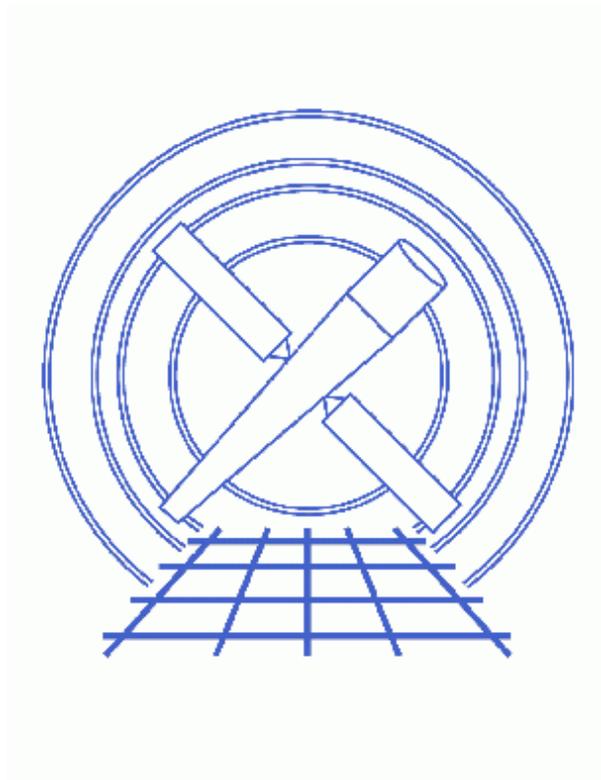


Processing ACA Monitor Window Data



CIAO 3.4 Science Threads

Table of Contents

- ***Background Information***
- ***Get Started***
 - ◆ Obtaining the ACA image data
 - ◆ About the data files
 - ◆ Download monitor photom
- ***Merge the Data Files***
- ***Run the monitor photom script***
- ***Examine the Results***
- ***Advanced Support***
- ***History***
- ***Images***
 - ◆ Dark current histogram
 - ◆ Selecting the ACA image data files in ChaSeR
 - ◆ Plot of the lightcurve in ACA magnitudes (m_ACA)
 - ◆ Count rate and magnitude vs time
 - ◆ Readout window column and row vs time
 - ◆ Quasi-periodic signal in the lightcurve

Processing ACA Monitor Window Data

CIAO 3.4 Science Threads

Overview

Last Update: 1 Dec 2006 – updated for CIAO 3.4: `adat71.fits` filenames updated to match results from *ChaSeR*; ChIPS version

Synopsis:

The Aspect Camera Assembly (ACA) is capable of providing simultaneous optical monitoring of a target during an observation. This thread describes the steps involved in processing the ACA monitor data.

Purpose:

Generate a photometric light curve for a Chandra target which was observed using an ACA monitor window. This is accomplished by use of the `monitor_photom S-lang` script.

Read this thread if:

you are working with any dataset that was processed with [standard data processing](#) (SDP) version DS 7.6.0 or higher; monitor window data were not available in the data products until this version. [Get Started](#) shows how to check the software version used in processing your data.

Related Links:

- [Proposers' Observatory Guide: Pointing Control and Aspect Determination System](#)

Proceed to the [HTML](#) or *hardcopy* (PDF: [A4](#) | [letter](#)) version of the thread.

Background Information

This thread assumes familiarity with the basic ACA operating principles as described in the [Pointing Control and Aspect Determination System](#) chapter of the [Proposers' Observatory Guide \(POG\)](#). This includes the concepts of image slots and readout windows.

The key challenge in deriving a photometric light curve from ACA monitor window data is determining the background dark current. Due to ionizing radiation in the space environment, more than 1/3 of the pixels on the ACA CCD have detectable damage as manifested by an elevated dark current, as shown in [Figure 1](#) .

Processing ACA Monitor Window Data – CIAO 3.4

The main gaussian peak of the dark current distribution, representing undamaged pixels, dominates below about 30 e⁻/sec. Above that value the pixels have been damaged by cosmic radiation. In aspect pipeline processing a pixel with a dark current of more than 200 e⁻/sec is considered "warm" because at this level centroiding can be perturbed. A 15th magnitude source produces only about 500 e⁻/sec, so for faint sources the impact of these warm pixels is significant. To make matters worse, many warm pixels show time dependent flickering behavior in which the pixel dark current can abruptly change by a factor of two or more. This flickering occurs on time scales of thousands of seconds, making it essentially impossible to produce a reliable dark current map using the ACA calibration mode. Instead we use on-the-fly warm pixel detection.

