two extensions and want the sum of 2.5 times the first extension and 0.5 times the second extension. Create the file weights.txt with contents

- 1: 2.5 2: 0.5

and issue the command

- uvotimsum infile=swuv_ex.img weightfile=weights.txt

SEE ALSO

farith, performs simple arithmetic operations on two images
uvotexpmap, creates UVOT exposure maps and mask files
ximage, has the ability to interactively sum images

LAST MODIFIED

August 2007

5.18 UVOTLSS

NAME

uvotlss - calculate large scale sensitivity corrections

USAGE

uvotlss infile=<filename> lssfile=<filename> x=<float> y=<float> input=<string> history=<boolean> chatter=<integer> DESCRIPTION

This tool calculates the large scale sensitivity correction (LSS) factor for a position, or updates and applies the LSS correction factors in a table created by uvotsource.

PARAMETERS

lssfile [filename] Name of large scale sensitivity file or CALDB.

input [TABLE—RADEC—IMAGE—DET—MM]

- Input mode. This determines how the infile, x, and y parameters are processed.

infile [filename]

- If input=TABLE, infile must give a FITS table created by uvotsource with DETX, DETY, AP_COI_SRC_RATE, AP_COI_SRC_RATE_ERR columns. If input=RADEC or input=IMAGE, infile must give a FITS sky image which contains WCS keywords for the sky coordinates and WCS D keywords for detector coordinates.

If lssfile=CALDB, infile is used to parameterize the CALDB lookup.

x [real]

- This gives the X coordinate for the input mode. If input=RADEC, x is the source R.A. [deg]. If input=IMAGE, x is the X FITS coordinate in the infile sky image [pixel]. If input=MM, x is the X coordinate in the UVOT TELDEF DET system [mm]. If input=DET, x is the X coordinate in the UVOT TELDEF DET system [pixel].

y [real]

- This gives the Y coordinate for the input mode. See the x parameter for a description of the coordinate system and units for each value of the input parameter.

(history = yes) [boolean]

- Write parameter history? The parameter history is only written for input=TABLE.

(chatter = 1) [enumerated integer]

- Standard HEAdas chatter parameter.

EXAMPLES

1. This example illustrates running uvotlss specifying the table extension to process:

- uvotlss input=TABLE infile=table.fits[2]

2. This example provides a sky image and specifies the source sky position on the command line:

- uvotlss input=RADEC infile=sky.fits[vv123091236I] x=243.435 y=-23.54

SEE ALSO

uvotsource

LAST MODIFIED

October 2007

5.19 UVOTMAG

NAME

uvotmag - obsolete tool

USAGE

Do not use.

DESCRIPTION

uvotmag is obsolete. Its functionality has been reassigned to uvotcoincidence and uvotflux.

PARAMETERS

EXAMPLES

SEE ALSO

uvotcoincidence, uvotflux

LAST MODIFIED

May 31, 2007

5.20 UVOTMAGHIST

NAME

uvotmaghist - make a light curve from the exposures in a UVOT image file

USAGE

uvotmaghist infile=<filename> outfile=<filename> plotfile=<filename> zerofile=<filename> coinfile=<filename>
psffile=<filename> syserr=<boolean> timezero=<float>: ra=<float> dec=<float> srcas=<float>
bkgas=<float> srcreg=<filename> bkgreg=<filename> exclude=<string> frametime=<float>
nsigma=<float> method=<string> logtime=<boolean> plotcol=<string> clobber=<boolean>
cleanup=<boolean> history=<boolean> chatter=<integer>

DESCRIPTION

This tool makes a history of one of the quantities computed by uvotmaghist (usually magnitude) of a source versus time. It does this by doing photometry on a source in every exposure in the input UVOT SKY image file (sw*.img). This software calls uvotsource to do photometry. The output quantites are the same as for uvotsource. An optional plot is produced that shows the light curve of the source. The "plotcol" parameter controls which output column to plot as a function of time. See the fhelp for uvotsource for details on how the photometry is done, on how to use the photometry control keywords, and on the output information.

The "timezero" parameter is used to specify the start time for output time column. If it is set to "0" then the TRIGTIME keyword, if present in the input FITS file header, is used as the start time. This option is intended to produce light curves relative to the BAT trigger time of a gamma-ray burst. "Timezero" is given in MET. The "logtime" keyword controls whether or not the time axis of the plot is in log or linear units.

The source and background regions can be specified as standard ftools or ds9 region files (via the "srcreg" and "bkgreg" parameters), or as coordinates (in decimal degrees)

Exposures can be excluded using the "exclude" keyword. See uvotimsum for details. The errors in the magnitudes and the flux densities are the statistical errors based on Poisson errors in the count rates. Systematic errors due to uncertainties in the photometric zero points and flux conversion factors can be included by setting "syserr" to "yes".

PARAMETERS

infile [filename]

- Input image file. All extensions (not the primary HDU) are analyzed, except those specifed by the <exclude> keyword.

outfile [filename]

- Output magnitude history file name. See clobber parameter.

plotfile [filename]

- Output GIF plot of magnitude history or NONE.

zerofile [filename]

- Name of zero points file or CALDB.

coinfile [filename]

- Name of coincidence loss correction file or CALDB.

psffile [filename]

- Name of point-spread function file or CALDB.

(syserr = NO) [boolean]

- Set to YES to include systematic errors in the output error calculation.

(timezero = 0) [real]

- The time at the left edge of the horizontal (time) axis in the magnitude history plot. The special value 0 results in the time being determined based on the input. If the input contains the TRIGTIME keyword, its value is used, otherwise, the earliest start time of the processed images is used.

