Running wavdetect

CIAO 3.4 Science Threads

Table of Contents

Running wavdetect

CIAO 3.4 Science Threads

Overview

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

Synopsis:

`wavdetect` operates on the input in two stages. First it detects possible source pixels in a dataset by repeatedly correlating it with "Mexican Hat" wavelet functions with different scale sizes. Pixels with sufficiently large positive correlation values are removed from the image as assumed sources, and subsequent correlations are performed at the same scale. The second stage generates a source list from information from the first stage at each wavelet scale.

Purpose:

To illustrate several ways to use the Detect tool `wavdetect`.

Read this thread if:

if you want to detect sources in an ACIS or HRC observation. This tool separates closely spaced point sources and finds extended sources so long as wavlet scales are chosen appropriately.

Related Links:

- Analysis Guide: Extended Sources
- Detect manual: other examples of using this tool are available in the "Running wavdetect" chapter.

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

Getting Started

Sample ObsID used: 578 (ACIS−S, 3C 295)

File types needed: evt2

The filtered event file, exposure maps, and output source lists for this thread are available in wavdetect.tar.gz (6.1 MB tarred & gzipped; 13 MB unpacked).

Due to the heavy computational load, it is recommended to run `wavdetect` on images no larger than 1024x1024; this limit is affected by the available memory as well. In the following examples, a full resolution 512x512 image centered on the cluster core is used:

```
unix% dmcopy \
      "acisf00578N002_evt2.fits[ccd_id=7][bin x=3848:4360:1.0,y=3860:4372:1.0]" \
      s3_img.fits
```

To view the file, load it into ds9:

```
unix% ds9 s3_img.fits &
```

which produces Figure 1 📷.

Running Wavdetect

A Simple Example

To run `wavdetect` with the basic parameters set, it is necessary to specify the input and output filenames only. Note that it is not required to create an ASCII version of the region file; if the `regfile` parameter is left blank, it will not be used.

```
unix% punlearn wavdetect
unix% pset wavdetect infile=s3_img.fits
unix% pset wavdetect outfile=s3_src.fits
unix% pset wavdetect scellfile=s3_scell.fits
unix% pset wavdetect imagefile=s3_imgfile.fits
unix% pset wavdetect defnbkgfile=s3_nbgd.fits
unix% pset wavdetect regfile=s3_src.reg
unix% wavdetect
Input file name (s3_img.fits):
Output source list file name (s3_src.fits):
Output source cell image file name (s3_scell.fits):
Output reconstructed image file name (s3_imgfile.fits):
Output normalized background file name (s3_nbgd.fits):
```

The contents of the parameter file may be checked using plist wavdetect.

There are many outputs from the tool:

- `outfile`: the FITS source list; contains same sources as `regfile` (see below)
- `scellfile`: the source cell map shows the image pixels which were used to estimate source properties.
- `imagefile`: the reconstructed source image is the combined correlation maxima regions, which correspond to probable sources. It is essentially what the data would look like if there were no background and noise fluctuations.
- `defnbkgfile`: the normalized background derived from the data after the identified sources have been taken out.
- `regfile`: the ASCII region file; contains same sources as `outfile` (see above).

Full descriptions of these files are given in the "Wavdetect Input Parameters & Data Products Reference" chapter of the Detect manual.

To display the image with the detections overlaid:

```
unix% ds9 s3_img.fits &
```

Load the source list from either file (Region −> Load Regions... −> `s3_src.fits[SRCLIST]` ***OR*** `s3_src.reg`). The results are shown in Figure 2.

Key parameters that you may wish to vary are `scales` and `sigthresh`. The `scales` parameter is a list of radii (in image pixels) of Mexican Hat wavelet functions; a wavelet transform is performed for each scale in the list. Usually a list of 5 scales is chosen, and each succeeding value is a factor of 2 or SQRT(2) larger than the previous one. The primary concern is to match the first scale size to the PSF. The choice of how many scaled wavelets to use depends somewhat on the available computing resources.

The `sigthresh` parameter is the significance threshold for source detection. Specifying a `sigthresh` that is proportional to the inverse of the number of pixels in the image allows ~1 false source per field (i.e. the default value of 10^{-6} corresponds to one spurious source in a 1000x1000 pixel map).

The result of running `wavdetect` on the same image with `scales="1.0 2.0 4.0 8.0 16.0"` and `sigthresh=4e−06` are shown in Figure 3 (zooming in reveals the core and both hotspots of this radio galaxy at the center of the cluster). The parameter file used for this run is given here.

With an Exposure Map

In regions with substantial exposure variations, `wavdetect` may inaccurately estimate detection significance. Using an exposure map can suppress false positives (erroneous detections) and can help refine source property estimates. On account of this, the number of sources detected with an exposure map may be different than the number detected without one.