The basic idea of the detection algorithm is to sample pixels from the outer edge of the 8x8 pixel readout window and look for values higher than a threshold. At the edge the contamination of the dark current measurement from the monitor star is minimized. Since the readout window moves with respect to the CCD due to dither, the edge pixels end up sampling much of the CCD region used for imaging. In this way one can build up a map of warm pixels.

There are three limitations to the algorithm:

- Background pixel sampling is not complete nor uniform.
- For brighter stars even the edge pixels have significant contribution from the star light so it is not possible to get a true dark current measurement. The `dark_ratio` tool parameter (see below) specifies that the warm pixel threshold be no less than `dark_ratio * avg_source_counts`.
- The current algorithm in the `monitor_photom` script does not account for flickering, though this could be done with some effort.

Get Started

Sample ObsID used: 4924 (ACIS-S, Mrk 590)

File types needed: evt1; PCAD L1 ACA image data (adat71.fits)

Check the ASCDSVER keyword in the header to see which software version was used in processing the data.

```
unix% dmkeypar acisf04924_001N002_evt1.fits ASCDSVER echo+
7.6.7.2
```

As mentioned in the Overview, only monitor window data processed with an ASCDSVER of DS 7.6.0 or higher should be used in this thread.

Note that monitor window data are only available if the observer has specified target photometry (Photometry=Y) in the observation setup. This can be determined by examining the Obscat parameters, available via [the Target Search Page](#) for a specific ObsID. If you would like to request custom processing of a dataset to obtain the window monitor data, please contact the [Helpdesk](#).

Obtaining the ACA image data

The PCAD Level 1 ACA image data files (`pcad...adat71.fits`) are a "supporting" data product. Currently, supporting products can only be downloaded via [ChaSeR](#); it is not possible to access them from [WebChaSeR](#).

Processing ACA Monitor Window Data – CIAO 3.4

In the *ChaSeR* "Search Results" window, click the "Supporting Products" button to open the selection window. Check the box for the data product "Aspect data", which has a filetype of "acadata" (level 1, detector = "pcad"); the correct selection is shown in [Figure 2](#) . Continue retrieving the data as usual.

About the data files

These files contain the processed ACA telemetry and image data for the observation. The "7" in "adat71" refers to image slot 7, which is the slot where monitor window data always appear. While there may be many more files in the data package (i.e. adat01 – adat61), we will only be using the `adat71.fits` data.

The key data columns in this file are:

Name	Unit	Datatype	Description
time	sec	Real8	Time at start of integration
aca_comp_bkg_avg	count	Real4	Image background level (e ⁻)
img_raw[8,8]	adu	Real4(8x8)	Raw image from ACA
img_corr[8,8]	count	Real4(8x8)	Calibrated image (e ⁻)
fit_resid[8,8]	count	Real4(8x8)	Gaussian fit residuals (e ⁻)
img_row0	pixel	Int2	Row position of lower left pixel
img_col0	pixel	Int2	Column position of lower left pixel
img_excl[8,8]		Byte(8x8)	Exclude pixel flag

Download monitor_photom

The most recent version of `monitor_photom` is 1.0 (23 June 2005):

```
unix% grep Id `which monitor_photom`  
% $Id: monitor_photom,v 1.0 2005/06/23 taldcroft Exp $
```

Please check that you are using the most recent version before continuing. If you do not have the script installed or need to update to a newer version, please refer to the [Scripts page](#).

Merge the Data Files

Typically the image data are split over a number of files, so the first step is to merge the ACA image data into a single file using `dmmmerge`:

```

unix% ls -l *adat71.fits > adat71.lis
unix% cat adat71.lis
pcadf205280136N002_adat71.fits
pcadf205280251N002_adat71.fits
pcadf205309295N002_adat71.fits
pcadf205322477N002_adat71.fits
pcadf205351521N002_adat71.fits
pcadf205361476N002_adat71.fits

unix% dmmerge infile=@adat71.lis outfile=pcad_adat71.fits \
      outBlock=ACADATA

```

The ACA files are input to the tool as a stack; see [ahelp_stack](#) for more information.

