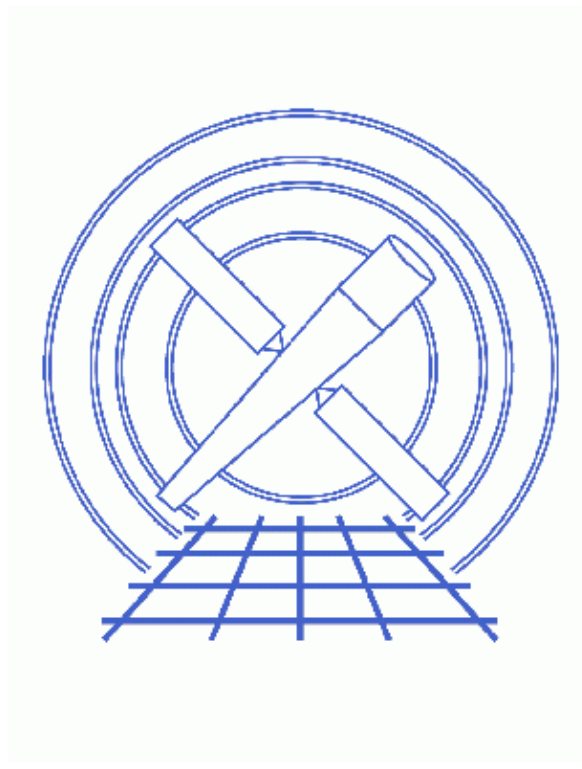


## Create an Image of Diffuse Emission



### *CIAO 3.4 Science Threads*

# Table of Contents

- ***Get Started***
- ***Identify and Remove Point Sources***
  1. Create an Image of the Region (dmcopy)
  2. Detect and Remove the Point Sources (wavdetect)
  3. Create Background Regions
  4. Fill in the Holes (dmfilth)
- ***Smooth the Image***
- ***Exposure–Correcting the Image (Optional)***
- ***Parameter files:***
  - ◆ wavdetect
  - ◆ dmfilth
  - ◆ aconvolve
- ***History***
- ***Images***
  - ◆ Diffuse emission plus point sources
  - ◆ Detected point sources
  - ◆ Modified background regions
  - ◆ Diffuse emission only
  - ◆ Smoothed image of the diffuse emission

# Create an Image of Diffuse Emission

## *CIAO 3.4 Science Threads*

### Overview

**Last Update:** 1 Dec 2006 – updated for CIAO 3.4: uses [aconvolve](#) instead of [csmooth](#) for smoothing, updated image to match

#### **Synopsis:**

The procedure used here is intended to make a nice image for poster or paper and to aid in understanding the morphology of the extended emission. Point sources are identified and removed, filling the gaps with a sampling of the background region. The filled image is then smoothed, with the option to exposure-correct the results. *Care must be taken in the scientific interpretation of the final image as it is highly processed.*

#### **Purpose:**

To create a smoothed image of diffuse emission.

#### **Read this thread if:**

you are working with an ACIS or HRC imaging observation and would like to create an image of the diffuse emission.

#### **Related Links:**

- Analysis Guide: [Extended Sources](#)
- The [File Format](#) section of the [Using CIAO Region Files](#): describes how to display region files on an event list, which is used often in this thread.
- [Creating Source and Background Files](#): this method could be used to create the source and background region files as well.

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

---

## Get Started

**Sample ObsID used:** 315 (ACIS-S, NGC 4038/39)

**File types needed:** evt2

All the source lists used in this thread are available for download as well. Simply <Shift>-click on the filename and save it to your working directory.

## Create an Image of Diffuse Emission – CIAO 3.4

This thread uses the `mkBgReg.pl` and `mkSubBgReg.pl` scripts. The most recent version of each of these scripts is v1.1:

```
unix% grep Version `which mkBgReg.pl`  
# Version 1.1  
  
unix% grep Version `which mkSubBgReg.pl`  
# Version 1.1
```

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


---

## Identify and Remove Point Sources

### 1. Create an Image of the Region (dmcopy)

First, we create an image of the region which contains the diffuse emission using `dmcopy`, simultaneously [filtering on energy](#):

```
unix% punlearn dmcopy  
unix% dmcopy "acisf00315N002_evt2.fits[energy=300:7000][bin x=4004.5:4404.5:1,y=3625.5:4025.5:1]" \  
diff_image.fits
```

This produces the 400x400 image shown in [Figure 1](#) . It is recommended that you make the image no larger than about 512x512; otherwise [running aconvolve](#) can require large amounts of memory and excessive CPU time.