(ra [real])

- Source RA [deg]. See the examples for specifying regions.

(dec [real])

- Source DEC [deg]. See the examples for specifying regions.

(srcas = 3) [real]

- Source region radius [arcsec]. See the examples for specifying regions.

(bkgas = 3) [real]

- Background region radius [arcsec]. See the examples for specifying regions.

(srcreg = NONE) [string]

- Source region file or NONE. See the examples for specifying regions.

(bkgreg = NONE) [string]

- Background region file or NONE. See the examples for specifying regions.

(timezero = 0) [real]

- The time at the left edge of the horizontal (time) axis in the magnitude history plot. The special value 0 results in the time being determined based on the input. If the input contains the TRIGTIME keyword, its value is used, otherwise, the earliest start time of the processed images is used.

(exclude = NONE) [string]

- A comma-delimited list of HDUs to exclude from processing. Elements of the list can be HDU names or (0-based) numbers.

(frametime = 0.0110329) [real]

- Frame time [s]. Used for calculating LOW_LIM_RATE and passed to uvotflux.

(nsigma = 3.0) [real]

- Required significance of faintest source.

(logtime = yes) [boolean]

- Whether to plot the time axis using log or linear scale.

(clobber = no) [boolean]

- Overwrite output file if it exists?

(cleanup = yes) [boolean]

- Remove temporary files?

(history = yes) [boolean]

- Write parameter history?

(chatter = 1) [enumerated integer]

- Standard HEAdas chatter parameter.

EXAMPLES

1. This example illustrates running uvotmaghist specifying the input and output file names and source position on the command line.

- uvotmaghist infile=images.fits outfile=maghist.fits ra=286.4 dec=-8.16

2. This example specifies source and background region files:

- uvotmaghist infile=images.fits srcreg=source.reg bkgreg=back.reg

3. This example specifies how to include systematic errors and exclude extension 3.

- uvotmaghist infile=images.fits srcreg=source.reg bkgreg=back.reg syserr=YES exclude=3

4. This example specifies how to produce a plot with a linear time axis starting at 2007-06-14T12:34:56.

- uvotmaghist infile=images.fits plotfile=lightcurve.gif srcreg=source.reg bkgreg=back.reg timezero=203517297 logtime=NO

SEE ALSO

uvotflux, uvotapercorr, uvotcoincidence, uvotsource, uvotimsum, uvotdetect

LAST MODIFIED

August 3 2007

5.21 UVOTMODMAP

NAME

uvotmodmap - correct images for modulo-8 spatial fixed-pattern noise

USAGE

uvotmodmap infile badpixfile outfile

DESCRIPTION

This tool takes an image file and matching quality image (bad pixel map) in order to correct the modulo-8 spatial fixed pattern noise. The tool outputs a corrected image file and optionally a map of the corrections. The quality image is expected to match the image file in number of extensions and dimensions of each image.

The final stage of the UVOT detector is a 256 by 256 pixel CCD. Individual events are centroided to one eighth of a physical CCD detector pixel by the onboard electronics. However, the centroiding

algorithm distributes the photons unevenly within each physical CCD pixel, corresponding to 8 by 8 image pixels or a single mod-8 tile.

The tool first steps a box of size ncell across the image, and within the box a sigma clipping algorithm is used to mask out sources more significant than nsig and any surrounding pixels affected by coincidence loss. Then the average of all remaining mod-8 tiles within a sliding box of size ncell is computed and used to produce the mod-8 map.

The algorithm resamples the image to give each image pixel equal area within a mod-8 tile. For each mod-8 tile in the mod-8 map, the pixel x-boundaries are determined such that within each row of pixels the counts/unit pixel area is the same for each pixel. This is then taken as the spatial layout of the pixel x-boundaries in the same mod-8 tile in the science image and the science image is resampled to a grid of evenly spaced pixels. Then the y-boundaries of each row are determined such that each row has the same number of counts/unit pixel area. This is then taken as the spatial layout of pixel y-boundaries in the same mod-8 tile in the science image, which is then resampled to a grid of evenly spaced rows. The same procedure is then repeated with the order reversed (remapping in y followed by remapping in x) and the final corrected science image is made from an average of the two resamplings. This is to ensure that the redistribution of photons is performed in an identical fashion in x and y.

PARAMETERS

infile [filename]

- Input FITS file containing images to which modulo-8 correctioed will be applied.

badpixfile [filename]

- Quality image file (FITS) containing bad pixel maps corresponding to each of the images found in infile. See uvotbadpix.

outfile [filename]

- Output image file name. The output file will have the same structure as the input file.

mod8file [filename]

- Mod8Map output image file name.

mod8product [boolean]

- If mod8product=yes, then write out Mod8Map. If mod8product=no, then do not write out Mod8Map.

nsig [integer]

- Significance level for sigma clipping. Suggested value 3.

ncell [integer]

- Size of sliding box in units of 8 pixels. Suggested value 16.

subimage [boolean]

- Apply correction to only part of the input image? The xmin/xmax/ymin/ymax parameters are only prompted for if subimage=yes.

xmin [integer]

- Sub-image X min.

xmax [integer]

- Sub-image X max.

ymin [integer]

- Sub-image Y min.

ymax [integer] Sub-image Y max.
(clobber = no) [boolean]

- Control whether existing output files will be overwritten.

(history = yes) [boolean]

- If history=YES, then a set of HISTORY keywords will be written to the header of the output file to record the task parameters.

(chatter = 1) [integer] <1-5>

- Standard HEAdas chatter parameter.