Run the monitor_photom script

The main processing steps of the `monitor_photom` script are:

- Read the image data file.
- Median filter image data in time on a pixel-by-pixel basis to remove cosmic rays.
- Search for "warm" pixels that have a dark current well outside the normal distribution. These pixels can significantly affect photometry as they dither in and out of the 8x8 pixel image readout window.
- Subtract the background from each image readout. For warm pixels the detected value for that pixel is used, while for all others the median background dark current reported by the ACA is used.
- Produce lightcurves in counts, counts/sec, and mags.

This script has a parameter file associated with it:

```

unix% plist monitor_photom

Parameters for monitor_photom.par

      infile =                ACA image data file
      outfile =                Output light curve
      (dark_ratio = 0.005)     Dark ratio
      (min_dark_limit = 80.0)  Minimum warm pixel dark current
      (min_dark_meas = 10)     Minimum warm pixel measurements
      (max_dither_motion = 10) Maximum possible dither motion (pixels)
      (mode = ql)

```

The parameters `dark_ratio`, `min_dark_limit`, and `min_dark_meas` affect the way in which warm pixels are detected. The default values in the supplied parameter file are a good starting place, but it is often helpful to adjust the `dark_ratio` and/or `min_dark_limit` to obtain better results.

Now run the script with the merged `pcad_adat71.fits` file as input:

```

unix% monitor_photom infile=pcad_adat71.fits outfile=monitor_lc.fits
Filtering image data (cosmic ray removal)...
  Image 23624 of 23624

Average counts (e-) = 2137.996582
Warm dark limit (e-) = 80.000000

Stacking dark current data...

```

```

Image 23624 of 23624

Warm pixel at CCD (row,col) = -18,10    Dark current (e-) = 85
Warm pixel at CCD (row,col) = -17,11    Dark current (e-) = 325
Warm pixel at CCD (row,col) = -11,11    Dark current (e-) = 615
Warm pixel at CCD (row,col) = -15,12    Dark current (e-) = 105
Warm pixel at CCD (row,col) = -8,16     Dark current (e-) = 115
Warm pixel at CCD (row,col) = -10,17    Dark current (e-) = 105
Warm pixel at CCD (row,col) = -17,19    Dark current (e-) = 170
Warm pixel at CCD (row,col) = -14,20    Dark current (e-) = 130
Warm pixel at CCD (row,col) = -13,20    Dark current (e-) = 175
Warm pixel at CCD (row,col) = -9,20     Dark current (e-) = 110
Warm pixel at CCD (row,col) = -7,20     Dark current (e-) = 350

Enter return to exit the program:

```

The script does cosmic ray removal, reports on the average image counts (in e⁻ for the 1.696 second integration) and the minimum dark current for warm pixels, and then makes a dark current stack to detect warm pixels. Finally, a plot of the lightcurve in ACA magnitudes (m_{ACA}) is created.

The lightcurve is written to the specified FITS file (`monitor_lc.fits`). It contains columns with the time, counts, count rate, magnitude, and background-subtracted image. The magnitude is defined as

$$m_{ACA} = 10.32 - 2.5 * \log_{10}(\text{cnt_rate} / 5263.0)$$

An approximate formula relating B and V magnitude to m_{ACA} is given in [the POG](#):

$$m_{ACA} = V + 0.426 - 1.06*(B-V) + 0.617*(B-V)^2 - 0.307*(B-V)^3$$

This is based on the typical spectral energy distribution of bright main sequence stars.

Examine the Results

A series of *ChIPS* commands are used as a first step in examining the results from the script:

```

unix% chips

Welcome to ChIPS, version CIAO 3.4
Copyright (C) 1999-2003, Smithsonian Astrophysical Observatory

chips> lc = readfile("monitor_lc.fits")
chips> chips.symbolstyle = _chips->none;
chips> chips.curvestyle = _chips->simpleline;
chips> () = curve((lc.time - lc.time[0])/1000 , lc.mag )
chips> ylabel 'Image magnitude (m_{ACA})'

chips> split 1 2
chips> d 2
chips> () = curve((lc.time - lc.time[0])/1000 , lc.cnt_rate )
chips> xlabel 'Time from Obs start (ksec)'
chips> ylabel 'Image count rate (e-/sec)'
chips> print postfile monitor_lc.ps

chips> d 1,2 limit x 20 25
chips> print postfile monitor_lc_zoom.ps

```

The plot results are shown in [Figure 4](#) (`monitor_lc.ps`).