The exposure map used in this example is included in the `wavdetect.tar.gz` file and was created by following the Compute an ACIS Exposure Map (Single Chip) thread. For the purpose of illustration, we used a monochromatic instrument map of 0.71 keV. When creating your own exposure map, be sure that the exposure map and input image are congruent (i.e. same size, image center, and binning factor).

Run `wavdetect`:

```
unix% punlearn wavdetect
unix% pset wavdetect infile=s3_img.fits
unix% pset wavdetect outfile=s3_expmap_src.fits
unix% pset wavdetect scellfile=s3_expmap_scell.fits
unix% pset wavdetect imagefile=s3_expmap_imgfile.fits
unix% pset wavdetect defnbkgfile=s3_expmap_nbgd.fits
unix% pset wavdetect regfile=s3_expmap_src.reg
unix% pset wavdetect expfile=s3_0.71keV_expmap.fits
unix% wavdetect
Input file name (s3_img.fits):
Output source list file name (s3_expmap_src.fits):
Output source cell image file name (s3_expmap_scell.fits):
Output reconstructed image file name (s3_expmap_imgfile.fits):
Output normalized background file name (s3_expmap_nbgd.fits):
```

The contents of the parameter file may be checked using plist wavdetect.

To display the image with an overlay of the source detections:

```
unix% ds9 s3_img.fits &
```

Load the source list from either file (Region −> Load Regions... −> `s3_expmap_src.fits[SRCLIST]` ***OR*** `s3_expmap_src.reg`). The results are shown in Figure 5.

There isn't any significant exposure variation across the field in this example, so the source properties do not change much in the exposure map run.

To examine the source properties, use either `dmlist` with the column names of interest:

```
unix% dmlist "s3_expmap_src.fits[cols RA,DEC,PSF_SIZE]" data

--------------------------------------------------------------------------------
Data for Table Block SRCLIST
--------------------------------------------------------------------------------

ROW    RA                   DEC                  PSF_SIZE

     1       212.8847083521       52.1878910087     0.61479133367538
     2       212.8714240484       52.2259177730     0.62263220548630
     3       212.8640791885       52.1921762232     0.47023788094521
     4       212.8558538789       52.1797387065     0.51768761873245
     5       212.8475613372       52.2255903030     0.53206640481949
     6       212.8362796737        52.202181660     0.40190583467484
     7       212.8362660158       52.2366611370     0.65922302007675
     8       212.8351087846       52.2029039026     0.40246155858040
     9       212.8310964079       52.2335976922     0.60252219438553
    10       212.8236615527       52.2284789414     0.55507236719131
    11       212.8067711419       52.2281781731     0.59245705604553
```

or *prism*:

```
unix% prism s3_expmap_src.fits &
```

Caveat: a detection tool, not a flux tool

This tool is designed to be used as a source *detection* algorithm, and only secondarily as a source flux measurement tool. Flux estimation with `wavdetect` is generally quite reliable, but there is a small fraction of cases where the algorithm will fail to return useful source parameter estimates. This is inevitable, as there is no perfect source detection algorithm.

In section 3.2.2 of "A Wavelet−Based Algorithm for the Spatial Analysis of Poisson Data" (P. E. Freeman, et al. 2002, ApJS, 138, 185) it says:

```
"We note two situations where care must be exercised in interpreting
results. First, the source cell for an [apparently] extended source
may be too small if the smoothing scale is ~ [the PSF scale]."
```

The workaround for this is to rerun `wavdetect` with the parameter `scales` set to "`4.0 5.0`" to get a better accounting of the source in question.

