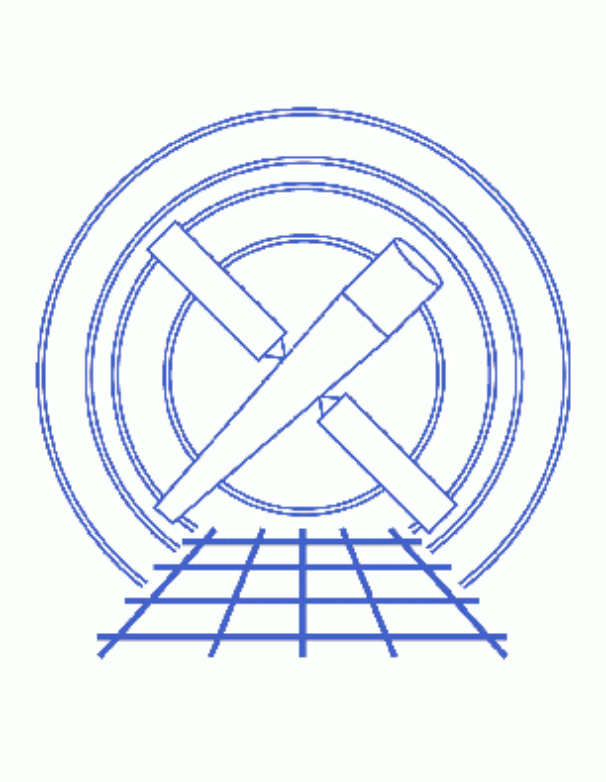
Create A True Color Image



CIAO 3.4 Science Threads

Table of Contents

- ***Get Started***
- ***Create Input Images***
- ***Create a True Color Image***
- ***Real vs. Virtual Image Files***
- ***Using the color_image Script***
- ***Overlaying a Source List***
    A. On an X−ray Image
    B. On an Optical Image
- ***Parameter files:***
    - dmimg2jpg
- ***History***
- ***Images***
    - X−ray true−color image of G21.5−09
    - True−color image with source list overlaid

Create A True Color Image

CIAO 3.4 Science Threads

Overview

Last Update: 8 Mar 2007 − added link to Create A True Color Image in ds9 thread

Synopsis:

Most astronomical images map color to intensity level, i.e. for a greyscale image, lighter tones correspond to a brighter intensity level or vice versa. An alternative way of presenting data is via an image that correllates color and energy or wavelength.

Purpose:

Use the CIAO tool `dmimg2jpg` to create a true color image and (optionally) overlay a source list on that image.

Read this thread if:

you would like to apply a specific color table to a given image in order to create a true color JPEG image. The output may also be a postscript file, which is extremely useful when producing images for the web and papers.

Related Links:

- Create A True Color Image in ds9 thread: using the ds9 imager to interactively create a true color 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

Create Input Images

First we need to create the input image files for dmimg2jpg. In this example, we use dmcopy to create 3 image files corresponding to the soft, medium, and hard energy bands. For the soft band (energy = 0.2 to 1.5 keV):

```
unix% punlearn dmcopy
unix% pset dmcopy infile="acisf01838N001_evt2.fits[energy=200:1500][bin x=3773:4380,y=4006:4509]"
unix% pset dmcopy outfile=soft_img.fits
unix% dmcopy
Input dataset/block specification (acisf01838N001_evt2.fits[energy=200:1500][bin x=3773:4380,y=4006:4509])
Output dataset name (soft_img.fits):
```

This dmcopy command takes the events in the given energy range and bins them to create an image of the S3 chip only (specified by the coordinates in the bin option). Since we did not give a blocking factor, the image is binned to the default value of 1; see ahelp dmcopy for information on blocking.

The command may be condensed to a single line instead. For the medium band (energy = 1.5 to 2.5 keV):

```
unix% dmcopy "acisf01838N001_evt2.fits[energy=1500:2500][bin x=3773:4380,y=4006:4509]" med_img.fits
```

And, finally, for the hard band (energy = 2.5 to 8.0 keV):

```
unix% dmcopy "acisf01838N001_evt2.fits[energy=2500:8000][bin x=3773:4380,y=4006:4509]" hard_img.fits
```

Create a True Color Image

Now we can use dmimg2jpg to combine the the soft, medium and hard images created in the previous section into a single RGB image:

```
unix% punlearn dmimg2jpg
unix% pset dmimg2jpg infile=soft_img.fits
unix% pset dmimg2jpg greenfile=med_img.fits
unix% pset dmimg2jpg bluefile=hard_img.fits
unix% pset dmimg2jpg outfile=truecolor_1838.jpg
unix% pset dmimg2jpg maxred=1 maxblue=1 maxgreen=1
unix% pset dmimg2jpg gridsize=60
unix% pset dmimg2jpg fontsize=1
unix% pset dmimg2jpg showaimpoint=no
unix% dmimg2jpg
Input file name (red image if true color) (soft_img.fits):
Green color channel file name (med_img.fits):
Blue color channel file name (hard_img.fits):
Output jpg file name (truecolor_1838.jpg):
```

The resulting image looks like Figure 1 . You can check the parameter file with plist dmimg2jpg.