Processing ACA Monitor Window Data – CIAO 3.4

There is an obvious jump around 17 ksec into the observation which is due to a limitation of the ACA flight software. Because the source in a monitor window may not be bright enough for the ACA to independently track, the ACA relies on the motion of another "designated track star" to move the monitor window in lock step. However, if this designated track star is lost by the ACA, even momentarily, then the ACA stops moving the monitor window and leaves it fixed at its last position on the CCD. Instead of the monitor window tracking the motion of the source as Chandra dithers, the source now dithers within the fixed window, possibly going completely outside the readout window. This can be explicitly seen via manipulation of the image data file (`pcad_adat71.fits`):

```
chips> clear

chips> dat = readfile("pcad_adat71.fits")
chips> () = curve((dat.time - dat.time[0])/1000 , dat.img_row0 )
chips> ylabel 'Readout window row'

chips> split 1 2
chips> d 2
chips> () = curve((dat.time - dat.time[0])/1000 , dat.img_col0 )
chips> xlabel 'Time from Obs start (ksec)'
chips> ylabel 'Readout window column'

chips> d 1,2 limit x 12 22
chips> print postfile img_row_col.ps
chips> exit
```

These commands result in [Figure 5](#) .

The impact to the photometry due to not tracking the source can be seen in [Figure 6](#)  (`monitor_lc_zoom.ps`). A quasi-periodic signal in the magnitude of the lightcurve is clearly visible.

Advanced Support

If the monitor photometry data indicate scientifically interesting results worthy of detailed study, users may wish to contact aspect_help@cfa.harvard.edu for further advice on possible techniques to reduce the systematic errors.

History

- 12 Jul 2005 original version, new for CIAO 3.2
- 14 Dec 2005 updated for CIAO 3.3: `adat71.fits` filenames updated to match results from *ChaSeR*
- 01 Dec 2006 updated for CIAO 3.4: `adat71.fits` filenames updated to match results from *ChaSeR*; ChIPS version

URL: http://cxc.harvard.edu/ciao/threads/monitor_photom/

Last modified: 1 Dec 2006

Image 1: Dark current histogram

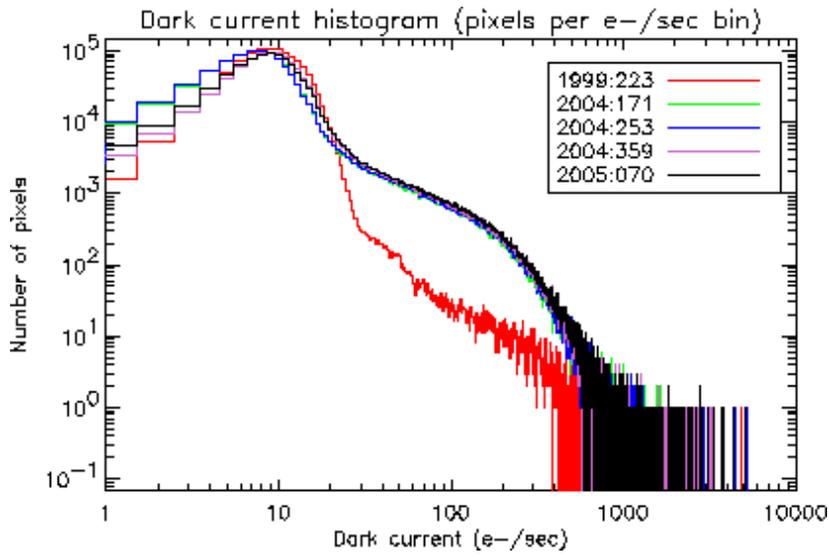


Image 2: Selecting the ACA image data files in ChaSeR

The screenshot shows the 'Chandra Search Results' window. At the top, a table displays search results for observation 701005. Below this, a 'Select supporting products' dialog box is open, allowing the user to choose which data products to include. The dialog contains a table with columns: SELECTED, LEVEL, DETE..., DATA PRODUCT, and FILETYPE. The 'Aspect data' product is checked in the 'SELECTED' column. Below the table are buttons for 'Select all', 'Deselect all', and 'OK'. At the bottom of the dialog, a 'Supporting' button and a status indicator 'Some products selected' are visible. The main window also shows a 'Total rows = 1' indicator.

