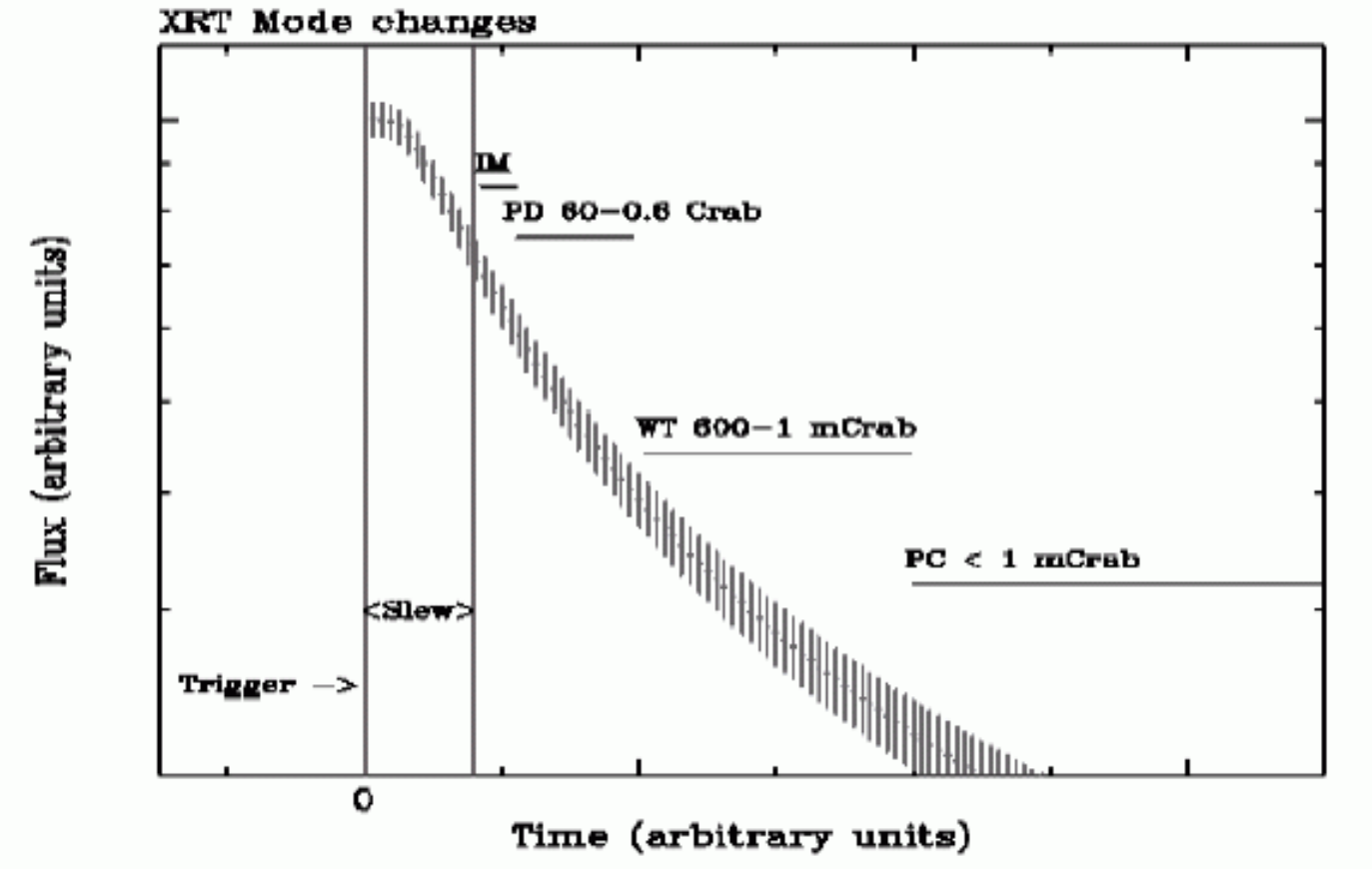
Swift Software Training - XRT

Kim Page

XRT Mode Changing



XRT Modes

- Imaging mode
  - Exposure of 0.1 or 2.5 seconds
  - No spectroscopy data

- Photo-Diode mode (Low-Rate and Piled-Up) – Currently disabled

No spatial information
High time resolution (0.14 ms)

- Windowed Timing mode

1-dimensional imaging
No calibration source included (since they are in the corners)
1.7 ms time resolution

- Photon Counting mode

Calibration sources screened out on ground when full window used
2.5 s time resolution

Important Files for the Pipeline

Under the target directory obtained from the Quick Look site or archive are various subdirectories containing different files, only some of which are important for the XRT:

- auxil - sw[obsid]sat.fits
- xrt
  - event – sw[obsid]x<mode><window><type>_uf.evt
  - hk – sw[obsid]x<hk>.hk

<mode> = pc, wt, lr, pu, im (Photon Counting, Windowed Timing, Low Rate Photo-Diode, Piled Up Photo-Diode, Imaging)

<window> = wn for PD, w1-w5 for WT, w1-w4 for PC (see XRT software guide)

<type> = po, sl, st (Pointing, Slew, Settling)

Event files are *_uf.evt (level 1, unfiltered) or *_ufre.evt (level 1a, reconstructed; WT/PD)

<hk> = hd, tr, en (Header, Trailer, Engineering. The hd file is used in data reduction)

Running the XRT pipeline

xrtpipeline cleanup=no >& log

- cleanup=no keeps all the temporary files the pipeline produces; this is mainly useful because it keeps the extraction region used, which is centred on the position given.
- Including the datamode input (e.g. datamode=pc) on the xrtpipeline command line will just process the mode stated (e.g. pc – Photon Counting)
- exprpcgrade, exprwtgrade and exprpdgrade can be used to specify the grades used (e.g. exprpcgrade=0 for single-pixel PC events). Defaults are 0-12, 0-2 and 0-5 for PC, WT and PD. This can also be done via XSELECT (see later).
- clobber=no is the default.
- createexpomap=yes is now (version 3.7) the default.
- The gti expression includes (ELV>=30||BR_EARTH>=120)&&(SUN_ANGLE>=45 &&ANG_DIST<=0.08&&MOON_ANGLE>=14)&&(CCDTemp>=-102&&CCDTemp<=-47) . To change this, include gtiexpr="expression" on the command line. Default is usually fine, though.