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

```
#
#   parameter file for wavdetect
#
#
#   input
#
        infile = s3_img.fits      Input file name
#
#   output
#
       outfile = s3_src.fits      Output source list file name
     scellfile = s3_scell.fits    Output source cell image file name
     imagefile = s3_imgfile.fits  Output reconstructed image file name
   defnbkgfile = s3_nbgd.fits     Output normalized background file name
#
#   scales
#
        scales = 2.0 4.0          wavelet scales (pixels)
      (regfile = s3_src.reg)      ASCII regions output file
#
#   output options
#
      (clobber = no)              Overwrite existing outputs?
       (kernel = default)         Output file format (fits|iraf|default)
     (ellsigma = 3.0)             Size of output source ellipses (in sigmas)
     (interdir = .)               Directory for intermediate outputs
#
#########################################################################
#
#   wtransform parameters
#
#
#   optional input
#
     (bkginput = )                Input background file name
  (bkgerrinput = no)              Use bkginput[2] for background error
#
#   output info
#
  (outputinfix = )                Output filename infix
#
#   output content options
#
    (sigthresh = 1e−06)           Threshold significance for output source pixel list
 (bkgsigthresh = 0.001)           Threshold significance when estimating bkgd only
#
#   exposure info
#
      (exptime = 0)               Exposure time (if zero, estimate from map itself
      (expfile = )                Exposure map file name (blank=none)
    (expthresh = 0.1)             Minimum relative exposure needed in pixel to analyze it
#
#   background
#
      (bkgtime = 0)               Exposure time for input background file
#
#   iteration info
#
      (maxiter = 2)               Maximum number of source−cleansing iterations
     (iterstop = 0.0001)          Min frac of pix that must be cleansed to continue
```

```
#
#   end of wtransform parameters
#
########################################################################
########################################################################
#
#   wrecon parameters
#
#
#   PSF size parameters
#
      (xoffset = INDEF)           Offset of x axis from optical axis
      (yoffset = INDEF)           Offset of y axis from optical axis
        (eband = 1.4967)          Energy band
      (eenergy = 0.393)           Encircled energy of PSF
     (psftable = ${ASCDS_CALIB}/psfsize20010416.fits -> /soft/ciao/data/psfsize20010416.fits) Table of PSF
#
#   end of wrecon parameters
#
########################################################################
#
#   run log verbosity and content
#
          (log = no)              Make a log file?
      (verbose = 0)               Log verbosity
#
#   mode
#
         (mode = ql)
```

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


#
#   parameter file for wavdetect
#
#
#   input
#
        infile = s3_img.fits      Input file name
#
#   output
#
       outfile = s3_src_2.fits    Output source list file name
     scellfile = s3_scell_2.fits  Output source cell image file name
     imagefile = s3_imgfile_2.fits Output reconstructed image file name
   defnbkgfile = s3_nbgd_2.fits   Output normalized background file name
#
#   scales
#
        scales = 1.0 2.0 4.0 8.0 16.0 wavelet scales (pixels)
      (regfile = s3_src_2.reg)    ASCII regions output file
#
#   output options
#
      (clobber = no)              Overwrite existing outputs?
       (kernel = default)         Output file format (fits|iraf|default)
     (ellsigma = 3)               Size of output source ellipses (in sigmas)
     (interdir = .)               Directory for intermediate outputs
#
#########################################################################
#
#   wtransform parameters
#
#
#   optional input
#
     (bkginput = )                Input background file name
  (bkgerrinput = no)              Use bkginput[2] for background error
#
#   output info
#
  (outputinfix = )                Output filename infix
#
#   output content options
#
    (sigthresh = 4e−06)           Threshold significance for output source pixel list
 (bkgsigthresh = 0.001)           Threshold significance when estimating bkgd only
#
#   exposure info
#
      (exptime = 0)               Exposure time (if zero, estimate from map itself
      (expfile = )                Exposure map file name (blank=none)
    (expthresh = 0.1)             Minimum relative exposure needed in pixel to analyze it
#
#   background
#
      (bkgtime = 0)               Exposure time for input background file
#
#   iteration info
#
      (maxiter = 2)               Maximum number of source−cleansing iterations
     (iterstop = 0.0001)          Min frac of pix that must be cleansed to continue
#
#   end of wtransform parameters
#
########################################################################
########################################################################
#
#   wrecon parameters
#
#
#   PSF size parameters
#
      (xoffset = INDEF)           Offset of x axis from optical axis
      (yoffset = INDEF)           Offset of y axis from optical axis
        (eband = 1.4967)          Energy band
      (eenergy = 0.393)           Encircled energy of PSF
     (psftable = ${ASCDS_CALIB}/psfsize20010416.fits −> /soft/ciao/data/psfsize20010416.fits) Table
#
#   end of wrecon parameters
#
########################################################################
#
#   run log verbosity and content
#
          (log = no)              Make a log file?
      (verbose = 0)               Log verbosity
#
#   mode
```

```
#
         (mode = ql)
```

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


#
#   parameter file for wavdetect
#
#
#   input
#
        infile = s3_img.fits      Input file name
#
#   output
#
       outfile = s3_expmap_src.fits Output source list file name
     scellfile = s3_expmap_scell.fits Output source cell image file name
     imagefile = s3_expmap_imgfile.fits Output reconstructed image file name
   defnbkgfile = s3_expmap_nbgd.fits Output normalized background file name
#
#   scales
#
        scales = 2.0 4.0          wavelet scales (pixels)
      (regfile = s3_expmap_src.reg) ASCII regions output file
#
#   output options
#
      (clobber = no)              Overwrite existing outputs?
       (kernel = default)         Output file format (fits|iraf|default)
     (ellsigma = 3.0)             Size of output source ellipses (in sigmas)
     (interdir = .)               Directory for intermediate outputs
#
#########################################################################
#
#   wtransform parameters
#
#
#   optional input
#
     (bkginput = )                Input background file name
  (bkgerrinput = no)              Use bkginput[2] for background error
#
#   output info
#
  (outputinfix = )                Output filename infix
#
#   output content options
#
    (sigthresh = 1e−06)           Threshold significance for output source pixel list
 (bkgsigthresh = 0.001)           Threshold significance when estimating bkgd only
#
#   exposure info
#
      (exptime = 0)               Exposure time (if zero, estimate from map itself
      (expfile = s3_0.71keV_expmap.fits) Exposure map file name (blank=none)
    (expthresh = 0.1)             Minimum relative exposure needed in pixel to analyze it
#
```