SEQ_NUM	OBSID	INSTR	GRAT	APP_E...	EXP_TI...	TARGET_NAME	PI_NAME	OBSERVER
701005	4924	ACIS-S	HETG	100	96.79	Mrk 590	Santos-Lleo	02

SELECTED	LEVEL	DETE...	DATA PRODUCT	FILETYPE
<input type="checkbox"/>	0.5	sim	SEA merged tlm	sim_mrg
<input type="checkbox"/>	0.5	sim	SIM coords	simcoor
<input type="checkbox"/>	1	acis	Summary	acis_summary
<input type="checkbox"/>	1	ephem	OBC Viewing an...	oangleephem
<input type="checkbox"/>	1	ephin	EPHIN clean HKP	ephhk_cln
<input type="checkbox"/>	1	ephin	EPHIN rates	ephrates
<input type="checkbox"/>	1	pcad	Asp. centroids	acacent
<input type="checkbox"/>	1	pcad	Aspect cal	acacal
<input checked="" type="checkbox"/>	1	pcad	Aspect data	acadata
<input type="checkbox"/>	1	pcad	Bad Pix	aca_badpix
<input type="checkbox"/>	1	pcad	FID properties	fidprops
<input type="checkbox"/>	1	pcad	GS properties	gsprops
<input type="checkbox"/>	1	pcad	Gyro cal	gyrocal
<input type="checkbox"/>	1	pcad	Gyro data	gyrodata
<input type="checkbox"/>	1	pcad	Kalman	kalman
<input type="checkbox"/>	1	tel	HRMA temps	hrmatrnp
<input type="checkbox"/>	1.5	acis/hrc	TG Events	tgevt1
<input type="checkbox"/>	1.5	acis/hrc	TG Source list	tgsrc

Image 3: Plot of the lightcurve in ACA magnitudes (m_{ACA})