EXAMPLES

Here is a typical uvotmodmap run:

- uvotmodmap infile=input.fits badpixfile=quality.fits outfile=output.fits

Alternatively, you could just run uvotmodmap and be prompted for the inputs.

SEE ALSO

uvotbadpix

LAST MODIFIED

June 2007

5.22 UVOTPICT

NAME

uvotpict - create a finding chart image

USAGE

uvotpict

DESCRIPTION

uvotpict retrieves an image of a particular region of the sky then annotates it with information from a source list.

The input source file must have the uvotstarid output source list format.

PARAMETERS

infile [filename]

- Name of the FITS file containing the source list.

outfile [filename]

- Name for output file.

ra [real, degrees]

- Right ascension at center of image.

dec [real, degrees]

- Declination at center of image.

(extname = SOURCES) [string]

- Name of the source list extension.

(outformat = png) [png—jpeg—fits—tiff—ppm]

- Output graphics format.

(skyserver = skys.gsfc.nasa.gov) [string]

- Name of Skyview server.

(skysurvey = Digitized Sky Survey) [string]

- Name of Skyview survey.

showcat [boolean]

- Indicate catalog matches?

cleanup [boolean]

- Remove temporary files?

clobber [boolean]

- Overwrite existing output files?

(chatter = 1) [integer, 0 - 5]

- Standard HEAdas verbosity parameter.

EXAMPLES

1. Execute uvotpict providing the source list, and center for the image

- uvotpict infile=sources.fits ra=161.41 dec=-1.4

SEE ALSO

uvottfc. uvotstarid.

LAST MODIFIED

April 2004

5.23 UVOTPRODUCT

NAME

uvotproduct - create level III science products from Level II UVOT data.

USAGE

uvotproduct infile=<filename> outfile=<filename> plotfile=<filename> srcreg=<filename> bkgreg=<filename> batpos=<string> xrtpos=<string> uvotpos=<string> groundpos=<string> reportfile=<filename>

DESCRIPTION

This tool creates level III science products from Level II sky image files. It is meant primarily for use in the Swift pipeline, but could also be of some use for initial data inspection by, e.g., Swift Burst Advocates. The batpos, xrtpos, uvotpos, and groundpos parameters all use the same scheme for specifying a position/error circle. They can either be a region file containing a single included fk5 circle, for example fk5;circle(280.32,-23.5,5") or a string of the form ra+dec∼error, for example 280.32-23.5∼5 would specify the same error circle as the preceding region file.

PARAMETERS

infile [filename]

- A comma separated list of files or @filename to load the file names from a file containing one file name per line.

outfile [filename]

- Name for output magnitude history.

plotfile [filename]

- Name for output magnitude history plot or NONE. The extension of plotfile can be used to control the PGPLOT device.

srcreg [filename]

- Source region file.

bkgreg [filename]

- Background region file.

batpos [string]

- A region file name or a string ra+dec∼error giving the BAT position/error circle.

xrtpos [string]

- A region file name or a string ra+dec∼error giving the XRT position/error circle.

uvotpos [string]

- A region file name or a string ra+dec∼error giving the UVOT position/error circle.

groundpos [string]

- A region file name or a string ra+dec~error giving a ground position/error circle.

reportfile [filename]

- Name for output report or NONE.

(plotmag = yes) [boolean]

- If true, plotfile will use magnitudes; otherwise rates.

(fcprefix = NONE) [boolean]

- Finding chart prefix or NONE. If this parameter is not NONE, then two jpg images will be produced starting with fcprefix. One will be an image from a DSS server. The other will be based on the earliest available V or WHITE exposure. The size of these images will depend on what positions are available. If only a BAT position is available, they will be 6' square; otherwise they will be 2' square.

(clobber = no) [boolean]

- Standard HEAdas clobber parameter; controls whether the output files are permitted to overwrite existing files.

(chatter = 1) [enumerated integer]

- Standard HEAdas chatter parameter (1-5) controlling the verbosity of the task.

EXAMPLES

The following examples illustrate running uvotproduct

1. run uvotproduct prompting for all mandatory options:

- uvotproduct

2. run uvotproduct specifying filenames and source positions on command line:

- uvotproduct infile=@img.lis outfile=maghist.fits plotfile=maghist.cps batpos=23.33-41.83~120 xrtpos=23.334-41.839~5.0 uvotpos=23.334743-41.839381~1.0 reportfile=summary.txt chatter=5

SEE ALSO

uvotmag, uvotmaghist

LAST MODIFIED

November 1, 2007

5.24 UVOTRMFGEN

NAME

uvotrmfgen - create a UVOT response matrix

USAGE

uvotrmfgen spectrum=<filename> areafile=<filename> outfile=<filename>

DESCRIPTION

This tool creates a UVOT response matrix given an input spectrum and effective area file. The output response matrix is OGIP compliant and appropriate for use with XSpec.

PARAMETERS

spectrum [filename]

- Wavelength scale extension (of uvotimgrism output).

(areafile=CALDB) [string]

- Effective area calibration file.

(lsffile = CALDB) [string]

- Line Spread Function calibration file.

outfile [filename]

- UVOT redistribution matrix (RMF) file.

(chatter = 1) [enumerated integer]

- Standard HEAdas chatter parameter. A range of 0 to 5 is enforced by the min and max fields in the parameter file.

(clobber = no) [boolean]

- Standard HEAdas clobber parameter.

