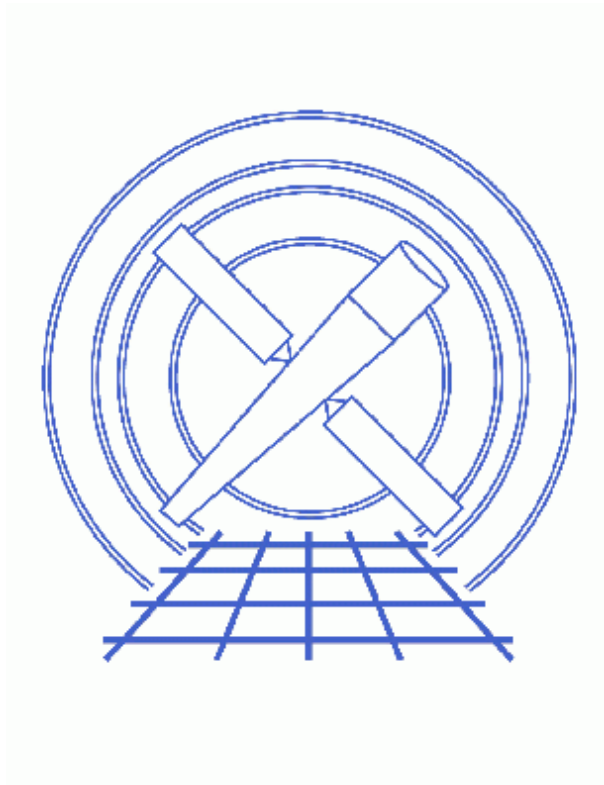


## Match the Binning of an Image



***CIAO 3.4 Science Threads***

# Table of Contents

- *Get Started*
- *Create an image*
- *What region does the image cover?*
  - A. *dmclist opt=cols*
  - B. *dmclist opt=subspace*
  - C. *dmcoords*
- *Run the script (get\_sky\_limits)*
- *Caveats*
- *Parameter files:*
  - ◆ *dmcoords*
  - ◆ *dmcoords*
- *History*
- *Images*
  - ◆ *Offset due to different binning filters*

---

# Match the Binning of an Image

*CIAO 3.4 Science Threads*

## Overview

*Last Update:* 1 Dec 2006 – reviewed for CIAO 3.4: no changes

### *Synopsis:*

There are times that you would like to create an image so that it matches an already-existing image. One example would be creating a full-resolution exposure map of a single detector chip.

### *Purpose:*

To use the `get_sky_limits` script to find the binning specification of an image.

### *Read this thread if:*

you want to create an image from a ACIS or HRC event file (or have an existing image) and to determine the binning specification for input into other CIAO tools, e.g. `dmcopyp` or `mkexpmap`, to match that image.

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

---

## Get Started

*Sample ObsID used:* 1838 (ACIS-S, G21.5-09)

*File types needed:* evt2

This thread uses the `get_sky_limits` script; for information about the script, consult the help file ("[ahelp get\\_sky\\_limits](#)"). The most recent version of `get_sky_limits` is v1.6 (02 November 2004):

```
unix% grep Id `which get_sky_limits`  
% $Id: get_sky_limits,v 1.6 2004/11/02 16:22:30 dburke 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](#).

---

## Create an image

As an example, we select a region of the observation that we are interested in:

```
unix% dmcoppy \  
      "acisf01838N001_evt2.fits[sky=box(4182,4392,1172,1184)][bin x>:::4,y>:::4]" img.fits
```

## What region does the image cover?

The region represented by the image can be found in one of several ways:

### A. dmlist opt=cols

The `cols` option of `dmlist` produces a report on the various coordinate systems stored for the image. In this case we find:

```
unix% dmlist img.fits cols
```

---

```
Columns for Image Block EVENTS_IMAGE
```

---

ColNo	Name	Unit	Type	Range
1	EVENTS_IMAGE[294,297]		Int2(294x297)	-

---

```
Physical Axis Transforms for Image Block EVENTS_IMAGE
```

---

```
Group# Axis#
```

1	1,2	sky(x) = (+3595.50) +(+4.0)* ((#1)-(+0.50))
		(y) (+3799.50) (+4.0) ((#2) (+0.50))

---

```
World Coordinate Axis Transforms for Image Block EVENTS_IMAGE
```

---

```
Group# Axis#
```

1	1,2	EQPOS(RA ) = (+278.3860) +TAN[(-0.000136667)* (sky(x)-(+4096.50))]
		(DEC) (-10.5899) (+0.000136667) ( (y) (+4096.50))

### B. dmlist opt=subspace

The `subspace` option of `dmlist` describes the filters that have been applied to the data:

```
unix% dmlist img.fits subspace | more
```

```
--- Component 1 ---
```

1 sky	Real8	Box(4182,4392,1172,1184)
1 sky	Real8	Field area = 1.39709e+06 Region area = 1.38765e+06
1 sky	[ 1] x	3596.0: 4768.0
1 sky	[ 2] y	3800.0: 4984.0

## Match the Binning of an Image – CIAO 3.4

2	time	Real8	TABLE GTI7	
				84245785.9546994567: 84253741.1547068655
3	ccd_id	Int2	7:7	
4	node_id	Int2	0:3	
5	expno	Int4	0:2147483647	
6	chipx	Int2	1:1024	
7	chipy	Int2	1:1024	
8	tdetx	Int2	1:8192	
9	tdety	Int2	1:8192	
10	detx	Real4	0.50:	8192.50
11	dety	Real4	0.50:	8192.50
12	pha	Int4	0:36855	
13	energy	Real4		0: 1000000.0
14	pi	Int4	1:1024	
15	fltgrade	Int2	0:255	
16	grade	Int2	0:0,2:2,3:3,4:4,6:6	
17	phas	Int2	-4096:4095	
... (other components listed) ...				

### C. dmcoords

The `dmcoords` tool can be used to calculate the edges of the image in a variety of coordinate systems. Here we use it in the non-interactive mode to find the bottom-left and top-right corners:

```
unix% punlearn dmcoords
unix% dmcoords img.fits opt=logical logicalx=0.5 logically=0.5
unix% echo "x = `pget dmcoords x` y = `pget dmcoords y`"
x = 3595.5 y = 3799.5
unix% punlearn dmcoords
unix% dmcoords img.fits opt=logical logicalx=294.5 logically=395.5
unix% echo "x = `pget dmcoords x` y = `pget dmcoords y`"
x = 4771.5 y = 5379.5
```

The parameter files after running these two commands can be found below: [bottom left](#) and [top right](#).

### Run the script (get\_sky\_limits)

The `get_sky_limits` script returns the text you would use in both a DM-binning and `mkexpmap xygrid` specification:

```
unix% get_sky_limits img.fits verbose=1
Checking binning of image: img.fits
Image has 294 x 297 pixels
Lower left (0.5,0.5) corner is x,y= 3595.5, 3799.5
Upper right (294.5,297.5) corner is x,y= 4771.5, 4987.5
DM filter is:
x=3595.5:4771.5:#294,y=3799.5:4987.5:#297
mkexpmap xygrid value is:
3595.5:4771.5:#294,3799.5:4987.5:#297
```

The binning specifications are also stored in the parameter file in the `dmfilter` and `xygrid` parameters:

## Match the Binning of an Image – CIAO 3.4

```
unix% pget get_sky_limits dmfilter
x=3595.5:4771.5:#294,y=3799.5:4987.5:#297
unix% pget get_sky_limits xygrid
3595.5:4771.5:#294,3799.5:4987.5:#297
```

As discussed in the [Caveats](#) section, you may get better results if you use the X, Y range from the data subspace, rather than using dmcoords.

---


## Caveats

As discussed in the [dmbinning](#) ahelp page, care must be taken when binning real-values columns (such as the SKY column of an event file), otherwise the edges of the files may not match up exactly. This can be seen in the above example if you use the dmfilter expression to create a copy of the image and subtract it from the original. In this case, 4 pixels at the bottom edge of the frame are different.

```
unix% dmcoppy \
  "acisf01838N001_evt2.fits[bin x=3595.5:4771.5:#294,y=3799.5:4987.5:#297]" img2.fits
unix% dmimgcalc img.fits img2.fits diff.fits sub
unix% dmstat diff.fits centroid=no
diff.fits
  min:      -1          @:      ( 3833.5 3801.5 )
  max:       0          @:      ( 3597.5 3801.5 )
  mean:     -4.5809569619e-05
  sigma:    0.006768121682
  sum:      -4
  good:     87318
  null:     0
```

The following example shows a larger difference:

```
unix% dmcoppy "acisf01838N001_evt2.fits[bin x=3570:4874:16,y=3650:4980:16]" zoom16_1.fits
unix% dmcoppy "acisf01838N001_evt2.fits[x=3570:4874,y=3650:4980][bin x:::16,y:::16]" zoom16_2.fits
unix% dmimgcalc zoom16_1.fits zoom16_2.fits zoom_diff.fits sub
unix% dmstat zoom_diff.fits centroid=no
zoom_diff.fits
  min:      -27          @:      ( 4090 4266 )
  max:       40          @:      ( 4042 4234 )
  mean:     0
  sigma:    1.3381518623
  sum:      0
  good:     6888
  null:     0
```

The resulting image (zoom\_diff.fits) is shown in [Figure 1](#) , and the [subspace](#) option of dmlist reports the X, Y ranges of the files to be:

```
zoom16_1.fits: x = 3570.0:4882.0  y = 3650.0:4994.0
zoom16_2.fits: x = 3570.0:4874.0  y = 3650.0:4980.0
```

---

Parameters for /home/username/cxcds\_param/dmcoords.par

infile = img.fits

Input dataset/block specification

## Match the Binning of an Image – CIAO 3.4

```
#
# Position of photon in different coord systems
#
    chip_id = 6           Chip ID number
    chipx = 992.2008917225646 Chip X [pixel]
    chipy = -117.7120835385116 Chip Y [pixel]
    tdetx = 0            TDETX [pixel]
    tdety = 0            TDETY [pixel]
    detx = 3827.862706467882 FPC X [pixel]
    dety = 4613.263006148494 FPC Y [pixel]
    x = 3595.5           Sky X [pixel]
    y = 3799.5           Sky Y [pixel]
    logicalx = 0.5       X coordinate in binned image [pixel]
    logically = 0.5      Y coordinate in binned image [pixel]
    ra = 18:33:49.370    RA [deg or hh:mm:ss]
    dec = -10:37:49.78   Dec [deg or dd:mm:ss]
    theta = 4.77581941165613 Off axis angle [arcmin]
    phi = 117.4674954453872 Azimuthal angle [deg]
    order = 0            Grating order
    energy = 1           Energy [keV]
    wavelength = 0       Wavelength [A]
    ra_zo = 18:33:49.370 RA of zero order
    dec_zo = -10:37:49.78 Dec of zero order
    (asolfile = none)    Input aspect solution file
    (option = )          Conversion option
#
# Override setup for observation
# All parameters here are strings so that they can
# be set blank, in which case the data file value is used
#
    (celfmt = hms)       RA and Dec format [deg or hms] (xx.xx or xx:xx:xx.x)
    (detector = )        Detector (ACIS or HRC-I or HRC-S)
    (grating = )         Grating
    (fpsys = )           FP convention
    (sim = )             SIM position (eg 0.0 0.0 -190.6)
    (displace = )        STF displacement (X,Y,Z,AX,AY,AZ)
    (ra_nom = )          Nominal pointing RA [deg or hh:mm:ss]
    (dec_nom = )          Nominal dec [deg or dd:mm:ss]
    (roll_nom = )        Nominal roll [deg]
    (ra_asp = )          Instantaneous pointing RA [deg]
    (dec_asp = )         Instantaneous pointing Dec [deg]
    (roll_asp = )        Instantaneous Aspect roll [deg]
#
    (geompar = geom)     Parameter file for Pixlib Geometry files
    (verbose = 0)        Debug Level
    (mode = ql)
```

---

Parameters for /home/username/cxcds\_param/dmcoords.par

```
    infile = img.fits     Input dataset/block specification
#
# Position of photon in different coord systems
#
    chip_id = 8           Chip ID number
    chipx = 420.8040825766388 Chip X [pixel]
    chipy = 1145.394151296928 Chip Y [pixel]
    tdetx = 0            TDETX [pixel]
    tdety = 0            TDETY [pixel]
```

## Match the Binning of an Image – CIAO 3.4

```
detx = 5339.916579114716 FPC X [pixel]
dety = 3351.094599709191 FPC Y [pixel]
  x = 4771.5           Sky X [pixel]
  y = 5379.5           Sky Y [pixel]
logicalx = 294.5       X coordinate in binned image [pixel]
logicaly = 395.5       Y coordinate in binned image [pixel]
  ra = 18:33:10.140    RA [deg or hh:mm:ss]
  dec = -10:24:52.40   Dec [deg or dd:mm:ss]
theta = 11.88772992568938 Off axis angle [arcmin]
  phi = 329.0580869148563 Azimuthal angle [deg]
  order = 0            Grating order
energy = 1             Energy [keV]
wavelength = 0         Wavelength [A]
  ra_zo = 18:33:10.140 RA of zero order
  dec_zo = -10:24:52.40 Dec of zero order
(asolfile = )         Input aspect solution file
(option = )           Conversion option

#
# Override setup for observation
# All parameters here are strings so that they can
# be set blank, in which case the data file value is used
#
  (celfmt = hms)       RA and Dec format [deg or hms] (xx.xx or xx:xx:xx.x)
(detector = )         Detector (ACIS or HRC-I or HRC-S)
(grating = )         Grating
(fpsys = )           FP convention
(sim = )             SIM position (eg 0.0 0.0 -190.6)
(displace = )        STF displacement (X,Y,Z,AX,AY,AZ)
(ra_nom = )          Nominal pointing RA [deg or hh:mm:ss]
(dec_nom = )          Nominal dec [deg or dd:mm:ss]
(roll_nom = )        Nominal roll [deg]
(ra_asp = )          Instantaneous pointing RA [deg]
(dec_asp = )          Instantaneous pointing Dec [deg]
(roll_asp = )        Instantaneous Aspect roll [deg]