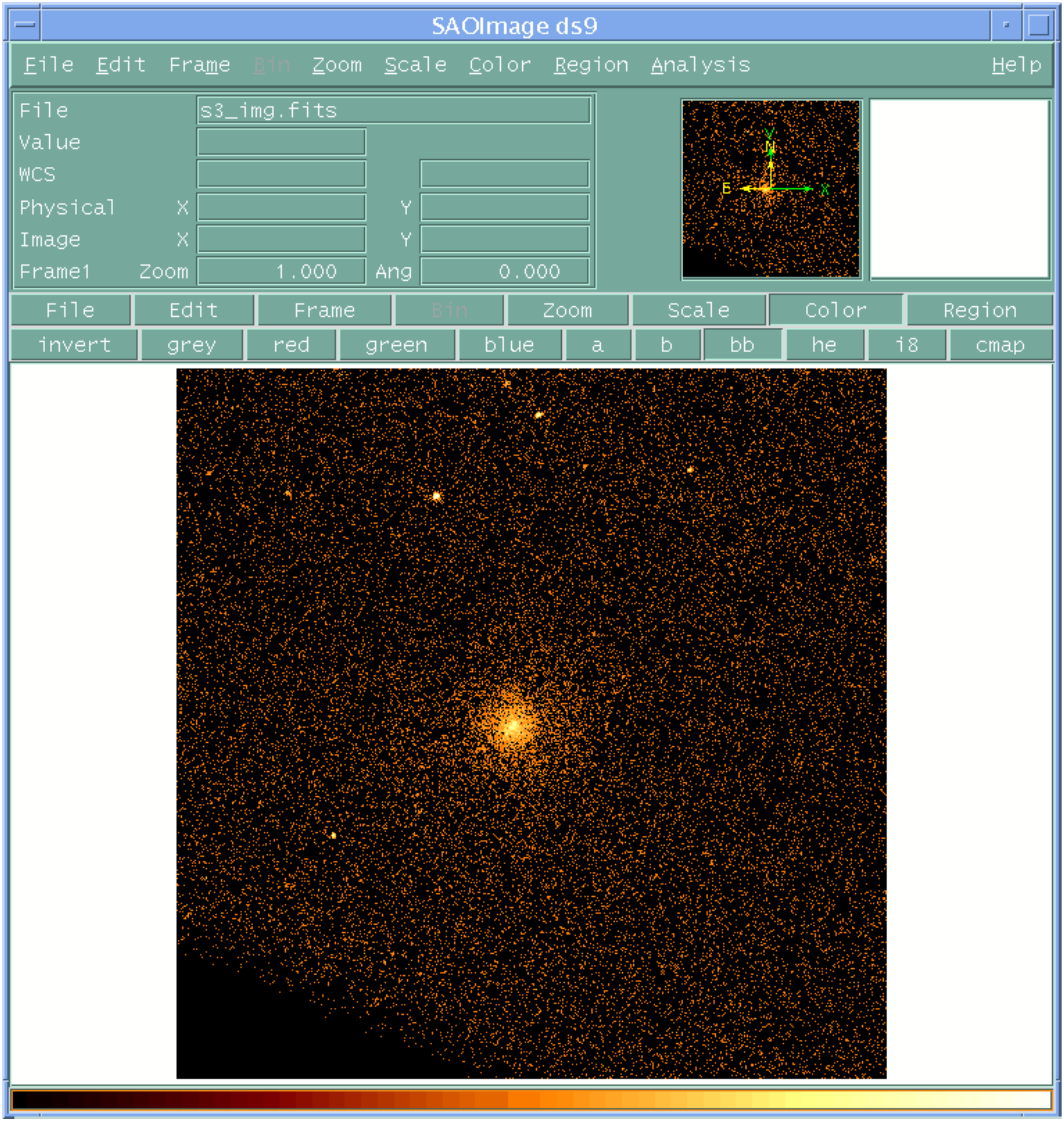
```
#   background
#
      (bkgtime = 0)               Exposure time for input background file
#
#   iteration info
#
      (maxiter = 2)               Maximum number of source−cleansing iterations
     (iterstop = 0.0001)          Min frac of pix that must be cleansed to continue
#
#   end of wtransform parameters
#
########################################################################
########################################################################
#
#   wrecon parameters
#
#
#   PSF size parameters
#
      (xoffset = INDEF)           Offset of x axis from optical axis
      (yoffset = INDEF)           Offset of y axis from optical axis
        (eband = 1.4967)          Energy band
      (eenergy = 0.393)           Encircled energy of PSF
     (psftable = ${ASCDS_CALIB}/psfsize20010416.fits −> /soft/ciao/data/psfsize20010416.fits) Table
#
#   end of wrecon parameters
#
########################################################################
#
#   run log verbosity and content
#
          (log = no)              Make a log file?
      (verbose = 0)               Log verbosity
#
#   mode
#
         (mode = ql)
```