XRT pipeline example

Using Cen A as an example (XRT dataset 00050950004):

```
============================================================
                Running SWIFT XRT pipeline
 Task: xrtpipeline Version: 0.9.3 Release Date: 2005-07-26
============================================================
```

Source RA position (POINT to use optical axis direction) (degrees or hh mm ss.s) [ ] : 13 25 27.6

Source DEC position (POINT to use optical axis direction) (degrees or dd mm ss.s)[ ] : -43 01 09

Target Archive Directory Path [ ] : ../00050950004

Stem for FITS input files [i.e. sw00000000000] [ ] :sw00050950004

Directory for outputs [ ] : ./

Output Files

Running xrtpipeline to completion produces (among others) the following files:

- Attitude/orbit and filter files (sw[obsid]s.attorb, sw[obsid]s.mkf, sw[obsid]smkf.conf)
- Cleaned (level 2) event-list for each mode (sw[obsid]x<mode><window><type>_cl.evt) There will also be a copy of the level 1 event-lists (*_uf.evt)
- Cleaned gti file per event-list (sw[obsid]x<mode><window><type>_clgti.fits)
- Bad (PC and WT) and hot pixel (PC) files (sw[obsid]x<mode><window><type>_ufbp.fits, *_ufhp.fits)
- Image (for PC and IM), spectrum, arf and light-curve (PC, WT and PD) (sw[obsid]x<mode><window><type>sk.img, *sr.pha, *sr.arf, *sr.lc)
- Plots of the image (for PC mode), spectrum and light-curve (PC, WT and PD) (sw[obsid]x<mode><window><type>sk.gif, *lc.gif, *ph.gif)

Hints and Tips

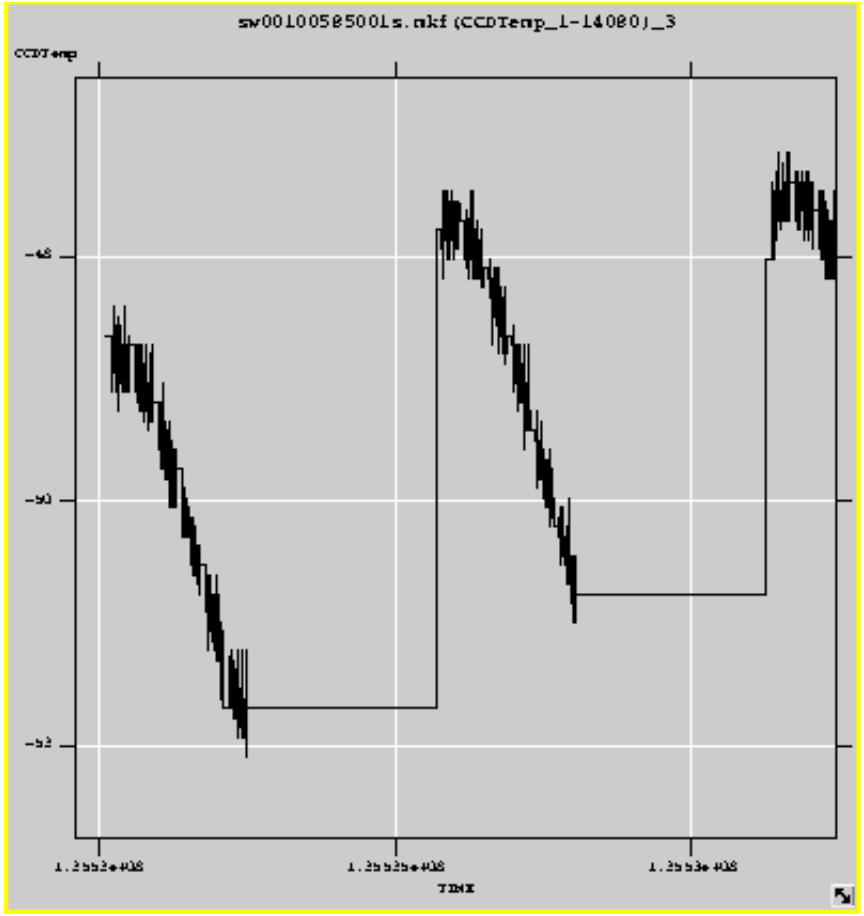
- Sometimes more data can be obtained for an observation by relaxing the observing constraints – specifically the (ELV>=30||BR_EARTH>=120) part. This is more relevant for GRBs than ToOs, though.
- If the count-rate within the field of view is close to the PC/WT boundary, mode-switching can occur. This means that, even if you expect your object to be in PC mode, a substantial portion can end up in WT event files. Time is also lost each time the mode changes. This is annoying, but there's nothing that you, as an observer, can do about it. ☹
- Beware of pile-up! If the count-rate for the source is more than ~0.6-0.7 ct s^{-1}, the PC data are likely to suffer from pile-up (> 100 ct s^{-1} for WT). A piled-up spectrum will be flatter/harder than one which has the pile-up correctly accounted for. A method for estimating the amount of pile-up will be covered later.

mkf File

The *mkf file contains the housekeeping parameters which are used for GTI screening; this includes information about the CCD temperature, the elevation of Swift (angle between the Earth's limb and the pointing direction) and the Bright Earth angle (angle from the illuminated face of the Earth). Temperatures below -47C are generally deemed acceptable.