---

### 2. Detect and Remove the Point Sources (wavdetect)

Using this image, we run `wavdetect` to identify the point sources:

```
unix% punlearn wavdetect  
unix% pset wavdetect infile=diff_image.fits  
unix% pset wavdetect outfile=sources.fits  
unix% pset wavdetect scellfile=sources_scell.fits  
unix% pset wavdetect imagefile=sources_image.fits  
unix% pset wavdetect defnbkgfile=sources_bkg.fits  
unix% pset wavdetect regfile=sources.reg  
unix% pset wavdetect scales="1 2 4"  
unix% pset wavdetect ellsigma=4  
unix% wavdetect  
Input file name (diff_image.fits):  
Output source list file name (sources.fits):  
Output source cell image file name (sources_scell.fits):  
Output reconstructed image file name (sources_image.fits):  
Output normalized background file name (sources_bkg.fits):
```

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


From inspection, one can see that many of the sources detected by `wavdetect` are not point sources, but clumps of diffuse emission. Scientific judgment must be used to modify or delete regions before saving the final region file. The [Using the Output of Detect Tools](#) thread shows how to display and modify source lists.

Save the modified region file in CIAO format (Region → File format → CIAO). In this example, the modified source list has been saved as `sources_mod.reg`:

```
unix% more sources_mod.reg
```

## Create an Image of Diffuse Emission – CIAO 3.4

```
# Region file format: CIAO version 1.0
ellipse(4055.1957,3893.7645,2.9460981,2.3517561,39.833389)
ellipse(4098.8472,3656.7917,6.5121918,3.5388501,51.12962)
ellipse(4106.8252,3907.1114,3.6603396,3.0782087,43.142921)
ellipse(4127.4146,3837.6585,3.0860283,2.4005349,42.822891)
ellipse(4129.2042,3872.1511,4.04951,3.0949769,46.016918)
ellipse(4134.2593,3829.9506,4.5580454,3.6698477,67.594635)
ellipse(4143.8718,3935.2821,5.4997063,3.4211953,35.669456)
ellipse(4148.9281,3750.2253,5.2349572,3.2586305,49.783184)
ellipse(4155.016,3796.696,3.7534854,2.9886484,63.30658)
ellipse(4161.1194,3773.1045,3.1660604,2.6469104,38.654858)
ellipse(4162.3623,3767.058,3.372299,2.75225,63.434948)
ellipse(4166.8967,3881.8016,4.2776256,3.2014892,46.307049)
ellipse(4169.0361,3898.9062,5.199728,3.9067812,58.489971)
ellipse(4191.2122,3758.3247,5.4182076,4.4239564,54.529949)
ellipse(4213.0702,3963.5088,6.1489725,4.6571512,57.640621)
ellipse(4223.1034,3752.069,3.4577184,2.3349586,50.309826)
ellipse(4223.1931,3888.9691,2.8335178,2.5363314,66.179474)
ellipse(4232.1721,3956.4796,5.0121182,5.845009,54.346333)
ellipse(4254.2364,3851.1877,4.8072686,3.1549962,59.28215)
ellipse(4326.4177,3651.5823,11.655077,6.2312627,55.26606)
ellipse(4336.2644,3887.1149,3.364907,2.5080459,51.259182)
ellipse(4242.04,3682.88,3.6650522,2.60603,49.658543)
ellipse(4246.15,3837.9167,5.4618821,3.8768628,45.48172)
ellipse(4248.7547,3677.9623,6.2279682,4.0094967,51.965843)
ellipse(4281.4821,3832.7857,7.0371718,4.8427992,67.028198)
```

An image of the region with `wavdetect` detections in red and the modified sourcelist in green can be viewed in [Figure 2](#) .

---

### 3. Create Background Regions

We create a stack of background regions using the script `mkBgReg.pl`:


```
unix% mkBgReg.pl
ASCII region file (CIAO format): sources_mod.reg
Multiply source radius by: 2
Output filename: bkg.reg
```