History

03 Jan 2005	reviewed for CIAO 3.2: no changes
03 Jun 2005	updated links for CIAO 3.2 version of Detect manual
19 Dec 2005	updated for CIAO 3.3: updated files in detect data tarfile
01 Dec 2006	reviewed for CIAO 3.4: no changes

URL: http://cxc.harvard.edu/ciao/threads/wavdetect/ Last modified: 1 Dec 2006

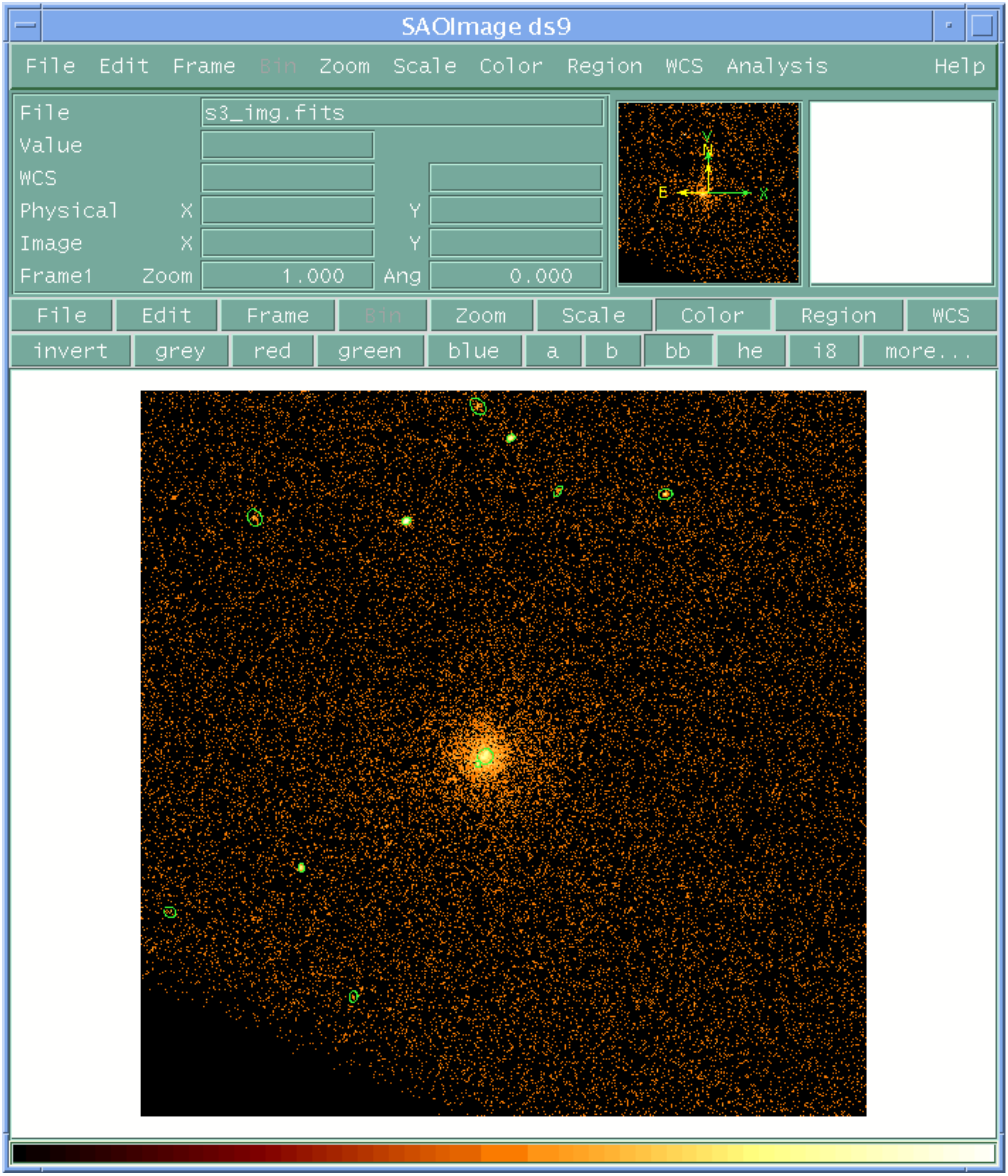
Image 1: Full resolution image of the cluster core