These can all be easily tabulated and plotted using fv.

XRONOS and XSPEC can be used to analyse the light-curves and spectra.



XSELECT - 1

Although xrtpipeline produces spectral and rate files, they are not background-subtracted and users are recommended to make their own. To extract spectra etc “by hand”, the cleaned event file should be read into XSELECT.

> xselect

Enter session name >[xsel]

> read event

Enter the Event file dir >[ ] ./

Enter Event file list > [ ] sw[obsid]xpcw2po_cl.evt

Got new mission: SWIFT

Reset the mission? >[yes]

Note that some PC eventlists will be w2 or w4 rather than w3. This signifies the size of the window, which has been changed over time. WT data will be of the form sw[obsid]xwtw2po_cl.evt.

XSELECT - 2

> extract image

```
Total      Good       Bad: Region     Time     Phase     Grade     Cut
 6258      6258                0          0         0         0       0
==========================================================================
Grand Total      Good   Bad:  Region  Time  Phase  Grade  Cut
 6258            6258                0        0        0        0     0
   in  3603.8     seconds
 Image          has  6258 counts for  1.737     counts/sec
```

> plot image – opens in ds9. Source and background regions can be defined and saved.

> filter region src.reg – clear region removes this filtering

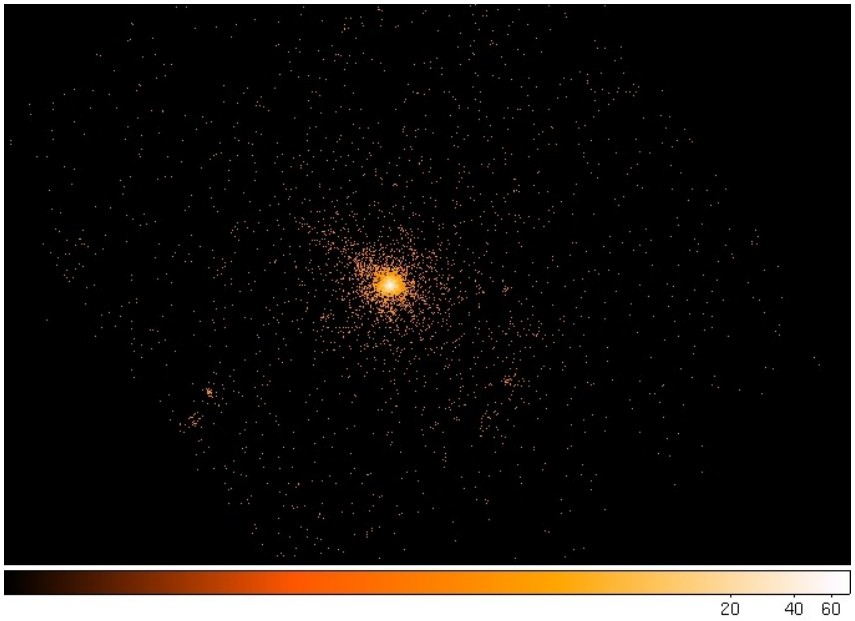
> Set binsize 250 – if you want bins of 250 s; type this BEFORE lc extraction!

> extract curve – light-curve produced for whatever filtering has been defined

> plot curve

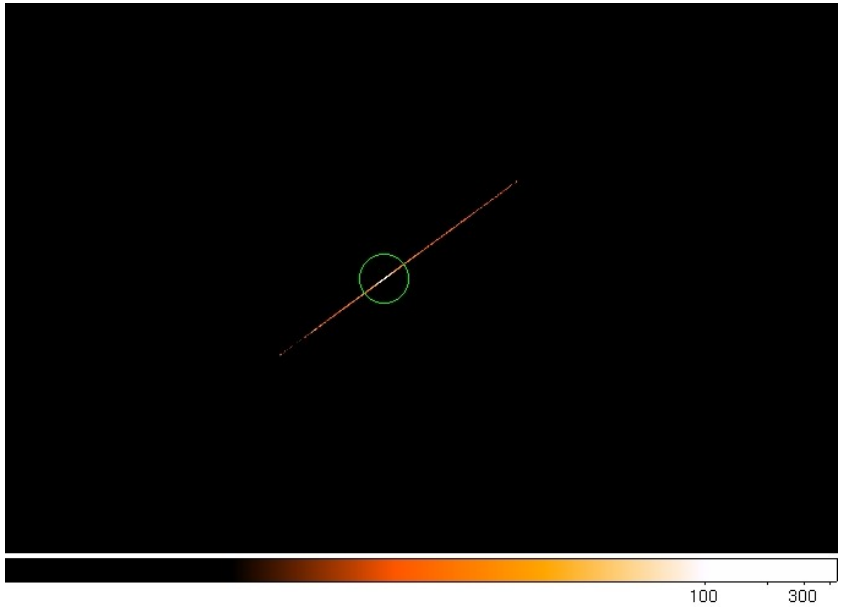
> save curve – N.B. XSELECT does not save anything by default.