A few notes on the parameters used above: maxred, maxblue, and maxgreen control the maximum pixel value to be used in the red, green, and blue files respectively. Experiment with these parameters to bring out the features in which you are interested. In general, lower maximum pixel values will bring out fainter features; for unsmoothed images, values near 1 may be necessary to display background noise.

The parameter showaimpointcontrols whether a cross is marked on the output image at the location of the aimpoint. By default, a sky grid is overlayed on the image, so gridsize=60 draws a grid line every 60 arcses. fontsize controls the size of font marking RA and Dec.

There is also a `regionfile` parameter that allows users to overlay a FITS formatted region (e.g. to mark the locations of sources). This option is discussed further in the Overlaying a Source List section.

Real vs. Virtual Image Files

It is also possible to use virtual files as input to `dmimg2jpg`, bypassing the `dmcopy` steps. To repeat the previous command with virtual file syntax:

```
unix% punlearn dmimg2jpg
unix% pset dmimg2jpg infile="acisf01838N001_evt2.fits[energy=200:1500][bin x=3773:4380,y=4006:4509]"
unix% pset dmimg2jpg greenfile="acisf01838N001_evt2.fits[energy=1500:2500][bin x=3773:4380,y=4006:45
unix% pset dmimg2jpg bluefile="acisf01838N001_evt2.fits[energy=2500:8000][bin x=3773:4380,y=4006:450
unix% pset dmimg2jpg outfile=virtual_1838.jpg
unix% pset dmimg2jpg maxred=1 maxblue=1 maxgreen=1
unix% pset dmimg2jpg gridsize=60
unix% pset dmimg2jpg fontsize=1
unix% pset dmimg2jpg showaimpoint=no
unix% dmimg2jpg
Input file name (red image if true color) (acisf01838N001_evt2.fits[energy=200:1500][bin x=3773:4380
Green color channel file name (acisf01838N001_evt2.fits[energy=1500:2500][bin x=3773:4380,y=4006:450
Blue color channel file name (acisf01838N001_evt2.fits[energy=2500:8000][bin x=3773:4380,y=4006:4509
Output jpg file name (virtual_1838.jpg):
```

Using this method is useful if disk space is an issue, since the intermediate files aren't written out.

Using the color_image Script

If you are running this thread repeatedly, e.g. to experiment with different parameters, you may wish to use the color_image script. The most recent version of color_image is v3.0 (27 February 2001):

```
unix% grep Version `which color_image`
# Version: 3.0 (27 February 2001)
```

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.

Executing the script without any input shows the syntax:

```
unix% color_image
Usage: color_image −i infile −o outfile −x xcenter −y ycenter −w width −b binfac
                  −S minsoft:maxsoft −M minmed:maxmed −H minhard:maxhard
       xcenter, ycenter − Image Center in Sky Coordinates (X,Y)
       width            − Size of image in X and Y
       binfac           − Binning factor in X and Y
       minsoft:maxsoft  − Min, Max energies in eV for soft band
       minmed:maxmed    − Min, Max energies in eV for medium band
       minhard:maxhard  − Min, Max energies in eV for hard band
Help:  color_image −h   − Print this message
```

To run the example used before in this thread:

```
unix% color_image −i acisf01838N001_evt2.fits −o script_1838.jpg −x 4076 −y 4257 −w 250 −b 1 \
      −S 200:1500 −M 1500:2500 −H 2500:8000
```

The script then prints the command that is running:

```
dmimg2jpg infile=acisf01838N001_evt2.fits[events][energy=200:1500][bin x=3951.5:4201.5:1,y=4132.5:4382.5:1
outfile=script_1838.jpg greenfile=acisf01838N001_evt2.fits[events][energy=1500:2500][bin
x=3951.5:4201.5:1,y=4132.5:4382.5:1] bluefile=acisf01838N001_evt2.fits[events][energy=2500:8000][bin
x=3951.5:4201.5:1,y=4132.5:4382.5:1] clobber=yes
```