EXAMPLES

The following example illustrates running uvotrmfgen

1. rrun uvotrmfgen specifying spectrum and output files on the command line

- uvotrmfgen spectrum=uvotimgrism.pha outfile=rmf.fits

SEE ALSO

uvotimgrism

LAST MODIFIED

July 2007

5.25 UVOTSCREEN

NAME

uvotscreen - filter a UVOT event list

USAGE

uvotscreen infile=<filename> attorbfile=<filename> outfile=<filename> aoexpr=<filtering expression>

DESCRIPTION

This tool screens UVOT level 1 event files. It has an optional preprocessing step that sets the QUALITY flags for intrinsically bad pixels. The output file contains the filtered events table and GTI extensions corresponding to each input GTI extension intersected with the GTIs of the filtering expression applied to the orbital/attitude quantities.

PARAMETERS

infile [filename]

- UVOT level 1 events file (see SSC UVOT Data Handbook).

outfile [filename]

- UVOT level 2 events file (see SSC UVOT Data Handbook).

attorbfile [filename]

- Orbit and attitude quantities. This file is produced by prefilter.

badpixfile [filename or NONE]

- Bad pixel list or NONE. If not NONE, used to update the QUALITY flags for hot, cold, dead, flickering bad pixels.

aoexpr [string]

- Orbit/attitude filtering expression. Passed to maketime.

evexpr [string]

- Event filtering expression. Applied to EVENT table.

(cleanup = yes) [boolean]

- Clean up temporary files?

(chatter = 1) [enumerated integer]

- Standard HEAdas chatter parameter. A range of 0 to 5 is enforced by the min and max fields in the parameter file.

(clobber = no) [boolean]

- Standard HEAdas clobber parameter.

(history = yes) [boolean]

- Standard HEAdas history parameter.

EXAMPLES

The following example illustrates running uvotscreen

1. run uvotscreen specifying input and output event files, orbit/attitude file, and filtering expression on command line

- uvotscreen infile=unfiltered.fits attorbfile=prefilter.fits \
  - outfile=filtered.fits aoexpr=SAA_FLAG.eq.0.and.SUN_ANGLE.gt.15

SEE ALSO

prefilter, maketime, extractor

LAST MODIFIED

May 2004

5.26 UVOTSEQUENCE

NAME

uvotsequence - List and visualize UVOT observing sequences

USAGE

uvotsequence imglist=<filename> attfile=<filename> trigtime=<float> plotseq=<boolean> chatter=<enumerate
integer>

DESCRIPTION

Given an ascii list of the full or relative paths and names of a series of UVOT FITS image files, and a suitable attitude file, this tool will determine the times of each Swift snapshot and map each image to to a particular snapshot. Results are provided as standard output. Optionally, the results may be plotted.

PARAMETERS

imglist [filename]

- An ascii list containing paths and names of UVOT FITS image files, e.g.:

/Volumes/data1/00111529000/sw00111529000ubb_sk.img
/Volumes/data1/00111529000/sw00111529000um2_sk.img
/Volumes/data1/00111529000/sw00111529000uuu_sk.img
/Volumes/data1/00111529000/sw00111529000uvv_sk.img
/Volumes/data1/00111529000/sw00111529000uw1_sk.img
/Volumes/data1/00111529000/sw00111529000uw2_sk.img

attfile [filename]

- Name of the corrsponding swift atttitue FITS table.

trigtime [float]

- Time of the BAT trigger in Mission Elapsed Time (MET). This is seconds since 2001 Jan 1, 00:00 UT, and can be obtained directly from BAT products.

(plotseq = yes) [boolean]

- Whether or not to plot results.

(chatter = 1) [enumerated integer]

- Standard HEAdas chatter parameter (1-5) controlling the verbosity of the task.

EXAMPLES

The following examples illustrate running uvotsequence

1. run uvotsequence prompting for all mandatory options:

- uvotsource

2. run uvotsequence specifying all arguments:

- uvotsource imglist=img.lis attfile=auxil/sw00111529000sat.fits trigtime=132853476.48 plotseq=yes chatter=1

SEE ALSO

LAST MODIFIED

April 13, 2005

5.27 UVOTSHIFTPHA

NAME

uvotshiftpha - shift a UVOT PHA in time assuming a power-law decay

USAGE

uvotshiftpha infile=<filename> intime=<float> outfile=<filename> outtime=<float> alpha=<float>

DESCRIPTION

In order to construct spectral energy distributions for use in xspec the .pha files for each UVOT filter must be adjusted to a common time. This routine adjusts the source count rate in a .pha file from "intime" to "outtime". It assumes that the count rate from the source is changing as a power law

- f(t) = f(t0)*(t/t0){}$\hat{}$alpha.

The count rate from the source is shifted to "outime" while the background count rate remains the same. &quotIntime" and "outtime" are specified in arbitrary units, but both times must be in the same units. The times are relative to some arbitrary reference time.

The routine works for .pha files with count rates and .pha files with total counts.

Be aware that this routine assumes a particular form for the light curve (i.e., a power law) and will produce invalid results if the light curve does not behave in this way.

PARAMETERS

infile [filename]

- Input .pha file for some UVOT lenticular filter.

intime [real]

- Time since BAT trigger of input file.

outfile [filename]

- Output .pha file.

outtime [real]

- Time since BAT trigger to shift to.

alpha [real]

- Power-law index alpha for the formula f(t) = f(t0)*(t/t0)$\hat{\{\}}$alpha

(cleanup = yes) [boolean]

- Remove intermediate files?

(history = yes) [boolean]

- Write history keywords?

(clobber = no) [boolean]

- Overwrite existing files?

(chatter = 1) [enumerated integer]

- Standard HEAdas chatter parameter.