SAOImage ds9
File Edit Frame Bin Zoom Scale Color Region Analysis Help
Value
WCS
Physical X Y
Image X Y
File Edit Frame Bin Zoom Scale Color Region
invert grey red green blue a b bb he i8 cmap
N
E
Y
X

File s3_img.fits

Frame1 Zoom 1.000 Ang 0.000

Image 2: A simple example − chip S3 with detections overlaid

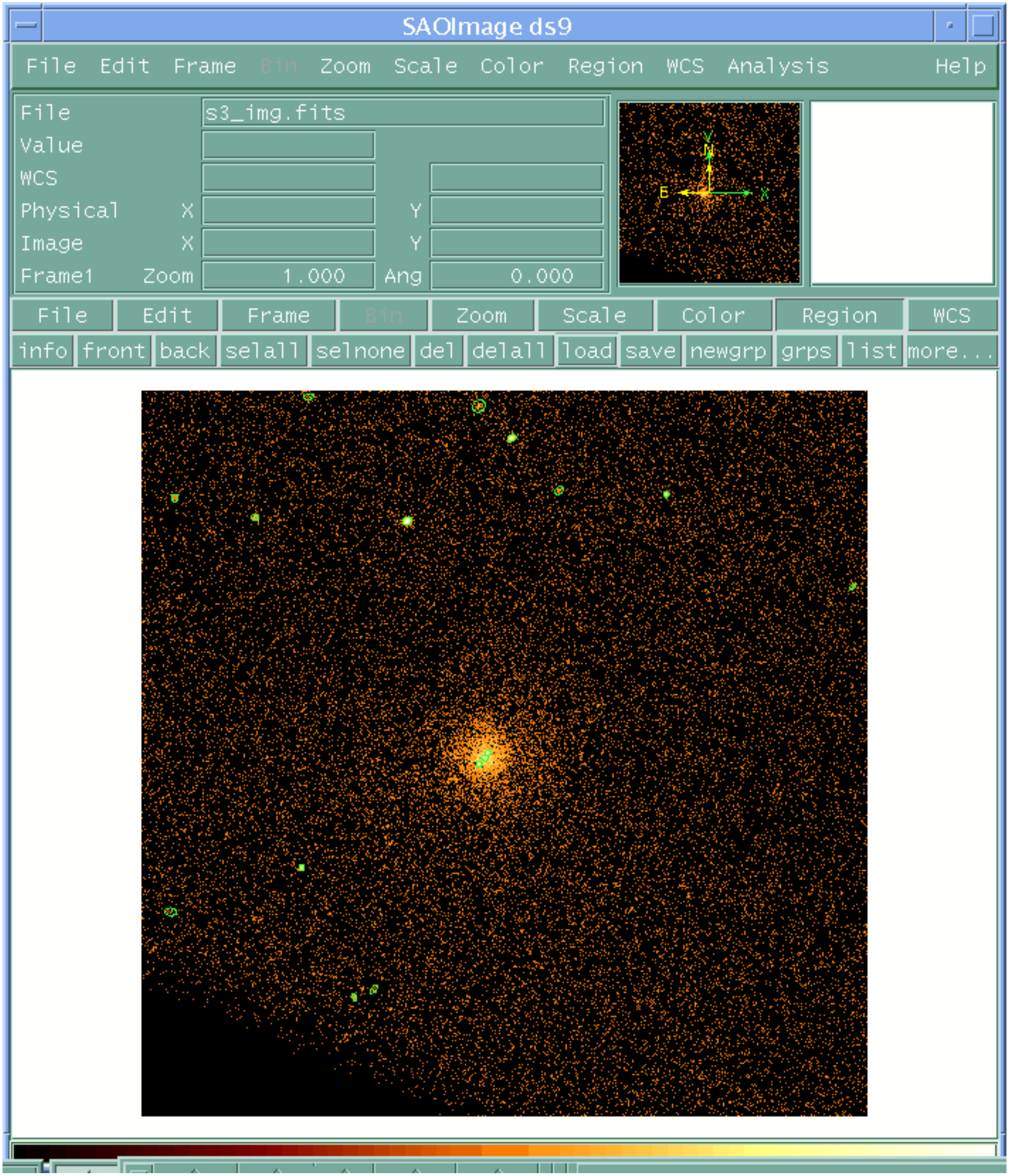


SAOImage ds9
File Edit Frame Bin Zoom Scale Color Region WCS Analysis Help
Value
WCS
Physical X Y
Image X Y
E N Y X
File Edit Frame Bin Zoom Scale Color Region WCS
invert grey red green blue a b bb he i8 more...