Overlaying a Source List

A. On an X−ray Image

It is straightforward to overlay a source list from one of the detect tools, onto the x−ray image; the file may be either FITS or ASCII format. Using the `regionfile` parameter in `dmimg2jpg`:

```
unix%  cat sources.reg
# Region file format: CIAO version 1.0
circle(4071,4247,75)
circle(4245,4092,20)

unix% punlearn dmimg2jpg
unix% pset dmimg2jpg infile=soft_img.fits
unix% pset dmimg2jpg greenfile=med_img.fits
unix% pset dmimg2jpg bluefile=hard_img.fits
unix% pset dmimg2jpg outfile=overlay_1838.jpg
unix% pset dmimg2jpg maxred=1 maxblue=1 maxgreen=1
unix% pset dmimg2jpg gridsize=60
unix% pset dmimg2jpg fontsize=1
unix% pset dmimg2jpg showaimpoint=no
unix% pset dmimg2jpg regionfile="region(sources.reg)"
unix% pset dmimg2jpg regioncolor=")colors.red"
unix% dmimg2jpg
Input file name (red image if true color) (soft_img.fits):
Green color channel file name (med_img.fits):
Blue color channel file name (hard_img.fits):
Output jpg file name (overlay_1838.jpg):
```

The regions are shown in Figure 2 . You can see a list of available colors by doing "`plist colors`"; note the close−parenthesis needed in the `regioncolor` syntax.

B. On an Optical Image

Adding a source list on top of an optical image requires the two data sets to have their WCS information in image coordinates. The following workaround is a bit lengthy, but produces nice results.

First, convert the source list from Chandra physical pixels to image coordinates in ds9:

- Load the x−ray image: `unix% ds9 acisf01838N001_evt2.fits &`
- Region −> Load Regions... −> sourcelist.fits[SRCLIST]
- Region −> File Coordinate System −> Equatorial J2000
- Region −> Save Regions... −> sourcelist_J2000.reg
- Load the optical image:

1. Frame −> New Frame
2. Open... −> optical.fits
3. Frame −> Match Frames −> WCS

The final step is need to make sure that the WCS axes are oriented the same for both images (e.g. North is "up").

- Region −> Load Regions... −> sourcelist_J2000.reg
- Region −> File Coordinate System −> Image
- Region −> Save Regions... −> sourcelist_image.reg

Now that the source list is in the coordinates of the optical image, use the `dmmakereg` tool to convert it to FITS format:

```
unix% punlearn dmmakereg
unix% dmmakereg region="region(sourcelist_image.reg)" outfile=region_optical.fits
```

Finally, run `dmimg2jpg` with `regionfile="region(region_optical.fits)"` to create the image.

```
Parameters for /home/username/cxcds_param/dmimg2jpg.par


#-------------------------------------------------------------------------
# dmimg2jpg.par                 Parameter file for dmimg2jpg tool
#-------------------------------------------------------------------------
# Input files.  infile = redfile for true color output
#-------------------------------------------------------------------------
        infile = soft_img.fits    Input file name (red image if true color)
     greenfile = med_img.fits     Green color channel file name
      bluefile = hard_img.fits    Blue color channel file name
#-------------------------------------------------------------------------
# Output files.  Currently to get a postscript file you need
# to make the jpeg file.
#-------------------------------------------------------------------------
       outfile = truecolor_1838.jpg Output jpg file name
#-------------------------------------------------------------------------
# In greenfile == 'none'  use a lookup table
#-------------------------------------------------------------------------
      (lutfile = )lut.grey -> /soft/ciao/data/grey.lut) Colormap file
 (colorstretch = 1)              Color lookup strech factor
   (colorshift = 0)              Color lookup table shift
       (invert = no)             Invert colors
#-------------------------------------------------------------------------
# Scaling functions.  Use INDEF to get min and max of data
#-------------------------------------------------------------------------
(scalefunction = log)            Scaling function(log|linear|power)
   (scaleparam = 3)              Scaling parameter (for non-linear scalefunction)
       (minred = INDEF)          Minimum value for the red color channel
     (mingreen = INDEF)          Minimum value for the green color channel
      (minblue = INDEF)          Minimum value for the blue color channel
       (maxred = 1)              Maximum value for the red color channel
     (maxgreen = 1)              Maximum value for the green color channel
      (maxblue = 1)              Maximum value for the blue color channel
#-------------------------------------------------------------------------
# Regions, aimpoint and grid overlay
#-------------------------------------------------------------------------
   (regionfile = )               Region overlay file
  (regioncolor = )colors.green -> 0 1 0) Region color triple
```

```
   (regionopt = individual)      Option of region shape drawing method (individal|combine)
(showaimpoint = no)              Put crosshair at aimpoint
   (showlabel = no)              Label the contours?
    (showgrid = yes)             Show grid on image
   (gridcolor = )colors.white -> 1 1 1) Grid color triple
    (gridsize = 60)              Gridsize [arcsec]
    (fontsize = 1)               Font label size
      (psfile = )                Optional post script file name
     (verbose = 0)               Level of verbose output
     (clobber = no)              Clobber existing outputs?
        (mode = ql)
```