The output file, `bkg.reg`, is simply a stack of background regions centered on the source regions with radii twice as large:

```
unix% more bkg.reg
ellipse(4055.1957,3893.7645,5.8921962,4.7035122,39.833389)
ellipse(4098.8472,3656.7917,13.0243836,7.0777002,51.12962)
ellipse(4106.8252,3907.1114,7.3206792,6.1564174,43.142921)
ellipse(4127.4146,3837.6585,6.1720566,4.8010698,42.822891)
ellipse(4129.2042,3872.1511,8.09902,6.1899538,46.016918)
ellipse(4134.2593,3829.9506,9.1160908,7.3396954,67.594635)
ellipse(4143.8718,3935.2821,10.9994126,6.8423906,35.669456)
ellipse(4148.9281,3750.2253,10.4699144,6.517261,49.783184)
ellipse(4155.016,3796.696,7.5069708,5.9772968,63.30658)
ellipse(4161.1194,3773.1045,6.3321208,5.2938208,38.654858)
ellipse(4162.3623,3767.058,6.744598,5.5045,63.434948)
ellipse(4166.8967,3881.8016,8.5552512,6.4029784,46.307049)
ellipse(4169.0361,3898.9062,10.399456,7.8135624,58.489971)
ellipse(4191.2122,3758.3247,10.8364152,8.8479128,54.529949)
ellipse(4213.0702,3963.5088,12.297945,9.3143024,57.640621)
ellipse(4223.1034,3752.069,6.9154368,4.6699172,50.309826)
ellipse(4223.1931,3888.9691,5.6670356,5.0726628,66.179474)
ellipse(4232.1721,3956.4796,10.0242364,11.690018,54.346333)
ellipse(4254.2364,3851.1877,9.6145372,6.3099924,59.28215)
ellipse(4326.4177,3651.5823,23.310154,12.4625254,55.26606)
ellipse(4336.2644,3887.1149,6.729814,5.0160918,51.259182)
ellipse(4242.04,3682.88,7.3301044,5.21206,49.658543)
```

## Create an Image of Diffuse Emission – CIAO 3.4

```
ellipse(4246.15,3837.9167,10.9237642,7.7537256,45.48172)
ellipse(4248.7547,3677.9623,12.4559364,8.0189934,51.965843)
ellipse(4281.4821,3832.7857,14.0743436,9.6855984,67.028198)
```

We do not want any of these to overlap with any source regions, so load the file in ds9 and modify the regions in cases where this occurs. Also be sure that none of the regions exceed the boundaries of the image. An image of the data with modified background regions can be viewed in [Figure 3](#) . These regions are saved as `bkg_mod.reg`:

```
unix% more bkg_mod.reg
# Region file format: CIAO version 1.0
ellipse(4055.1957,3893.7645,7.9999535,5.7154179,39.833389)
ellipse(4098.8472,3656.7917,13.024384,7.0777002,51.12962)
ellipse(4106.8252,3907.1114,4.2559794,9.1839537,43.142921)
ellipse(4127.4146,3837.6585,8.0056945,4.4763904,42.822891)
ellipse(4129.2042,3872.1511,8.09902,6.1899538,46.016918)
ellipse(4134.2593,3829.9506,9.3865666,5.5845183,67.594635)
ellipse(4143.8718,3935.2821,4.9544978,5.4075842,35.669456)
ellipse(4148.9281,3750.2253,10.469914,6.517261,49.783184)
ellipse(4155.016,3796.696,7.5069708,5.9772968,63.30658)
ellipse(4160.1194,3773.6045,7.2594208,3.6636984,38.654858)
ellipse(4162.3623,3767.058,5.1332518,4.3976302,63.434948)
ellipse(4166.8967,3881.8016,8.5552512,6.4029784,46.307049)
ellipse(4169.0361,3898.9062,10.399456,7.8135624,58.489971)
ellipse(4191.2122,3758.3247,10.836415,8.8479128,54.529949)
ellipse(4213.0702,3963.5088,5.8869509,6.606619,57.640621)
ellipse(4223.1034,3752.069,6.9154368,4.6699172,50.309826)
ellipse(4223.1931,3888.9691,5.6670356,5.0726628,66.179474)
ellipse(4232.1721,3956.4796,10.024236,11.690018,54.346333)
ellipse(4254.2364,3851.1877,9.6145372,6.3099924,59.28215)
ellipse(4326.4177,3651.5823,23.310154,12.462525,55.26606)
ellipse(4336.2644,3887.1149,6.729814,5.0160918,51.259182)
ellipse(4242.04,3682.88,12.448016,3.2562096,49.658543)
ellipse(4246.15,3837.9167,5.5514289,6.9530192,45.48172)
ellipse(4248.7547,3677.9623,12.576657,5.0829714,51.965843)
ellipse(4281.4821,3832.7857,9.8866826,7.9729324,67.028198)
```

Since the POISSON method in `dmfilth` (see the [next section](#)) expects a stack of background regions which do not include any source regions, we must subtract the corresponding source region from each background region. Here, we use the script `mkSubBgReg.pl`:

```
unix% mkSubBgReg.pl
ASCII source region file (CIAO format): sources_mod.reg
ASCII background region file (CIAO format): bkg_mod.reg
Output filename: bkg_sub.reg
```