#
(geompar = geom)     Parameter file for Pixlib Geometry files
(verbose = 0)         Debug Level
(mode = ql)
```

---

## History

04 Jan 2005 updated for CIAO 3.2: minor change to dmcoords parameter file

16 Dec 2005 reviewed for CIAO 3.3: no changes

01 Dec 2006 reviewed for CIAO 3.4: no changes

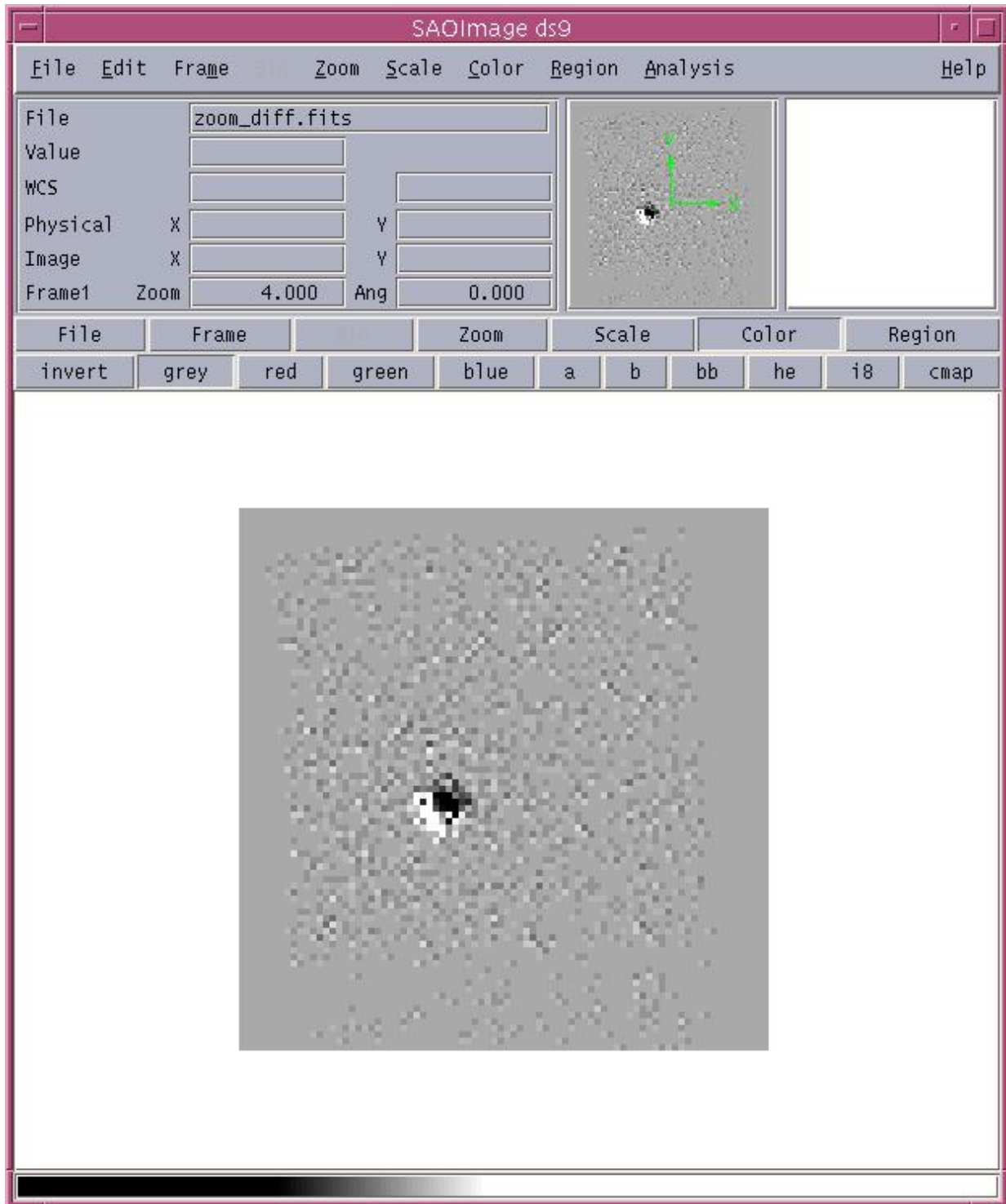
---

URL: [http://cxc.harvard.edu/ciao/threads/match\\_binning/](http://cxc.harvard.edu/ciao/threads/match_binning/)

Last modified: 1 Dec 2006



### Image 1: Offset due to different binning filters



## Match the Binning of an Image – CIAO 3.4