EXAMPLES

The following examples illustrate running uvotshiftpha 1. Run uvotshiftpha to shift the PHA from 1234 s after the BAT trigger to 2345 s after the bat trigger assuming a power law light curve with a slope of -1.

- uvotshiftpha infile=v.pha intime=1234 outfile=vshift.pha outtime=2345 alpha=-1.

SEE ALSO

uvot2pha, xspec

LAST MODIFIED

February 2007

5.28 UVOTSKYCORR

NAME

uvotskycorr - attempt to aspect correct UVOT sky images

USAGE

uvotskycorr skyfile=<filename>

DESCRIPTION

This tool iterates over UVOT image files attempting to determine or applying aspect corrections.

PARAMETERS

what [string] (ID—SKY)

- Whether to find corrections (what=ID) or apply corrections (what=SKY).

skyfile [filename]

- Name of input image file(s). This can be a comma-delimited list of file names or @<file> where <file> contains the names of the files to process, one per line.

corrfile [filename]

- Input corrections file for what=SKY.

attfile [filename]

- Input attitude file.

outfile [filename]

- Output file name. For what=ID, the aspect corrections will be written to this file.

(starid = NONE) [string]

- Parameters to pass to star identification.

(catspec = usnob1.spec) [filename]

- Catalog descriptor file(s). A comma-delimited list allows specifying multiple catalog descriptors which will be used left to right.

(cleanup = yes) [boolean]

- Remove intermediate files?

(history = yes) [boolean]

- Write history keywords?

(clobber = no) [boolean]

- Overwrite existing files?

(chatter = 1) [enumerated integer]

- Standard HEAdas chatter parameter.

EXAMPLES

The following examples illustrate running uvotskycorr 1. run uvotskycorr to find corrections

- uvotskycorr what=ID infile=usk.img

SEE ALSO

tristarid aspcorr catspec

LAST MODIFIED

November 2005

5.29 UVOTSOURCE

NAME

uvotsource - Instrumental source magnitude derived from image

USAGE

uvotsource image=<filename> srcreg=<filename> bkgreg=<filename> sigma=<float> zerofile=<filename> coinfile=<filename> psffile=<filename> lssfile=<filename> syserr=<boolean> frametime=<float> apercorr=<string> output=<string> outfile=<filename> cleanup=<boolean> clobber=<boolean> chatter=<integer>

DESCRIPTION

This tool performs aperture photometry on a single source in a UVOT SKY exposure (sw*.img+extension). It returns information about the count rate from the source, the source's magnitude, and flux density information.

The user specifies the source extraction region and background region using region files that are in the standard ftool or ds9 format. Region files must use the fk5 coordinate system. The RA and Dec coordinates can be specified in either decimal degrees or in sexagisemal format.

If "srcreg" is set to NONE then the tool will compute the "sigma"-sigma limiting magnitude for the exposure. The "sigma" parameter tells the tool what level of significance to use to compute the background limit. Photometric and coincidence loss calibration data are read from the files specified by "zerofile" and "coinfile" respectively. If a large-scale sensitivity map is available it can be specified using "lssfile" If "apercorr" is set to CURVEOFGROWTH the PSF data is read from "psffile". The "psffile" is not used if "apercorr=NONE" The special value CALDB tells the tool to obtain calibration data from the Swift/UVOT calibration database. The user must have the $CALDB environment variable correctly set to use this option. If "frametime" is set to DEFAULT the frame time is read from the image header. In some cases this keyword may not be present in the FITS file, so a value can be specified by the user.

There is currently one option for doing photometry. The tool does simple aperture photometry. All of the counts in the "srcreg" region are summed and divided by the exposure time to produce a count rate. The background count rate is subtracted and the magnitude is computed from the coincidence-corrected net count rate. Photometry is done as follows.

Extract the raw counts in three apertures. The user-supplied source region, "srcreg". The source aperture may be any size or shape that can be described in a valid region file. The source region should be selected to maximize the science return. In general faint point sources should use circular apertures with a radius that maximizes the signal-to-noise ratio in the aperture. This is typically about 3 arcsec. For bright sources the standard photometric aperture (defined in "zerofile") is usually preferred. The source aperture should be chosen to minimize contamination from other sources.

The user-supplied background region, "bkgreg". The background region may be any size or shape that can be described in a valid region file. It should be chosen to have the same background properties as the source region. It should be free of contaminating sources and large enough so that the mean pixel value is not biased by Poisson statistics. The background value is computed by taking the mean of the pixel values in the background region.

A coincidence-loss correction aperture, defined in "coinfile" This region is a circular aperture with an radius defined by the COIAPT column in the COINCIDENCE extension of the "coinfile" file. It is centred on the centre of the "srcreg" region. This is the region that is used to compute the coincidence loss correction factor.

Counts are extracted from the input exposure by the XImage "counts" command. See the XImage User's Guide for details on how counts are extracted.

Calculate the coincidence loss correction factor from the count rate in the coincidence-loss correction aperture. See the fhelp for uvotcoincidence for details about coincidence-loss corrections.\

Apply the coincidence-loss factor to the raw count rate in the source aperture.

Scale the background count rate to the area of the coincidence-loss aperture and then apply the coincidence loss factor.

Scale the coincidence-corrected background rate to the area of the user-supplied source aperture and subtract this from the coincidence-corrected rate in the source aperture to get the coincidence-corrected net count rate from the source.