XSELECT Images

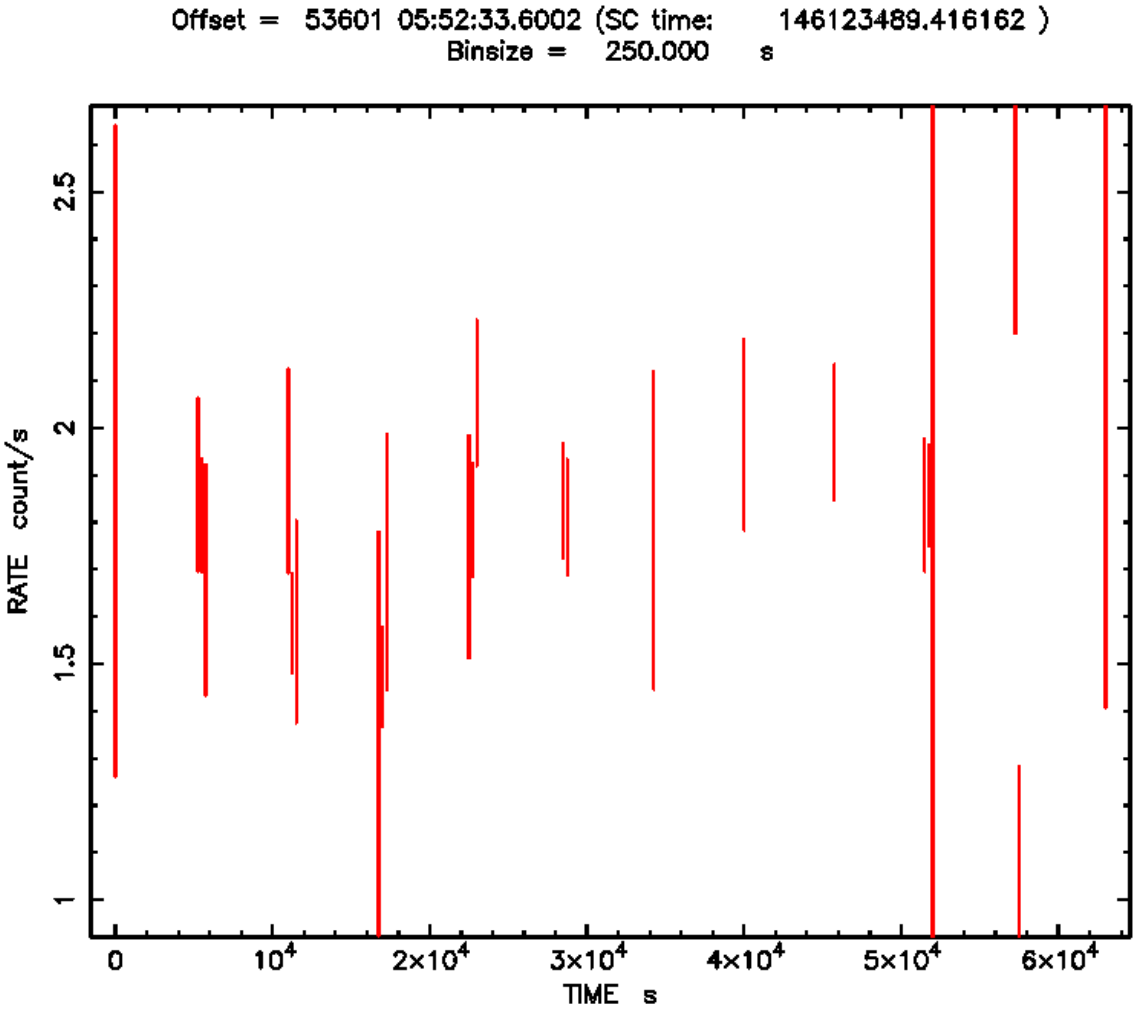


← PC image of Cen A

WT image of Cen A →



XSELECT Light-curve



XSELECT - 3

> filter time cur – allows GTIs to be defined from the light-curve; can also do filter time file <filename> if there is a GTI file already or filter time scc (= spacecraft clock – i.e. time since start of observation).

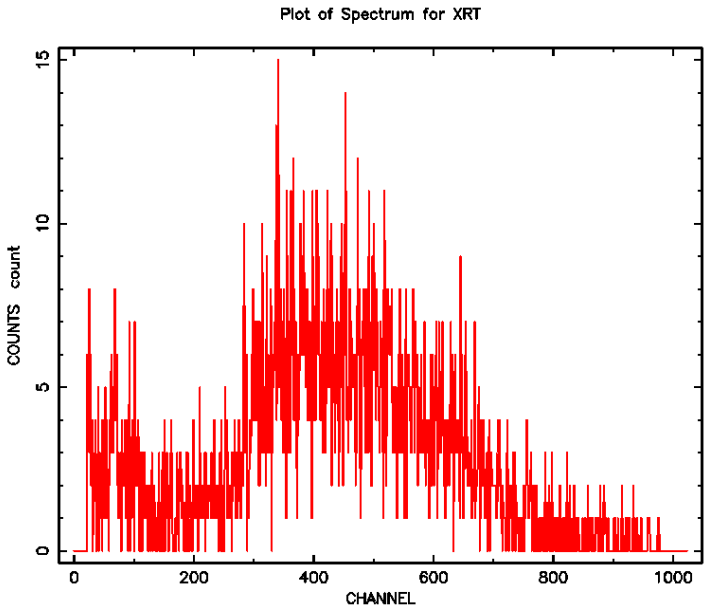
> extract spectrum – uses all previous filtering (grades, regions, times)

> save spectrum – by default this will be a PI file, suitable for XSPEC and for use with the RMFs (providing no filtering with pha_cutoff was performed).

Note: additional GTI screening can also be performed in XSELECT, using the mkf file. For example, select mkf "ELV>5".

By default, PC data are grades 0-12 and WT are 0-2. The modes are also calibrated for grade 0 alone, though (filter grade 0 in XSELECT).