The output region file, `bkg_sub.reg`, looks like this (note that it will not load into ds9 due to the subtracted regions):

```
unix% more bkg_sub.reg
ellipse(4055.1957,3893.7645,7.9999535,5.7154179,39.833389)-ellipse(4055.1957,3893.7645,2.9460981,2.3517
ellipse(4098.8472,3656.7917,13.024384,7.0777002,51.12962)-ellipse(4098.8472,3656.7917,6.5121918,3.53885
ellipse(4106.8252,3907.1114,4.2559794,9.1839537,43.142921)-ellipse(4106.8252,3907.1114,3.6603396,3.0782
ellipse(4127.4146,3837.6585,8.0056945,4.4763904,42.822891)-ellipse(4127.4146,3837.6585,3.0860283,2.4005
ellipse(4129.2042,3872.1511,8.09902,6.1899538,46.016918)-ellipse(4129.2042,3872.1511,4.04951,3.0949769,
ellipse(4134.2593,3829.9506,9.3865666,5.5845183,67.594635)-ellipse(4134.2593,3829.9506,4.5580454,3.6698
ellipse(4143.8718,3935.2821,4.9544978,5.4075842,35.669456)-ellipse(4143.8718,3935.2821,5.4997063,3.4211
ellipse(4148.9281,3750.2253,10.469914,6.517261,49.783184)-ellipse(4148.9281,3750.2253,5.2349572,3.25863
ellipse(4155.016,3796.696,7.5069708,5.9772968,63.30658)-ellipse(4155.016,3796.696,3.7534854,2.9886484,6
ellipse(4160.1194,3773.6045,7.2594208,3.6636984,38.654858)-ellipse(4161.1194,3773.1045,3.1660604,2.6469
ellipse(4162.3623,3767.058,5.1332518,4.3976302,63.434948)-ellipse(4162.3623,3767.058,3.372299,2.75225,6
ellipse(4166.8967,3881.8016,8.5552512,6.4029784,46.307049)-ellipse(4166.8967,3881.8016,4.2776256,3.2014
ellipse(4169.0361,3898.9062,10.399456,7.8135624,58.489971)-ellipse(4169.0361,3898.9062,5.199728,3.90678
ellipse(4191.2122,3758.3247,10.836415,8.8479128,54.529949)-ellipse(4191.2122,3758.3247,5.4182076,4.4239
ellipse(4213.0702,3963.5088,5.8869509,6.606619,57.640621)-ellipse(4213.0702,3963.5088,6.1489725,4.65715
```

```
ellipse(4223.1034,3752.069,6.9154368,4.6699172,50.309826)-ellipse(4223.1034,3752.069,3.4577184,2
ellipse(4223.1931,3888.9691,5.6670356,5.0726628,66.179474)-ellipse(4223.1931,3888.9691,2.8335178
ellipse(4232.1721,3956.4796,10.024236,11.690018,54.346333)-ellipse(4232.1721,3956.4796,5.0121182
ellipse(4254.2364,3851.1877,9.6145372,6.3099924,59.28215)-ellipse(4254.2364,3851.1877,4.8072686,
ellipse(4326.4177,3651.5823,23.310154,12.462525,55.26606)-ellipse(4326.4177,3651.5823,11.655077,2
ellipse(4336.2644,3887.1149,6.729814,5.0160918,51.259182)-ellipse(4336.2644,3887.1149,3.364907,2
ellipse(4242.04,3682.88,12.448016,3.2562096,49.658543)-ellipse(4242.04,3682.88,3.6650522,2.60603
ellipse(4246.15,3837.9167,5.5514289,6.9530192,45.48172)-ellipse(4246.15,3837.9167,5.4618821,3.87
ellipse(4248.7547,3677.9623,12.576657,5.0829714,51.965843)-ellipse(4248.7547,3677.9623,6.2279682
ellipse(4281.4821,3832.7857,9.8866826,7.9729324,67.028198)-ellipse(4281.4821,3832.7857,7.0371718
```


## 4. Fill in the Holes (dmfilth)

The tool `dmfilth` offers several options for extrapolating over point source regions (see the [ahelp file](#) for more information.) Here, we use the POISSON method, which assigns pixel values to the source region by sampling the Poisson distribution whose mean is that of the pixel values in the background region.

We run `dmfilth` using the region files created in the previous sections:

```
unix% punlearn dmfilth
unix% pset dmfilth infile=diff_image.fits
unix% pset dmfilth outfile=diff_image_filled.fits
unix% pset dmfilth method=POISSON
unix% pset dmfilth srclist=@sources_mod.reg
unix% pset dmfilth bkglist=@bkg_sub.reg
unix% pset dmfilth randseed=123
unix% dmfilth
Input event file (diff_image.fits):
Enter output file name(s) (diff_image_filled.fits):
Interpolation method (POLY|DIST|GLOBAL|POISSON|BILINT) (POISSON):
List of sources to fill in (@sources_mod.reg):
List of background regions (@bkg_sub.reg):
```

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

The output file is shown in [Figure 4](#) . As expected, the point sources are no longer visible.

## Smooth the Image

The "filled-in" image file is smoothed via the tool `aconvolve`. A gaussian is used for the kernel specification (`kernel spec`) and the kernel is normalized by the area (`normkernel`).