Apply the photometric calibrations from "zerofile" to the coincidence-corrected net count rate from the source to obtain the magnitude and flux density information described below. Note that the photometric calibrations assume that the source region is the same as the standard photometric

aperture, so the values returned will not be on the standard UVOT photometric system if some other aperture region is used. See the description of the CURVEOFGROWTH aperture correction for details on how to correct for this.

If "apercorr=NONE" then no aperture corrections are performed. In order to perform aperture corrections to convert the measured magnitude to the standard UVOT system use "apercorr=CURVEOFGROWTH". This does aperture photometry with one additional step. Uvotsource will only return magnitudes and flux densities that are on the standard UVOT photometric system if the source region is the same as the standard photometric aperture defined in "zerofile". To bring these magnitudes onto the standard system use "apercorr=CURVEOFGROWTH", which computes an aperture correction to the coincidence-loss corrected net count rate from the source. See the fhelp for uvotapercorr for details on aperture corrections and how they are applied. Be aware that "apercorr=CURVEOFGROWTH" assumes that the source is a point source.

The aperture correction applied by CURVEOFGROWTH is approximate and is intended for preliminary data analysis. It does not take into account changes in the PSF due to temperature or count rate variations (see the uvotapercorr fhelp). For high-precision photometry "apercorr=NONE"should be used, and aperture corrections that take these factors into account should be performed by the user.

After the optional aperture correction has been applied there is an optional correction for variations in the large-scale sensitivity of the detector. This is not done if "lssfile=NONE". See the fhelp page for uvotlss for more details.

Uvotsource returns the following information:

Source Information

- The position is the position specified in the "srcreg" region file. The exposure time is the value of the EXPOSURE keyword.

Magnitude Information

- The magnitude of the source, its one-sigma statistical error, and the significance of the detection. Background is the magnitude of the sky. Background-limit is the "sigma"-sigma limiting magnitude of the exposure. Coincidence-limit is the magnitude corresponding to a count rate of one count per frame time.

Flux Information

- The flux density information is given in cgs units of flux per Angstrom unit. Flux densities are computed from flux conversion factors in "zerofile" assuming a mean GRB spectrum. They do not reflect the actual spectrum of the source.

Coincidence Corrected Rate Information

- This is the count rate of the source after all corrections have been applied. This includes aperture corrections if the "apercorr=CURVEOFGROWTH" was used and large-scale sensitivity corrections if a large-scale sensitivity map was specified.

Raw Rate Information

- This is the raw count rate from the source after subtracting the background count rate, but before applying coincidence, aperture, or large-scale sensitivity corrections.

Flux mJy

- The flux density information is given in milliJansky. Flux densities are computed from flux conversion factors in "zerofile" assuming a mean GRB spectrum. They do not reflect the actual spectrum of the source.

The output FITS file contains the following columns.

MET

- The mission elapsed time (MET), in seconds since the reference time, of the observation. The point in the exposure that MET refers to is specified by the TIMEPIXR keyword in the header of "outfile"

EXTNAME

- The name of the FITS extension that this source is in.

TSTART

- The MET start time of the exposure

TSTOP

- The MET stop time of the exposure.

EXPOSURE Please see <http://swift/gsfc.nasa.gov/docs/swift/analysis/uvot_digest.html> for details about how the UVOT exposure time keywords are used. The EXPOSURE column contains the total exposure time with the following corrections applied.

The time that the filter wheel was in BLOCKED is subtracted.

The time lost due to the on-board shift-and-add algorithm is subtracted.

The time lost because the DPU stalled due to high count rates is subtracted.

The dead time is corrected for.

EXPOSURE represents the actual time that the detector was was detecting photons.

TELAPSE

TSTOP - START

SRC_AREA

- The area of the user-supplied source extraction region in square arcseconds.

BKG_AREA

- The area of the user-supplied background region in square arcseconds.

PLATE_SCALE

- The plate scale, in arcsec per pixel, of the image.

RAW_TOT_CNTS

- The total measured counts in the source region.

RAW_TOT_CNTS_ERR

- The one-sigma error in RAW_TOT_CNTS. This error is calculated using binomial statistics since the number of counts that can be measured is limited by the number of frames in the exposure. The binomial error approaches the Poisson error in the limit of small count rates.

RAW_BKG_CNTS

- The total measured counts in the background region.

RAW_BKG_CNTS_ERR

- The one-sigma error in RAW_BKG_CNTS. This error is calculated using binomial statistics since the number of counts that can be measured is limited by the number of frames in the exposure. The binomial error approaches the Poisson error in the limit of small count rates.

RAW_STD_CNTS

- The total measured counts in the coincidence loss region.

RAW_STD_CNTS_ERR

- The one-sigma error in RAW_STD_CNTS. This error is calculated using binomial statistics since the number of counts that can be measured is limited by the number of frames in the exposure. The binomial error approaches the Poisson error in the limit of small count rates.

RAW_TOT_RATE

- The total measured count rate in the source region.

RAW_TOT_RATE_ERR

- The one-sigma error in RAW_TOT_RATE. RAW_TOT_RATE_ERR = RAW_TOT_CNTS_ERR / EXPOSURE.

RAW_BKG_RATE

- The total measured count rate in the background region.

RAW_BKG_RATE_ERR

- The one-sigma error in RAW_BKG_RATE. RAW_BKG_RATE_ERR = RAW_BKG_CNTS_ERR / EXPOSURE.

RAW_STD_RATE

- The total measured count rate in the coincidence loss region.

RAW_STD_RATE_ERR

- The one-sigma error in RAW_STD_RATE. RAW_STD_RATE_ERR = RAW_STD_CNTS_ERR / EXPOSURE.