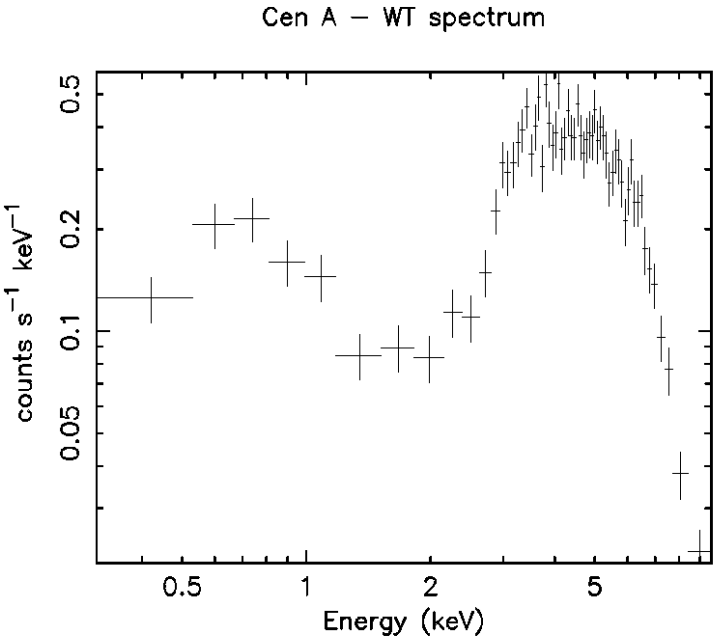
Spectrum



Spectrum within XSELECT

1 channel = 10 eV

"Prettified" within XSPEC.



Pile-up - 1

Although Swift has different modes designed for different count-rates, there can be problems with pile-up (generally in PC mode, but really bright objects can be piled-up in WT), where more than one photon hits a single (or adjacent) pixel before the charge has been read out. Where this is the case, an annular region should be used to extract the spectrum, rather than a circle, in order to exclude the core of the PSF. A quick way to estimate the radius which should be excluded is to run the PSF command in XIMAGE:

> ximage

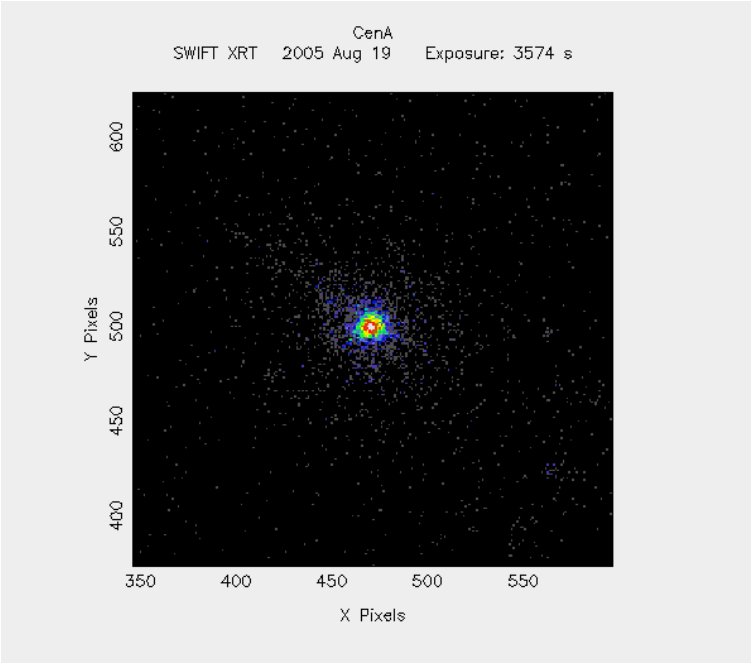
> read sw[obsid]xpcw2posk.img

> cpd /xtk

> disp

> psf/cur

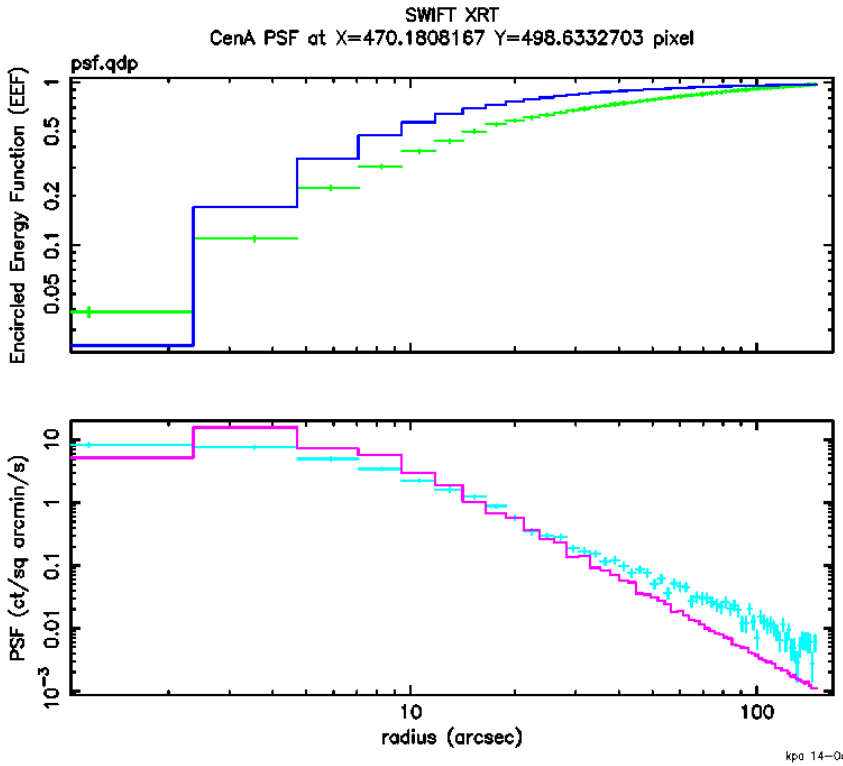
– click in centre and as far out as you want to determine the PSF