For this data, we define the `kernel spec` as `lib:gaus(2,5,1,10,10)`. This means the Gaussian:


- has 2 dimensions;
- is embedded in an array 5 sigma in size;
- is normalized to 1;
- has a sigma of 10 pixels along each axis.

Users will have to experiment with different `kernel spec` definitions to find the optimal one for the dataset.

```
unix% punlearn aconvolve
unix% pset aconvolve infile=diff_image_filled.fits
unix% pset aconvolve outfile=smoothed_fill.fits
unix% pset aconvolve kernel spec="lib:gaus(2,5,1,10,10)"
unix% pset aconvolve method=fft
```

```
unix% aconvolve
Input file name (diff_image_filled.fits):
Kernel specification (lib:gaus(2,5,1,10,10)):
Output file name (smoothed_fill.fits):
```

The contents of the parameter file may be checked using [plist aconvolve](#).

Figure 5  shows the output smoothed image (smoothed\_fill.fits).

## Exposure–Correcting the Image (Optional)

From this point, it is also possible to incorporate an exposure map in order to create an exposure–corrected image. Unless there are significant exposure variations across the field, this will not make a difference in the final image; exposure–correcting the data used in this thread did not have a visible effect on the output.

The steps to include an exposure correction are as follows:

1. Create an exposure map (expmap.fits) for the data by following one of the [Exposure Map threads](#).
2. Use [dmimgcalc](#) to divide the *unsmoothed* image (diff\_image\_filled.fits) by the exposure map:

```
unix% dmimgcalc infile="diff_image_filled.fits" infile2="expmap.fits" \
  outfile="image_div_expmap.fits" operation="div" weight="1" weight2="1"
```

3. Smooth the exposure–corrected image with [aconvolve](#), as in the [Smooth the Image section](#).

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

```
#
# parameter file for wavdetect
#
# input
#
#   infile = diff_image.fits  Input file name
#
# output
#
#   outfile = sources.fits      Output source list file name
#   scellfile = sources_scell.fits Output source cell image file name
#   imagefile = sources_image.fits Output reconstructed image file name
#   defnbkgfile = sources_bkg.fits Output normalized background file name
#
# scales
#
#   scales = 1 2 4              wavelet scales (pixels)
#   (regfile = sources.reg)     ASCII regions output file
#
# output options
#
#   (clobber = no)             Overwrite existing outputs?
#   (kernel = default)         Output file format (fits|iraf|default)
#   (ellsigma = 4)             Size of output source ellipses (in sigmas)
#   (interdir = .)             Directory for intermediate outputs
```



## Create an Image of Diffuse Emission – CIAO 3.4

```
#
#####
#
#   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
#       (bkgthresh = 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) Tab
#
#   end of wrecon parameters
#
#####
#
#   run log verbosity and content
#
#       (log = no)              Make a log file?
#       (verbose = 0)           Log verbosity
#
#   mode
#
#       (mode = ql)
```

## Create an Image of Diffuse Emission – CIAO 3.4

Parameters for /home/username/cxods\_param/dmfilth.par

```
##
## DMFILTH -- fill in the hole
##
    infile = diff_image.fits  Input image file
    outfile = diff_image_filled.fits Enter output file name(s)
    method = POISSON          Interpolation method
    srclist = @sources_mod.reg List of sources to fill in
    bkglist = @bkg_sub.reg    List of background regions
    (randseed = 123)          Seed for random number generator
    (clobber = no)            OK to overwrite existing output file(s)?
    (verbose = 0)             Verbosity level
    (mode = ql)
```

---

Parameters for /home/username/cxods\_param/aconvolve.par

```
#
# aconvolve.par file
#
    infile = diff_image_filled.fits Input file name
    outfile = smoothed_fill.fits Output file name
    kernelspec = lib:gaus(2,5,1,10,10) Kernel specification
#
# auxillary outputs
#
    (writekernel = no)        Output kernel
    (kernelfile = ./.)       Output kernel file name
    (writefft = no)          Write fft outputs
    (fftroot = ./.)         Root name for FFT files
#
# processing parameters
#
    (method = fft)           Convolution method
    (edges = wrap)           Edge treatment
    (const = 0)              Constant value to use at edges with edges=constant
    (pad = no)               Pad data axes to next power of 2^n
    (center = no)            Center FFT output
    (normkernel = area)      Normalize the kernel
#
# user specific comments
#
    (clobber = no)           Clobber existing output
    (verbose = 0)            Debug level
    (kernel = default)       Output format kernel
    (mode = ql)
```