File s3_img.fits

Frame1 Zoom 1.000 Ang 0.000

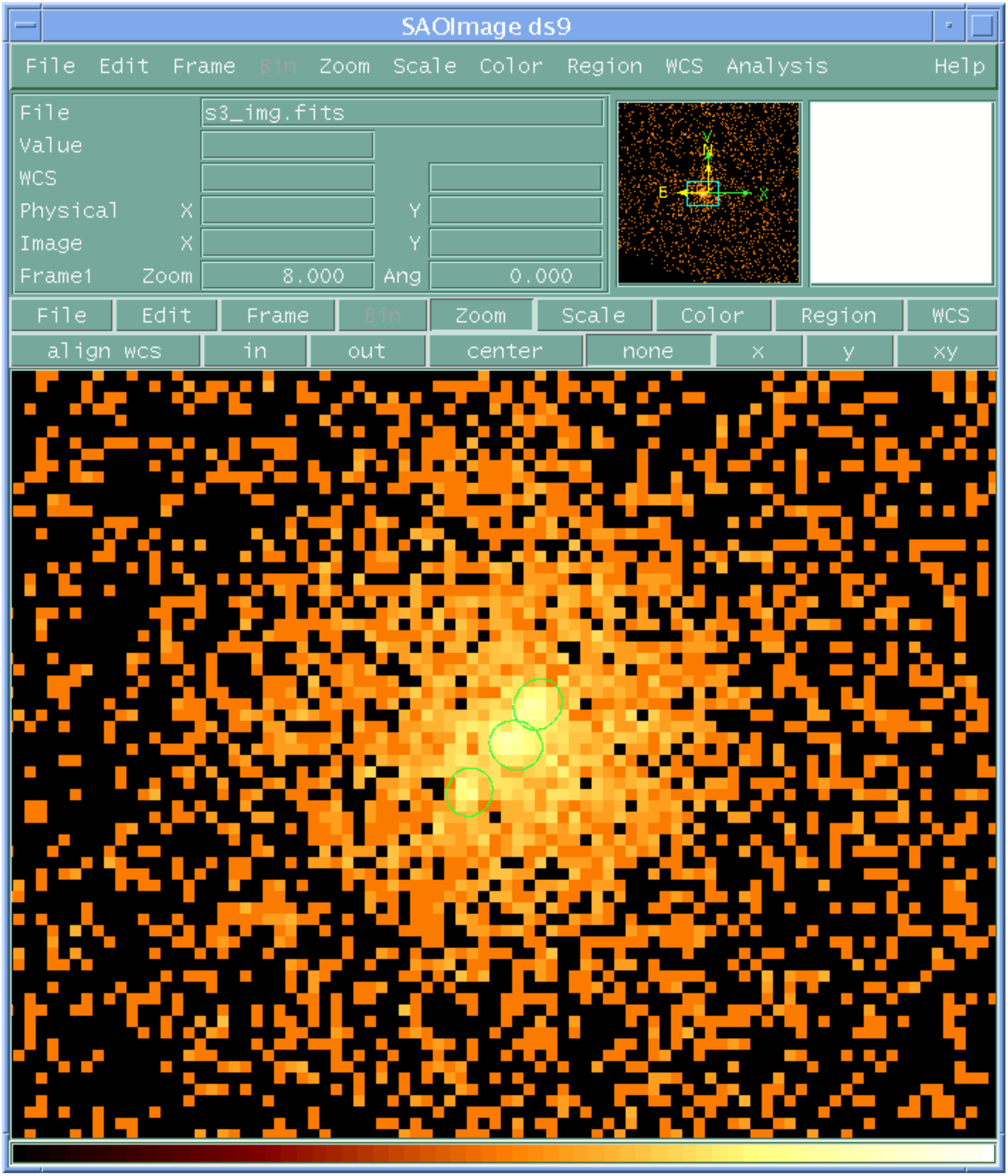
Image 3: Chip S3 with detections overlaid (different scales and sigthresh)


SAOImage ds9
File Edit Frame Bin Zoom Scale Color Region WCS Analysis Help
Value
WCS
Physical X Y
Image X Y
N
E
X
Y
File Edit Frame Bin Zoom Scale Color Region WCS
info front back selall selnone del delall load save newgrp grps list more...