Pile-up - 2

This then takes you into QDP, with the PSF profile, which can be fitted with the King function ("model king") out in the wings – away from pile-up. The parameters measured from in-flight calibration are:

$$r_c \sim 5.8;\ \beta \sim 1.55 \quad \text{where } \mathbf{PSF(r) = [1 + (r/r_c)^2\ ]^{-\beta}}$$



Either use the command

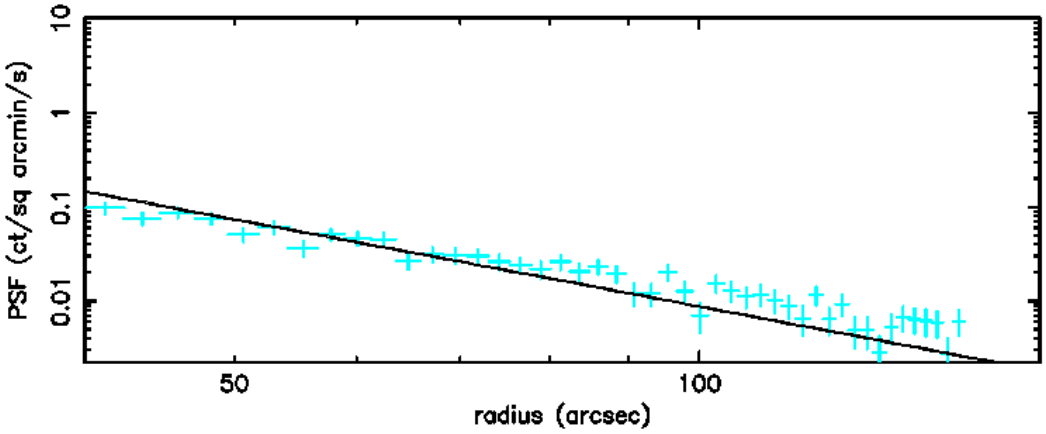
col off 1 2 3 4 6

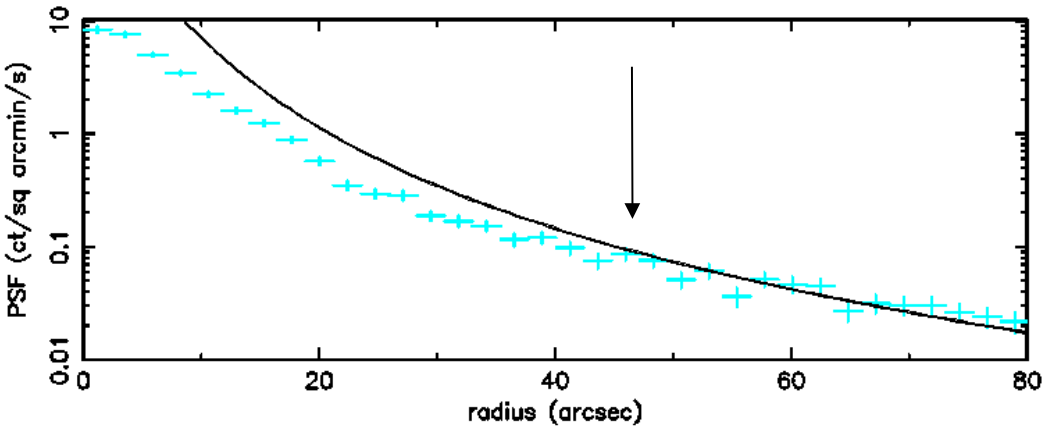
or

fit on 5

The cyan coloured data points in the lower panel are the only ones we're interested in.