---

## History

16 Dec 2004 updated for CIAO 3.2: minor changes to `csmooth` parameter file

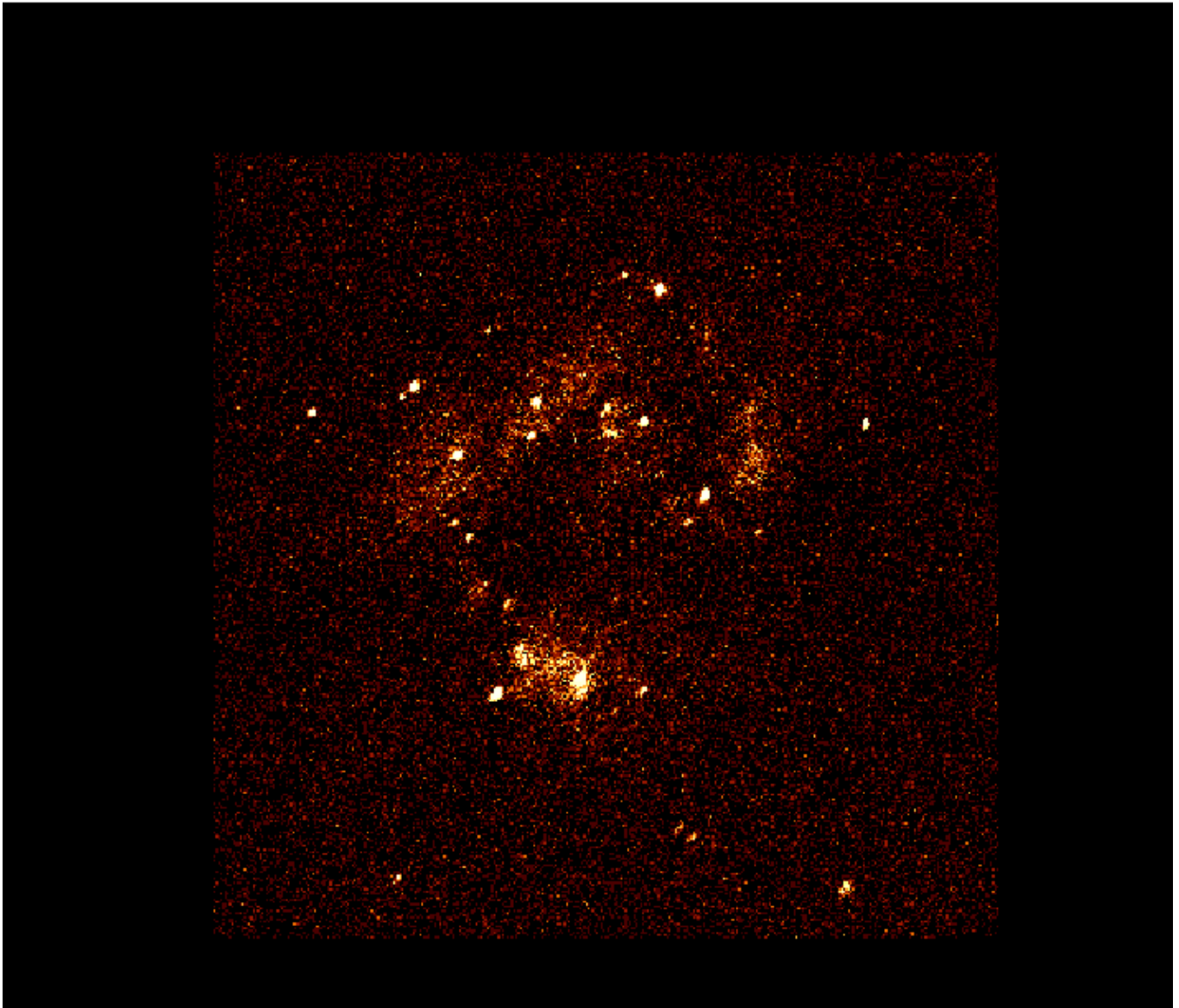
21 Dec 2005 reviewed for CIAO 3.3: no changes

01 Dec 2006 updated for CIAO 3.4: uses `aconvolve` instead of `csmooth` for smoothing, updated image to match

URL: [http://cxc.harvard.edu/ciao/threads/diffuse\\_emission/](http://cxc.harvard.edu/ciao/threads/diffuse_emission/)