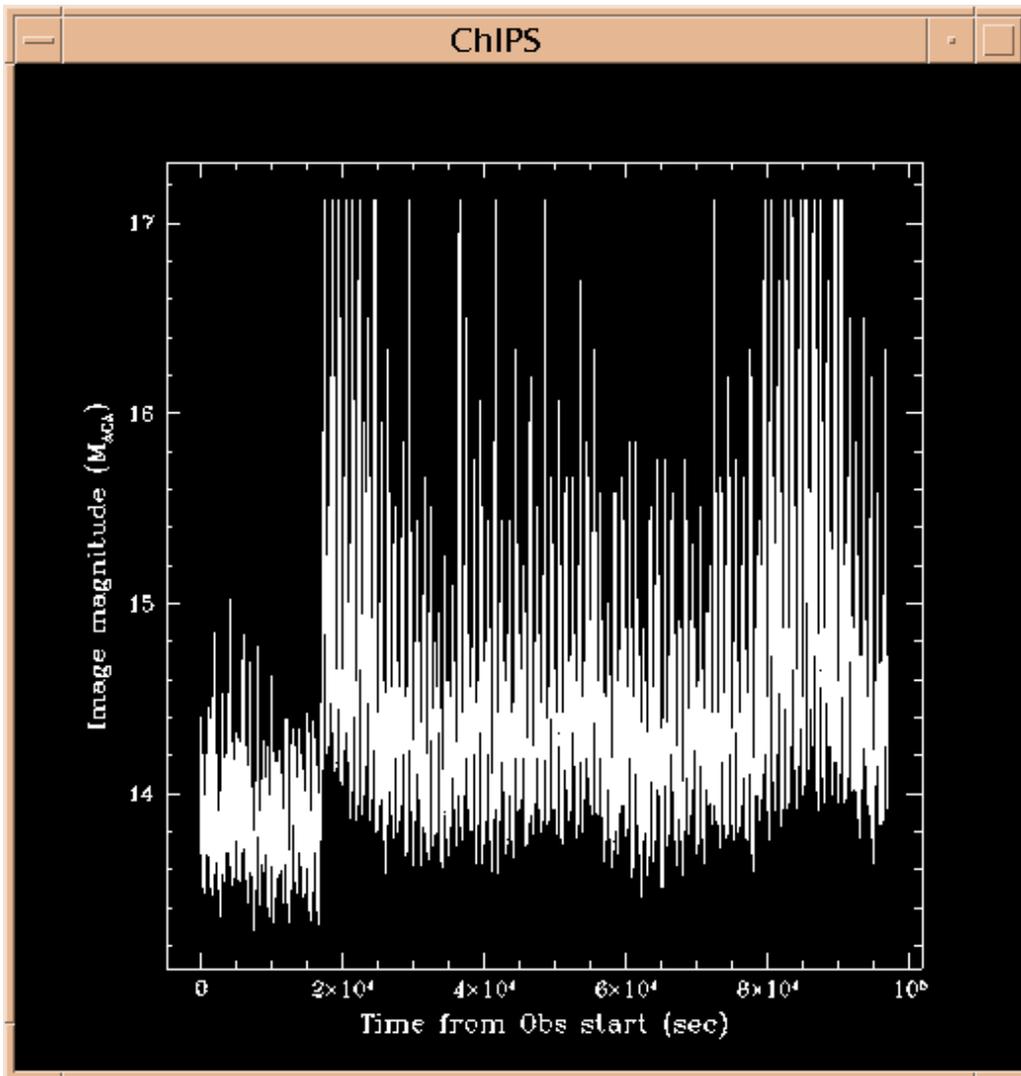


Image 4: Count rate and magnitude vs time

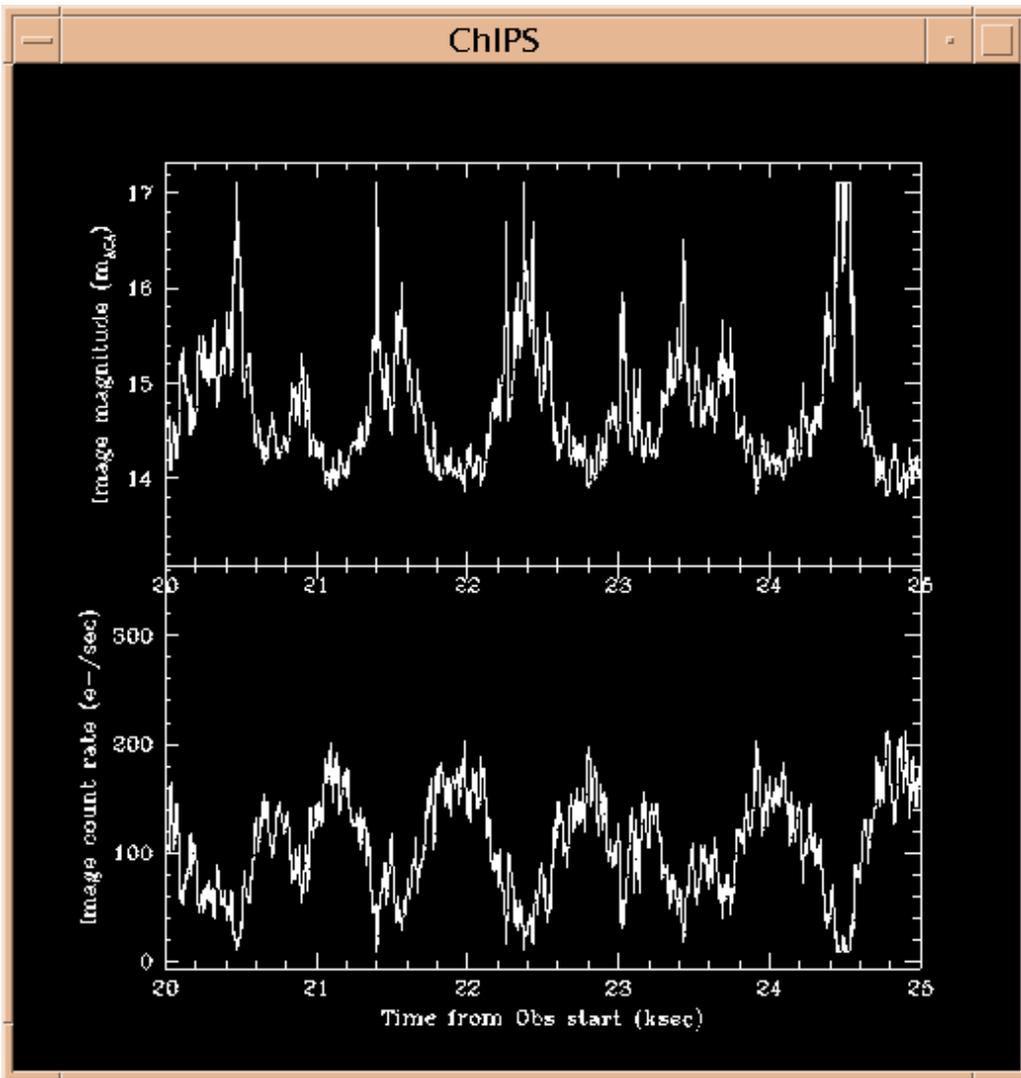