Pile-up - 3

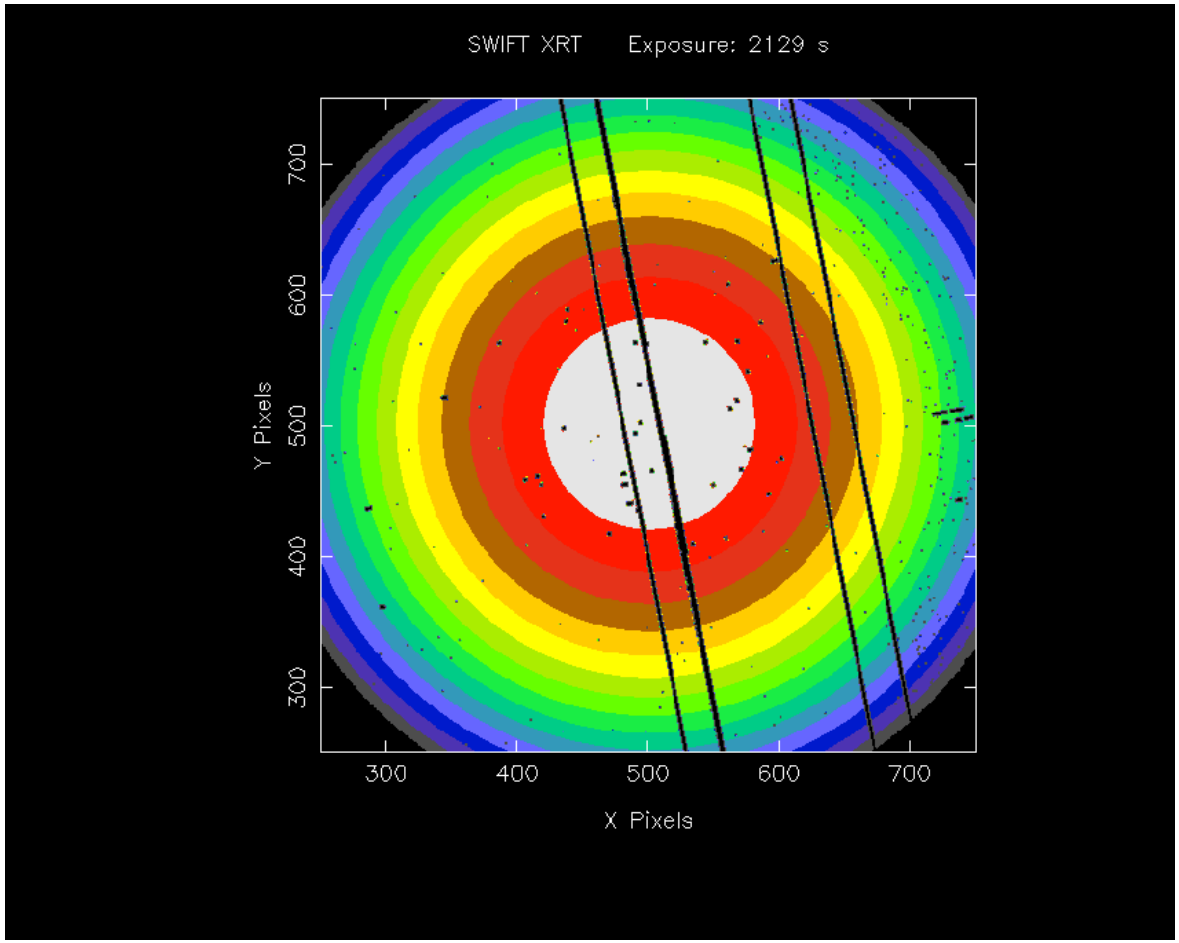




The black line shows the King profile fitted to the wings of the PSF where pile-up is negligible. Extrapolating down to smaller radii shows where the model and data start to diverge; this is the radius which should be excluded when extracting spectra/light-curves.

See also
http://www.swift.ac.uk/pileup.shtml

Exposure Maps - 1



Positioning a source (even partly) over the bad columns leads to a loss of flux. In order to correct for this, an exposure map needs to be generated for each orbit of interest.

This figure shows an example exposure map for PC mode data, demonstrating the bad columns and pixels in the field of view.

Exposure Maps - 2

When fitting a spectrum, an exposure map is required for whichever time interval is being considered, otherwise the calculated flux will be incorrect. The first step is to extract an eventlist for the relevant period of time. If you are using an entire Obs ID, then the default *cl.evt file is fine. Such an exposure map is automatically created by xrtpipeline (for versions 3.7 and up. Otherwise you need to include createexpomap=yes on the command line).

If a shorter interval is required:

Use filter time cursor to pick the orbit(s) of interest within XSELECT. Then a relevant event list needs to be produced:

> extract event copyall=yes

Note that the copyall=yes part is important; it keeps the “badpix” extension, which is needed to run the exposure map command.

If you wish to continue using the full dataset within this XSELECT session, make sure you say NO when prompted to say whether this new event list should be used as the input data file; then type clear event.

Exposure Maps - 3

xrtexpomap is the name of the Swift FTool which generates exposure maps. For each separate event file, this should be run as:

> xrtexpomap attfile=[obsid]/auxil/sw[obsid]sat.fits hdfile=[obsid]/xrt/hk/sw[obsid]xhd.hk infile=orbit1.evt stemout=orbit1 outdir=./

This will make an exposure map called orbit1_ex.img.

Exposure Maps - 4

If you wish to combine more than one exposure map (for example if you want a single spectrum formed from more than one Obs ID – each has separate sat.fits and xhd.hk files, so they need to be considered one by one), they can be added within XIMAGE. Let's say we have a spectrum from datasets 001, 002 and 003. From these event lists, we have produced exposure maps 001_ex.img, 002_ex.img and 003_ex.img. Then:

```
> ximage
> read 001_ex.img
> read 002_ex.img
> sum
> save
> read 003_ex.img
> sum
> save
> write/fits 001-003_ex.img
```

Because the vignetting keyword is not copied by XIMAGE, the user should type (for e.g.)

```
> fparkey F 001-003_ex.img+0 VIGNAPP add=yes
```

Ancillary Response Files - 1

Ancillary Response Files (ARFs) can be built using the tool xrtmkarf. Note that Swift calibration is always on-going!

> xrtmkarf expofile=sw[obsid]xpcw3po_ex.img

Name of the input PHA FITS file [ ] : PC.pi

PSF correction active?(yes/no) [ ] : yes NB. “no” for extended sources.

Name of the output ARF FITS file [ ] : PC_exp.arf

Source X coordinate (SKY for PC and WT modes, DET for PD mode): [ ] : -1

Source Y coordinate (SKY for PC and WT modes, DET for PD mode): [ ] : -1

-1 takes the centre of the extraction region (information stored in the header of the spectrum) as the position of the source.

The RMFs are located in the CALDB, wherever it is installed on your system.