File s3_img.fits

Frame1 Zoom 1.000 Ang 0.000

Image 4: Galaxy core and hotspots (zoom of Image 3)

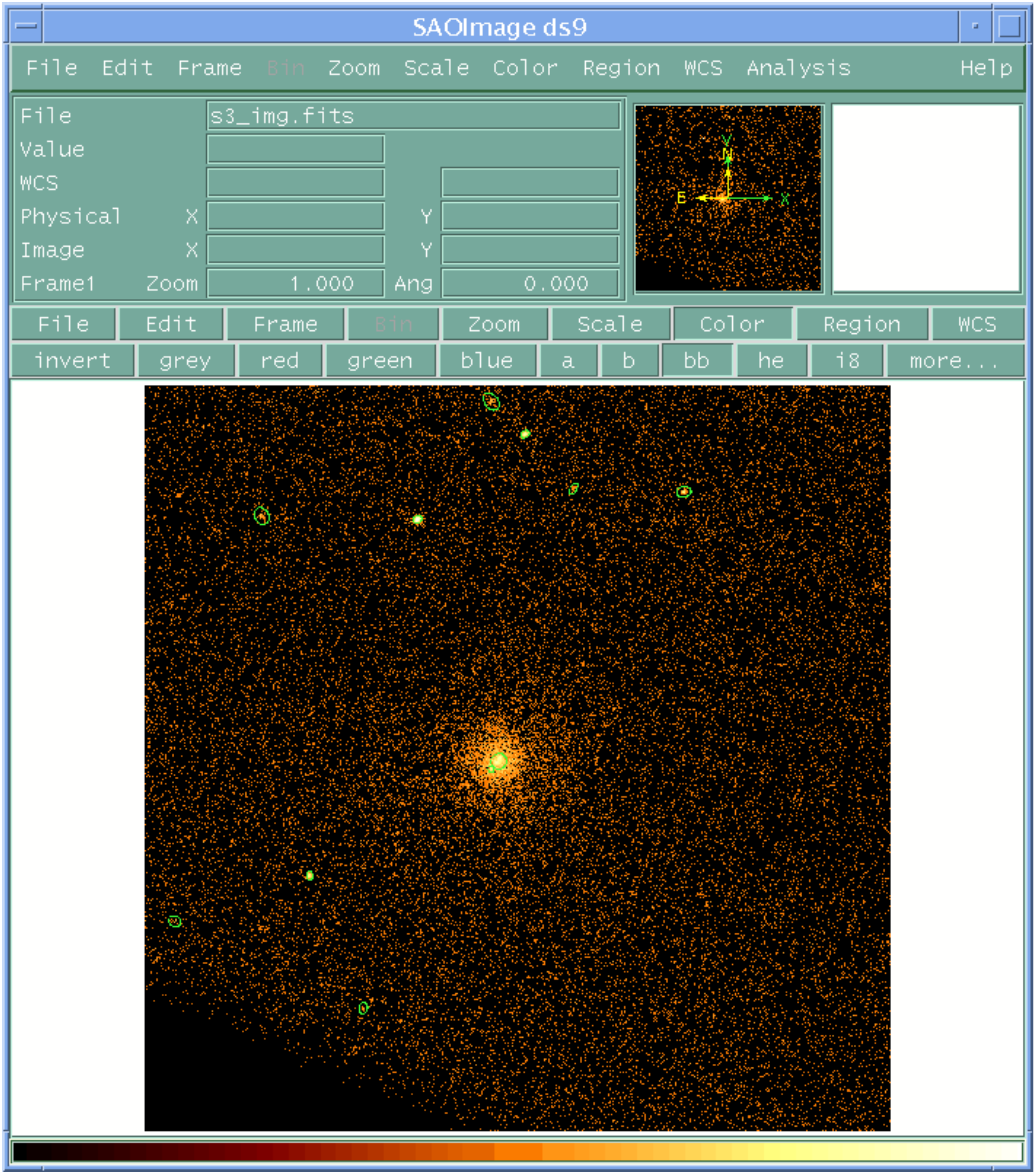


SAOImage ds9
File Edit Frame Bin Zoom Scale Color Region WCS Analysis Help
Value
WCS
Physical X Y
Image X Y
N
E
X
File Edit Frame Bin Zoom Scale Color Region WCS
align wcs in out center none x y xy

File s3_img.fits

Frame1 Zoom 8.000 Ang 0.000

Image 5: With an exposure map − chip S3 with detections overlaid


SAOImage ds9
File Edit Frame Bin Zoom Scale Color Region WCS Analysis Help
File
Value
WCS
Physical X Y
Image X Y
Frame1
Y
N
E
X
File Edit Frame Bin Zoom Scale Color Region WCS
invert grey red green blue a b bb he i8 more...

File s3_img.fits
Zoom 1.000 Ang 0.000