Image 5: Readout window column and row vs time

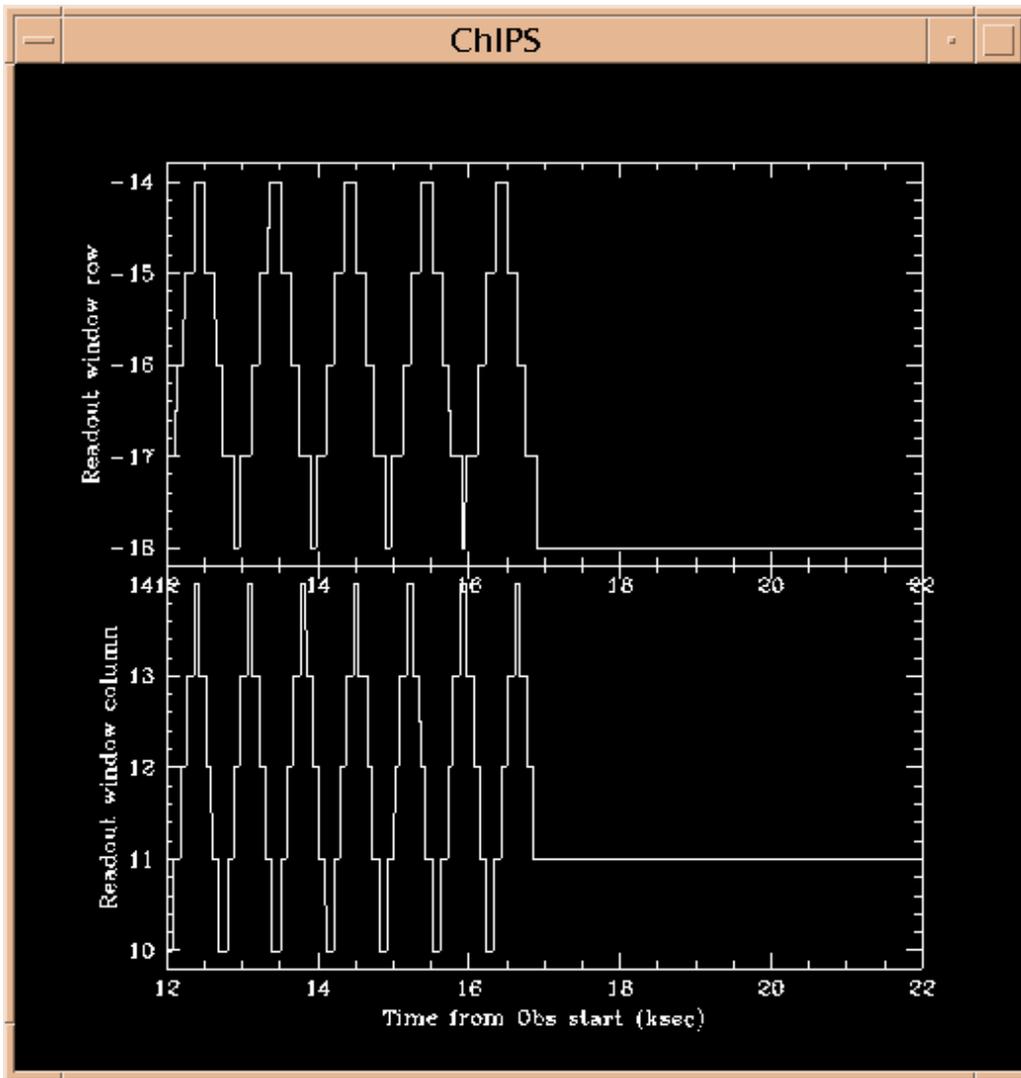


Image 6: Quasi-periodic signal in the lightcurve