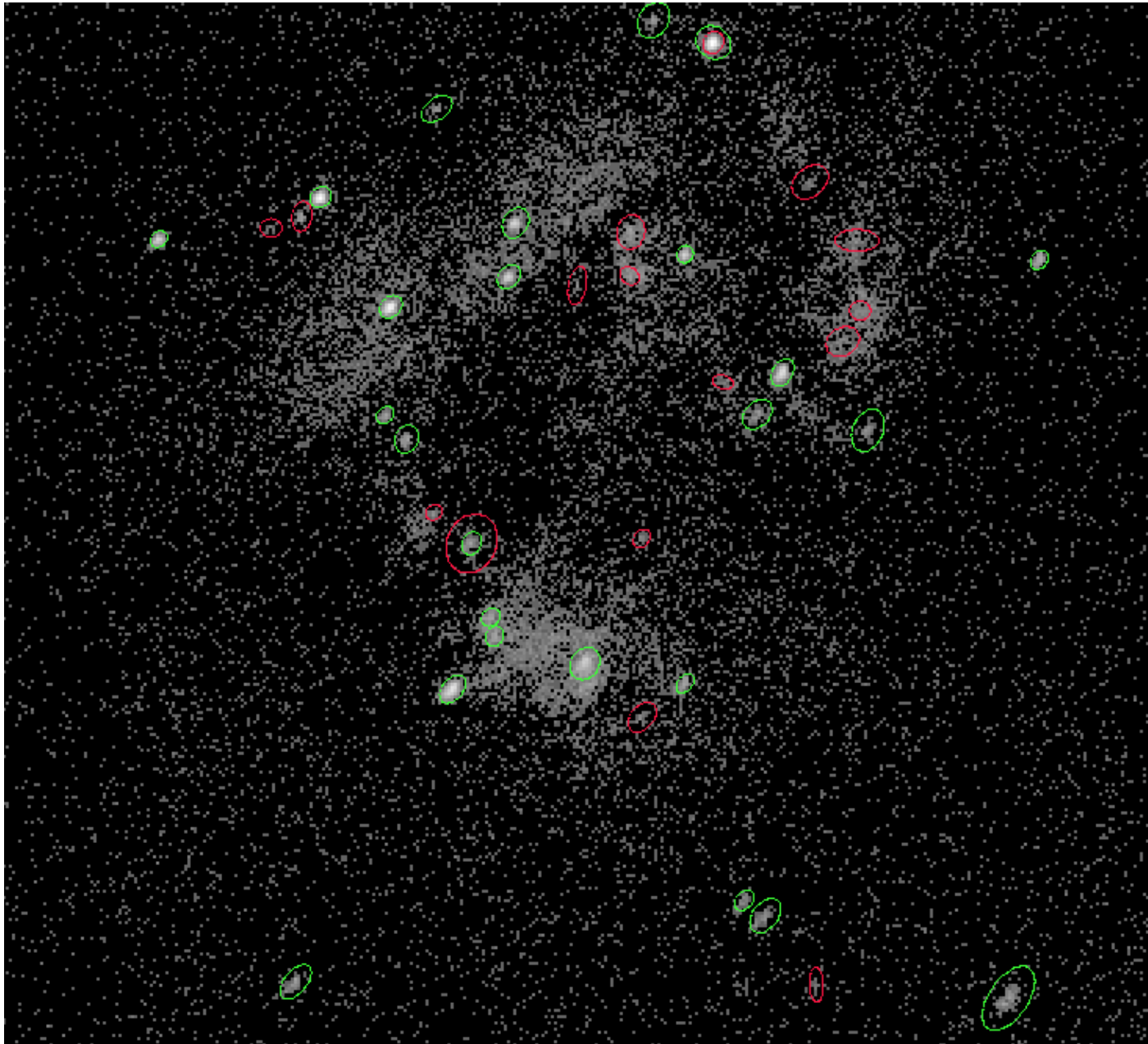
Last modified: 1 Dec 2006

**Image 1: Diffuse emission plus point sources**

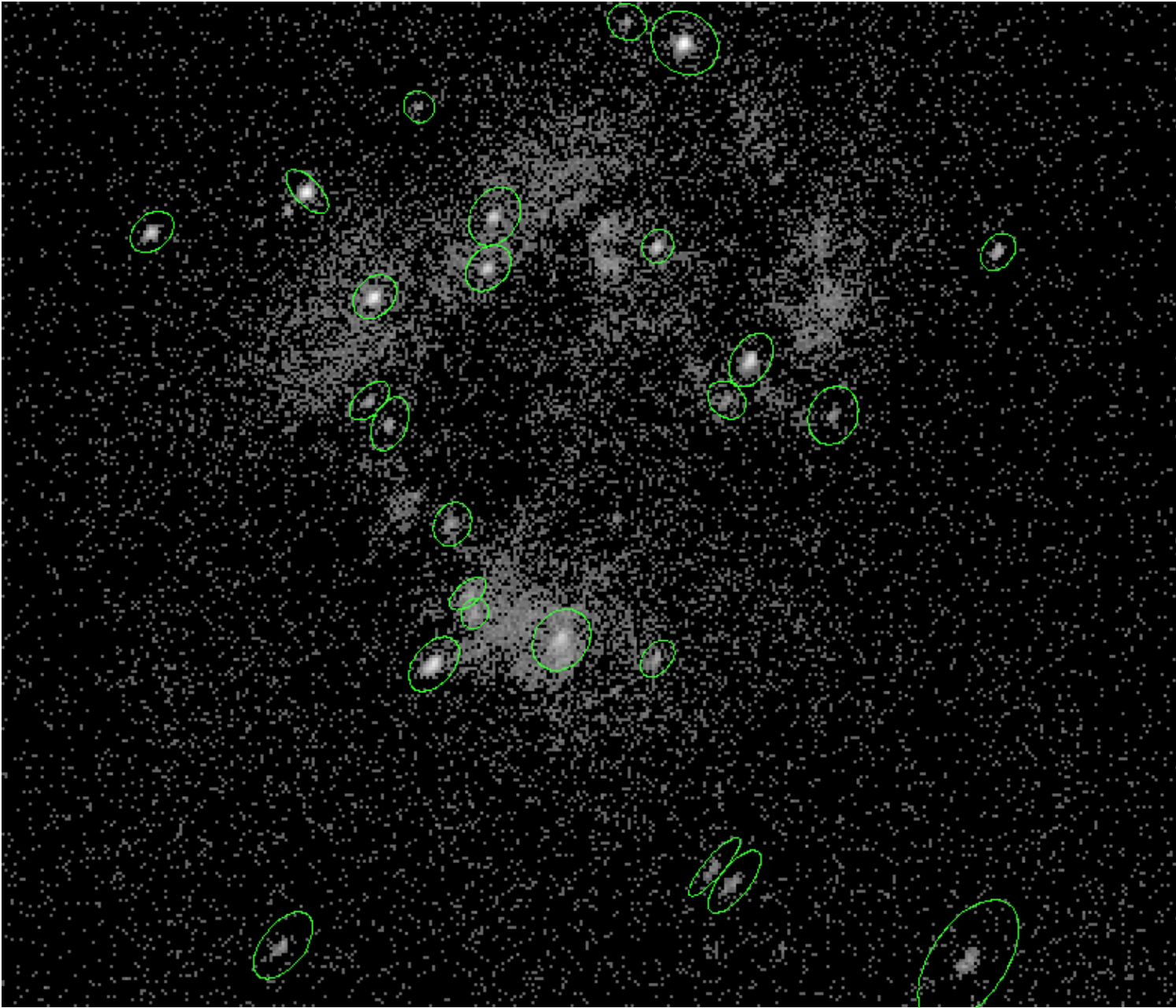


## Image 2: Detected point sources

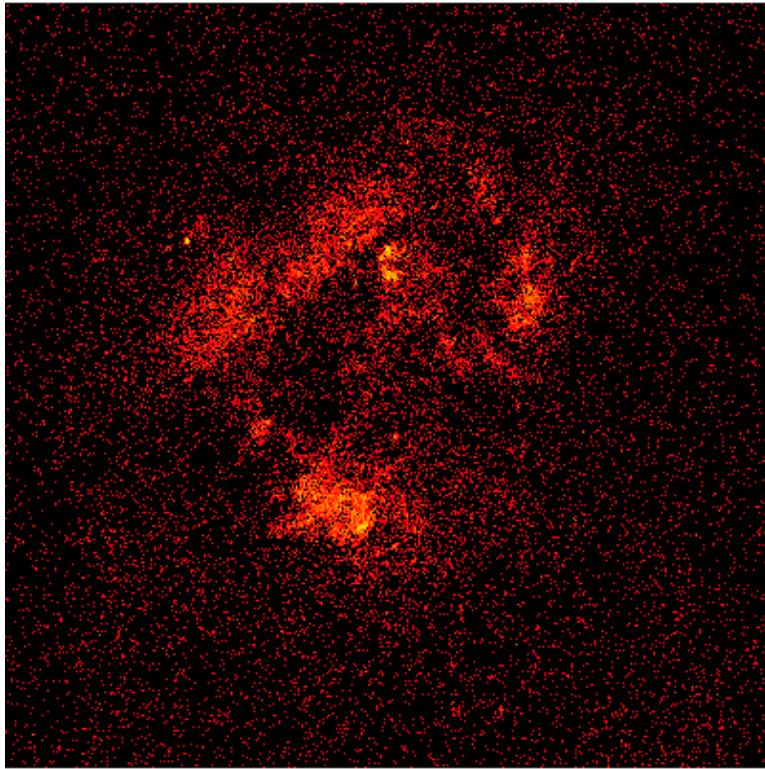
The original wavdetect sourcelist is in red; the modified list is in green.



### Image 3: Modified background regions



**Image 4: Diffuse emission only**



**Image 5: Smoothed image of the diffuse emission**