History

16 Dec 2004	reviewed for CIAO 3.2: no changes
21 Dec 2005	reviewed for CIAO 3.3: no changes
01 Dec 2006	reviewed for CIAO 3.4: no changes
08 Mar 2007	added link to Create A True Color Image in ds9 thread

URL: http://cxc.harvard.edu/ciao/threads/true_color/ Last modified: 8 Mar 2007

Image 1: X−ray true−color image of G21.5−09

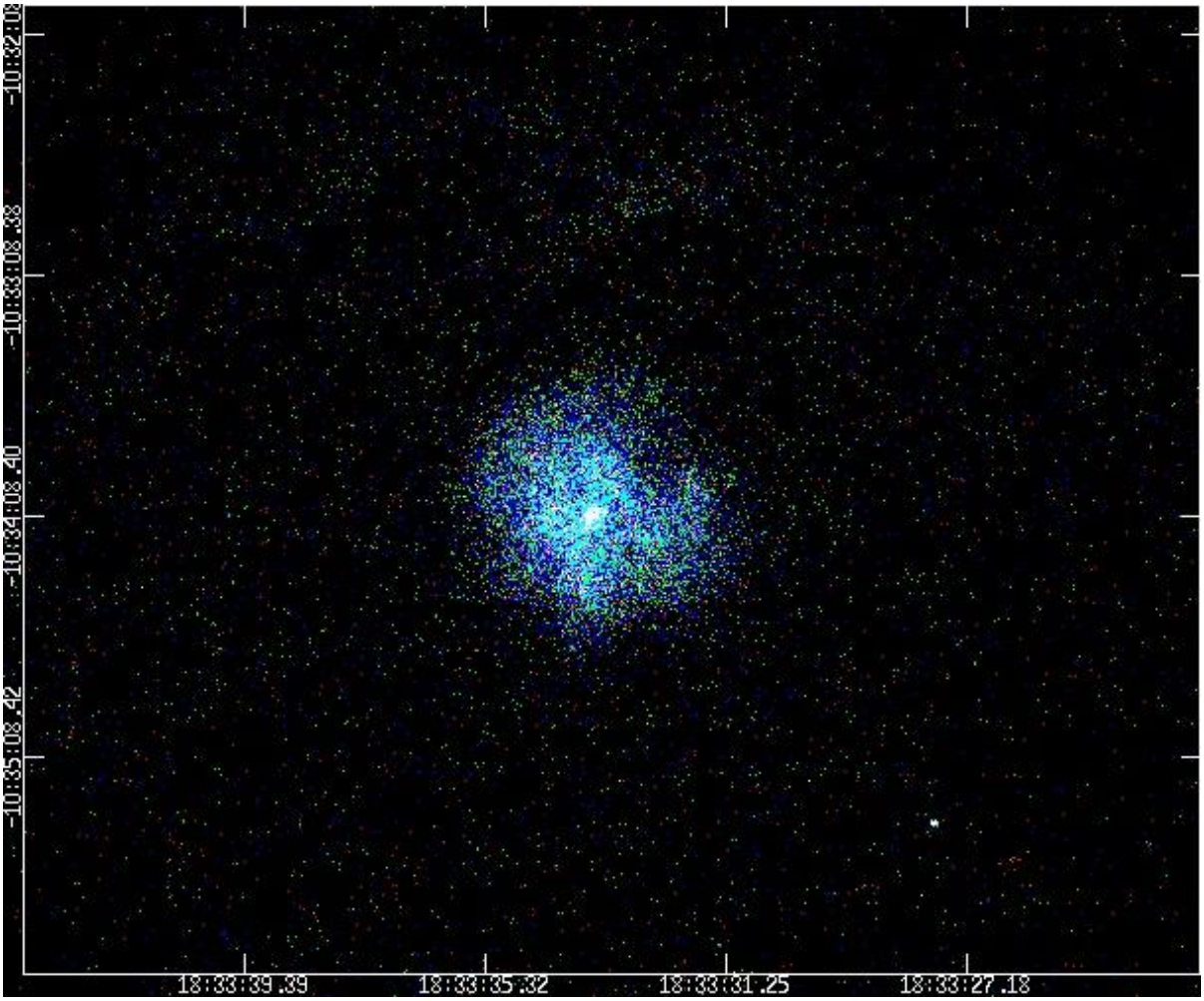
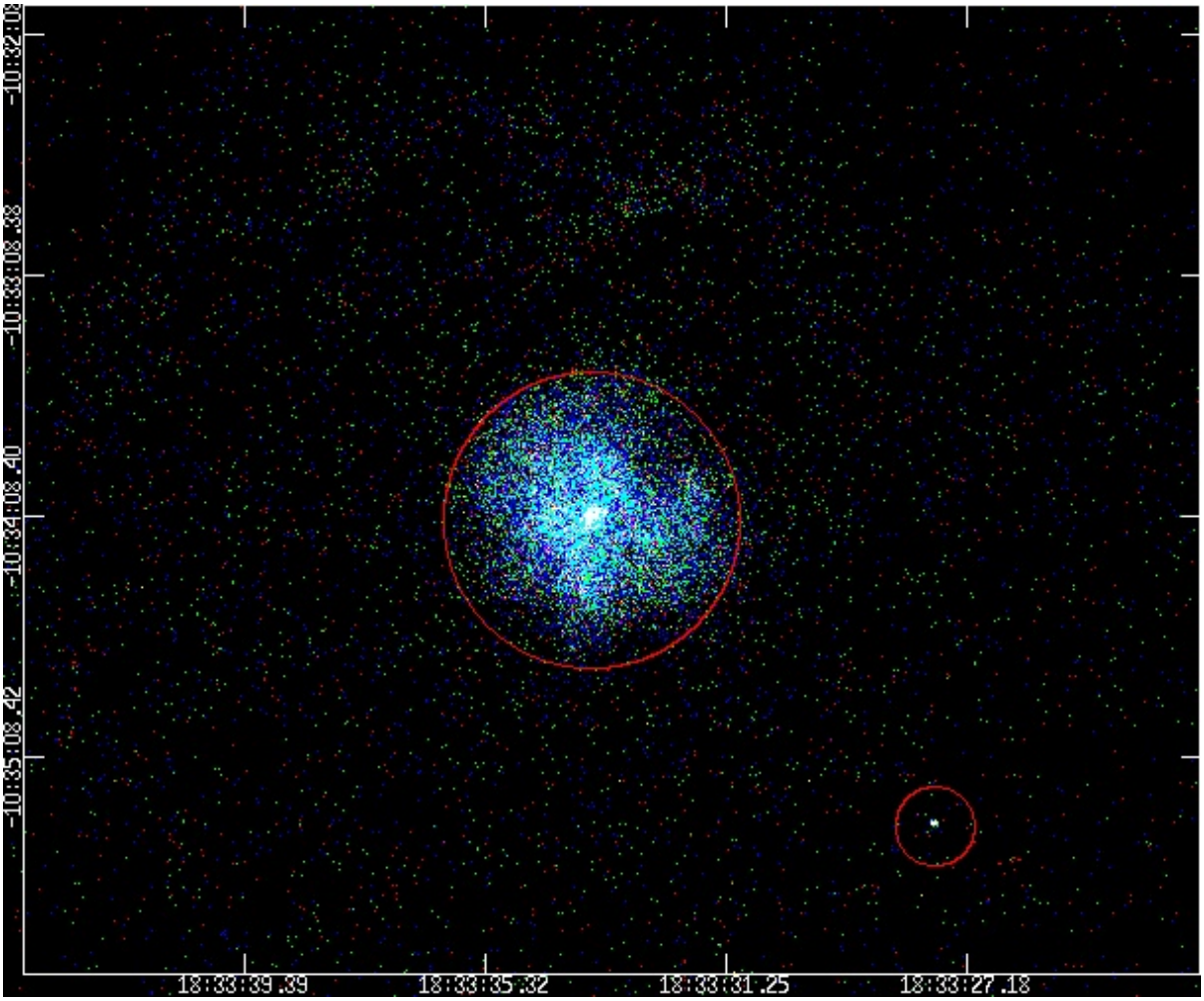



Image 2: True−color image with source list overlaid