COI_STD_FACTOR

- The coincidence-loss correction factor for the coincidence-loss region. This value is the multiplicative correction that is applied to raw count rate (after correcting to the STD_AREA) to get the coincidence-corrected count rate.

COI_STD_FACTOR_ERR

- The one-sigma error in COI_STD_FACTOR.

COI_BKG_FACTOR

- The coincidence-loss correction factor for the background region. This value is the multiplicative correction that is applied to the background count rate (after correcting to the STD_AREA) to get the coincidence-corrected background count rate.

COI_BKG_FACTOR_ERR

- The one-sigma error in COI_BKG_FACTOR.

COI_TOT_RATE

- The coincidence-loss correction count rate in the source region.

COI_TOT_RATE_ERR

- The one-sigma error in COI_TOT_RATE. COI_TOT_RATE_ERR = COI_STD_FACTOR * RAW_TOT_RATE_ERR.

COI_BKG_RATE

- The coincidence-loss correction count rate in the background region.

COI_BKG_RATE_ERR

- The one-sigma error in COI_BKG_RATE. COI_BKG_RATE_ERR = COI_STD_FACTOR * RAW_BKG_RATE_ERR.

COI_SRC_RATE

- The coincidence-loss correction count rate from just the source.

COI_SRC_RATE_ERR

- The one-sigma error in COI_SRC_RATE. COI_TOT_RATE_ERR = sqrt(COI_TOT_RATE_ERR**2 - (COI_BKG_RATE_ERR*SRC_AREA)**2).

AP_FACTOR

- The aperture correction factor to be multiplied by COI_SRC_RATE. AP_FACTOR = 1.0 if "apercorr=NONE" was selected.

AP_FACTOR_ERR

- The one-sigma error in AP_FACTOR. AP_FACTOR_ERR = AP_COI_SRC_RATE_ERR / COI_SRC_RATE_ERR.

AP_COI_SRC_RATE

- The source count rate with coincidence-loss corrections and aperture corrections applied.

AP_COI_SRC_RATE_ERR

- The one-sigma error in AP_COI_SRC_RATE. This error is computed as the quadratic sum of COI_SRC_RATE_ERR and the systematic error in the shape of the point-spread function. The systematic error in the shape of the point-spread function is currently assumed to be 5%.

LSS_FACTOR

- The large-scale sensitivity factor to be multiplied by AP_COI_SRC_RATE. LSS_FACTOR = 1.0 if no large-scale sensitivity map was specified.

LSS_RATE

- The source count rate with coincidence-loss corrections, aperture corrections, and large-scale sensitivity corrections applied.

LSS_RATE_ERR

- The one-sigma error in LSS_RATE. This is computed by dividing AP_COI_SRC_RATE_ERR by LSS_FACTOR.

MAG

- The magnitude of the source in the UVOT system.

MAG_ERR

- The one-sigma error in MAG.

MAG_BKG

- The sky magnitude in the UVOT system.

MAG_BKG_ERR

- The one-sigma error in MAG_BKG

MAG_LIM

- The "sigma"-sigma limiting magnitude in the UVOT system.

MAG_LIM_SIG

- "sigma" for MAG_LIM.

MAG_COI_LIM

- The magnitude where the count rate is the inverse of the frame time.

FLUX_AA

- The flux density in erg/s/cm$\hat{}\{\}$2/Angstrom.

FLUX_AA_ERR

- The one-sigma error in FLUX_AA.

FLUX_AA_BKG

- The flux density of the sky in erg/s/cm$\hat{}\{\}$2/Angstrom.

FLUX_AA_BKG_ERR

- The one-sigma error in FLUX_AA_BKG.

FLUX_AA_LIM

- The "sigma"-sigma limiting flux density in erg/s/cm{}2/Angstrom.

FLUX_AA_COI_LIM

- The flux density where the count rate is the inverse of the frame time.

FLUX_HZ

- The flux density in milliJansky.

FLUX_HZ_ERR

- The one-sigma error in FLUX_HZ.

FLUX_HZ_BKG

- The flux density of the sky in milliJansky.

FLUX_HZ_BKG_ERR

- The one-sigma error in FLUX_HZ_BKG.

FLUX_HZ_LIM

- The "sigma"-sigma limiting flux density in milliJansky.

FLUX_HZ_COI_LIM The flux density where the count rate is the inverse of the frame time. The errors in the computed quantities are the 1-sigma statistical errors based on binomial statistics in the count rates. The computed quantities are appended to the FITS file specified by "outfile". If "syserr=YES" then the systematic errors in the calibration are added in quadrature to the statistical errors.

PARAMETERS

image [filename]

- Name of a FITS image (filename + extension).

srcreg [filename]

- Name of an ASCII source region file. Set to NONE to compute a limiting magnitude.

bkgreg [filename]

- Name of an ASCII background region file.

sigma [float]

- Level of detection significance over the mean background.

(zerofile = CALDB) [string]

- Zero points file. The special value CALDB indicates to read from the calibration data base.

(coinfile = CALDB) [string]

- Coincidence loss correction file. The special value CALDB indicates to read from the calibration data base.

(lssfile = CALDB) [string]

- Large-scale sensitivity correction file. The special value CALDB indicates to read from the calibration data base.

(psffile = CALDB) [string]

- Radial encircled energy function (PSF) file. This is only used if CURVEOFGROWTH is specified. The special value CALDB indicates to read from the calibration data base.

(syserr = no) [boolean]

- Are systematic errors in the photometric calibration to be used in the error calculations. If set to YES then systematic errors are added in quadrature to the statistical errors. If set to NO then only statistical errors are returned.