Ancillary Response Files - 2

For both pile-up and exposure map corrections, the ratio between the corrected and uncorrected ARFs needs to be determined to calculate the factor by which light-curve data points should be multiplied, since only the spectra can make use of the corrected ARF directly.
i.e., as well as the example on the previous page, also run:

> xrtmkarf
Name of the input PHA FITS file [ ] : PC.pi
PSF correction active?(yes/no) [ ] : no
Name of the output ARF FITS file [ ] : PC_no.arf

Then, the easiest method for determining the ratio is:

> fstatistic arf specresp – (where arf = name of ARF)

Then compare the “max of selected column” for the ARF with/without the correction. The ratio is the correction factor.

xrtlccorr

There is a useful task called xrtlccorr which calculates the bad column correction factor for every individual orbit and corrects the corresponding light-curves.

> xrtlccorr pcnframe=0 (or wtnframe=0 for WT data)

Name of the input region file or NONE to read region from lcfile: NONE

Name of the input Light Curve FITS file or NONE to read region from regionfile: PC.lc

Name of the Corrected Light Curve : PC_corr.lc

Name of the output file: pc.corr

Name of the input Attitude FITS file : sw[obsid]sat.fits

Name of the output Instrument Map File : DEFAULT

Name of the input Event FITS file : sw[obsid]*cl.evt

Name of the input Housekeeping Header Packets FITS file : sw[obsid]xhd.hk

The pc.corr file has a correction factor for every orbit of data (or every 10s if you don't include pcnframe=0). The input file can be for a single orbit or the whole observation – either will be corrected. The output file will always list the correction factors for every orbit in the cl.evt file, though.

lcmath

The FTool lcmath should be used to perform background subtraction on the light-curve files:

> lcmath

Name of input FITS file [ ] PC_corr.lc

Name of background FITS file [ ] PCback.lc

Name of output FITS file [ ] PC_corr_sub.lc

Scaling factor for input [ ] 1.

Scaling factor for background [ ] 0.25

Add instead of subtract? [ ] no

The background scaling factor is simply the ratio of the area of the source extraction region to that of the background region. So, if the source region has a radius of 30 pixels and the background 60 pixels, the background scaling factor would be $\pi 30^2/\pi 60^2 = 0.25$

flx2xsp - 1

There's a handy little FTool called flx2xsp which takes an ASCII file and converts it to a pha/rsp combination which works in XSPEC, thus allowing all the “spectral” models (power-law, broken power-law etc) to be fitted.

For each light-curve bin, the input file has to be in the format:

tstart tstop CR*(tstop-tstart) CR_Err*(tstop-tstart)

i.e., if you've extracted a light-curve in XSELECT, with 5 s binning, for example, the light-curve file itself would look something like this (NB Should background subtract using lcmath; may also need to correct for pile-up/bad columns…):

```
!tstart CR            CR_Err
0       2.57969999 1.15367699
5       2.06376004 1.03188002
10      1.03188002 0.729649305
15      3.6115799  1.36504889
...
```

This is particularly useful for objects like GRBs where the brightness varies a lot over time, so fixed time bins are not really appropriate.

flx2xsp - 2

If the start time of this observation is 4745.36 s after the trigger time, this needs to be converted into a file like:

4745.36 4750.36 12.9 5.77

4750.36 4755.36 10.32 5.16

4755.36 4760.36 5.16 3.65

If you were to save this in a file called pc1.txt, then you'd run:

> flx2xsp pc1.txt pc1.pha pc1.rsp

pc1.pha can then be read into grppha and XSPEC as though it were a spectrum.

Note that flx2xsp has to be done on each (bright) orbit individually; the software does not cope with the long orbit gaps.

flx2xsp - 3

At later times, you might only want 1 bin per orbit (whatever the number of counts in there, as long as it's a significant detection), or even 1 bin spanning a number of orbits, when it's even fainter.

In this case, you can still use flx2xsp. For example, say we have 2 orbits spanning 1e4 to 3e4 seconds after the trigger, with an on-source exposure time of 10 ks. During this time, there may be 25 background-subtracted source counts. In that case, I would make a 1-line ASCII file:

10000 30000 50 10

where 10000 = tstart and 30000 = tstop

50 = (25/10000)*(30000-10000) ie the back-sub count rate * tstop-tstart

10 = (5/10000)*(30000-10000) where 5 is the error on the count rate.

This, of course, doesn't need to go through grppha after the flx2xsp step, since there's only 1 bin anyway.

Alternatively...

Although this presentation has tried to cover all the basics of how to extract XRT light-curves and spectra, there is a quick and easy way to "cheat"!

Light-curves (and hardness ratios) and time-averaged spectra are routinely created for every XRT-detected GRB and are available online:

http://www.swift.ac.uk/xrt_products

From these pages, there are also links to the Burst Analyser, which combines BAT and XRT data to create flux light-curves in different energy bands.

Similarly, if you want to create a light-curve or spectrum for a non-GRB source, go to

http://www.swift.ac.uk/user_objects

See Evans et al. (2009, MNRAS, 397, 1177), Evans et al. (2007, A&A, 469, 379) and Evans et al. (2010, A&A, 519, A102) for details. These products are completely suitable for scientific analysis and have been used in many papers.

Useful websites

http://www.swift.ac.uk – UK Swift Science Data Centre

http://www.swift.ac.uk/archive/ql.php – Quick Look site (data from past 7 days)

http://www.swift.ac.uk/swift_live – Data Archive

http://www.swift.ac.uk/analysis – Guide to data processing

http://www.swift.ac.uk/analysis/xrt/digest_sci.php – Things to be aware of related to XRT

http://swift.gsfc.nasa.gov/docs/swift/swiftsc.html – NASA Swift website

http://heasarc.gsfc.nasa.gov/docs/heasarc/caldb/swift – Swift calibration files

http://www.swift.ac.uk/xrt_curves - GRB light-curve repository

http://www.swift.ac.uk/user_objects - Build Swift-XRT products for any object

UKSSDC Help Desk: swifthelp@le.ac.uk