(output = magnitude—flux—rate—raw—fluxj—ALL) [string]

- Output units. The options are instrumental filter magnitude, flux density in erg/s/cm$\hat{}$2/Angstrom, corrected count rate in counts/s, raw count rate in counts/s, flux density in milliJanskys. ALL indicate that all sets of units are output.

(frametime = DEFAULT) [float]

- The frame time, in seconds, for the exposure. If DEFAULT is specified then "frametime" is read from the image header.

(apercorr = NONE) [string]

- Aperture correction calculation algorithm. NONE or CURVEOFGROWTH. CURVEOFGROWN assumes a point source.

(output = magnitude—flux—rate—raw—fluxj—ALL) [string]

- Output units. The options are instrumental filter magnitude (mag), flux density in erg/s/cm$\hat{}$2/Angstrom (flux), corrected count rate in counts/s (rate), raw count rate in counts/s (raw), flux density in milliJanskys (fluxj). The special value ALL indicate that all sets of units are output.

outfile [string]

- Output FITS file to append results to. Uvotsource will create a new file if the specified file does not exist. The special value NONE indicates to not create an output file.

(cleanup = yes) [boolean]

- Standard HEAdas cleanup parameter; controls whether temporary files are removed at the end of processing.

(clobber = no) [boolean]

- Standard HEAdas clobber parameter; controls whether to overwrite pre-existing output files.

(chatter = 1) [enumerated integer]

- Standard HEAdas chatter parameter (1-5) controlling the verbosity of the task.

EXAMPLES

The following examples illustrate running uvotsource

1. run uvotsource prompting for all mandatory options:

- uvotsource

2. run uvotsource specifying all arguments:

- uvotsource image=sky.img.gz+1 srcreg=src.reg bkgreg=bkg.reg sigma=5 zerofile=CALDB coinfile=CALDB psffile=CALDB lssfile=CALDB syserr=NO frametime=DEFAULT apercorr=NONE output=ALL outfile=sources.fits cleanup=YES clobber=NO chatter=1

3. run uvotsource with the curve of growth method and include systematic errors:

- uvotsource image=sky.img.gz+1 srcreg=src.reg bkgreg=bkg.reg sigma=5 syserr=YES apercorr=CURVEOFGROWTH outfile=sources.fits

SEE ALSO

uvotflux, uvotmaghist, uvotapercorr, uvotcoincidence, uvotdetect

LAST MODIFIED

February 19, 2008

5.30 UVOTTFC

NAME

uvottfc - Swift UVOT TDRSS finding chart processing

USAGE

uvottfc infile=<swufc.fits>

DESCRIPTION

uvottfc reads a FITS-ified Swift UVOT TDRSS finding chart packet and generates a source list and a sparse image.

PARAMETERS

infile [filename]

- Name of the input FITS TDRSS finding chart packet.

outfile [filename]

- Name of the output source list file.

sparsefile [filename]

- Name of the output sparse image of NONE.

teldeffile [filename]

- Name of the UVOT telescope definition file or CALDB.

badpixlist [filename]

- Name of the UVOT bad pixel list or CALDB.

(smooth = yes) [boolean]

- Whether or not to smooth the sparse image.

(exposure = DEFAULT) [string]

- Exposure duration estimate [s] or DEFAULT. The user can also override the default scale for rate errors with exposure∼scale. For example, 200∼1 indicates no increase in rate errors (a scale factor of 1) due to exposure uncertainty. The special string DEFAULT indicates to look up the exposure duration in a table- the finding chart exposure duration has changed during the mission.

(chatter = 1) [integer, 0 - 10]

- Controls the amount of informative text written to standard output. chatter values above 5 are probably only useful for testing.

EXAMPLES

1. Execute uvottfc prompting the user for parameter values.

- uvottfc

2. Execute uvottfc overriding the default exposure time.

- uvottfc infile=swufc.fits exposure=300

SEE ALSO

LAST MODIFIED

October 2005

Chapter 6

Glossary

Provided below are list of common acronyms used in this document.

Table 4.1: List of acronyms

Acronym	Big words
BAT	Burst Alert Telescope
CALDB	Calibration Database
FITS	Flexible Image Transport System
GSFC	Goddard Space Flight Center
HEASARC	High Energy Astrophysics Science Archive and Research Center
HEASoft	High Energy Astrophysics Software
HK	Housekeeping
NASA	National Aeronautics and Space Administration
OGIP	Office of Guest Investigator Programs
SDC	Swift Data Center
SSC	Swift Science Center
UVOT	Ultraviolet and Optical Telescope
XRT	X-ray Telescope

Chapter 7

References

ds9: http://hea-www.harvard.edu/RD/ds9
HEAdas: http://heasarc.gsfc.nasa.gov/docs/software/lheasoft/headas/developers_guide
HEASARC: http://heasarc.gsfc.nasa.gov
HEASoft: http://heasarc.gsfc.nasa.gov/docs/software/lheasoft
sextractor: http://terapix.iap.fr/rubrique.php?id_rubrique=91
SDC: http://swift.gsfc.nasa.gov/sdc
SSC: http://swift.gsfc.nasa.gov
XIMAGE: http://heasarc.gsfc.nasa.gov/docs/software/lheasoft/xanadu/ximage/ximage.html
XRONOS: http://heasarc.gsfc.nasa.gov/docs/software/lheasoft/xanadu/xronos/xronos.html
XSPEC: http://heasarc.gsfc.nasa.gov/docs/software/lheasoft/xanadu/xspec/index.html