DETECT Talk for CIAO workshop
D. Harris



## 1.0 GENERAL - Philosophy....what it is, and what it isn't.

VLA radio maps are pleasing to work with. The synthesized beam is reasonably constant (until delay beam smearing kicks in) and the noise level is sensibly constant so long as there are not serious phase errors remaining in the data. When you want to list all sources detected at 5 sigma or above, it is a fairly simple task to obtain reliable results. For photon starved X-ray data however, statistical fluctuations can pass as sources, the PSF varies across the field, and estimating the background at a given location can be difficult. Visual inspection can easily identify an unresolved weak source on axis, but the same source off-axis may well be lost in the background.

CIAO has three imperfect tools to find candidate source locations. Each has strengths and weaknesses, and my advice is to use all three and compare the results if you really want to get a reliable list of candidates.

The critical functions of detect algorithms are to determine the background (i.e. what would have been there without the source) and to isolate source counts and regions for the candidate list. Each of the tools in CIAO performs these operations differently, and thus they will provide different results for any particular observation.

Detect algorithms serve many purposes, and user's perceptions vary widely. My own view is that the primary purpose of the tools is to locate positions for candidate sources. The fact that the tools also provide various quantitative values for properties of these candidates is useful, but not critical. Source properties should be evaluated with other tools. For example, on-axis, the local maximum will probably be quasi-identical to the centroid of the distribution, but for extended sources or sources far from the field center, this will not generally be true.

An extreme opinion is that the only unique role of detect is for investigators who are doing statistics of sources and for these people, it is vital to understand the accuracy of the process. For celldetect, Dobrzycki has written a chapter in the DETECT manual which deals with this question. Almost all approaches to the statistical properties of detection methods rely on simulations so that one 'knows the truth' a priori.

The DETECT manual is somewhat out of date. It is being revised, but in the meantime, the helpfiles should be considered as the primary source of information whenever there is a conflict between manual and helpfile. The theory chapters in the manual give conceptual details for each method.

## 2.0 CELLDETECT

Most users are familiar with the 'sliding cell' which locates positions for which the counts in the (cell - background) exceeds some estimate of normal fluctuations in the background (parameter 'thresh'). With normal usage, adjacent candidates are put together and a centroid is found for the distribution.

The default setting of the parameters are designed to maximize detection of point sources, and this limitation should not be forgotten.

## 2.1 Background

There are 3 methods available:

a background map, provided by the user (similar to ROSAT 'mdetect')

a local background which uses a frame around the detect cell ('ldetect')

a fixed (numerical) value.

If a reliable background map is available, this is the preferred method since it avoids the possibility that the local frame might run into a neighboring source or that the brightness gradient of an extended source might be so small that the s/n never exceeds the threshold.

However, most often, the frame is employed for background.

## 2.2 Cellsize

If a fixed cellsize is not chosen and a PSF library is available, celldetect automatically adjusts the size of the cell to match the PSF size off-axis. The user has control of this via the parameters 'eenergy', and 'eband'. Since the PSF library has been revised, running celldetect with default parameters will give different results now than a year ago.

1 VG
22cellsiz1.ps

## 2.3 Recursive blocking

In order to deal with large data spaces (defined as TLMIN/TLMAX in the fits header) if the space is greater than 2048 pixels, the inner 2048 is examined at block=1 (i.e. native pixel size), and then the whole image is blocked by 2, and the inner 2048 is again examined except for the inner quadrant that has already been dealt with. This process continues until the entire data space is covered.

2 VGs
23cellsiz2.ps
23cellsiz4.ps

## 2.4 Region files

celldetect (and the others) provide an option to output region files as a convenience to over-plot detections on images. The size of the ellipse is determined by a multiplicative factor (ellsigma) and the measured standard deviation of the distribution. An equivalent region file can be produced from the standard fits output file.

3VGs
24295cell1over.ps
2
3

## 2.5 Exposure maps

This option is provided to curtail the false detections which often occur at the edges of chips or fields. The user must generate an exposure map and control is given via the parameter 'expratio', the ratio of average exposure in the background frame to that in the detect cell. Note however that all detections will be reported in the standard fits output; the rejected detections will not appear in the region file.

VG 25cell_expmap.ps

2.6 The PSF library

VG
26cellpar.ps

Reviewing the parameter file for celldetect, besides the s/n threshold parameter ('thresh') which determines how deep celldetect will go, we note the parameters which control the cellsize: eband, eenergy, and psftable. The idea is that as the PSF size increases (as described by the PSF library file), so too will the cellsize increase. The user can choose what fraction, eenergy (for 'encircled energy') of the total number of counts (of an unresolved source) should fall within the detect cell, and also at what energy this is specified. The PSF library gives eenergy as a function of radius for many energies and off-axis angles, so celldetect bases the cell size on that information. From time to time, the PSF library is updated with better estimates and this affects the operation of celldetect.

The viewgraph shows a section of an HRC observation of 3C273. The old PSF library did not cover the outer parts of the field, but even in the areas of common coverage, only the stronger features were detected with both PSF libraries.

VG 26diffPSFcell.ps

3.0 WAVDETECT

Our implementation of wave detect, like most others, uses a 'Mexican hat' function which contains equal positive and negative contributions. This function is convolved with the image to locate candidate sources.

There are two parts to wavdetect which can be run independently by the knowledgeable user to obtain more versatility. The first part (wtransform) produces the correlation maps at each scale and generates lists of candidate positions. The second part (wrecon) uses these outputs to define a cell containing most of the source counts and then obtains relevant source parameters.

3.1 Background

This is generated by the tool. Locations of high correlation (image/wavelet) are removed and the process is repeated until few candidates are found. Then the residual provides a measure of the background. In the viewgraph, there was not a scale large enough to remove the cluster; hence it is 'detected' as a background feature.

VG - 31wav_bkg.ps

It is also possible for the user to supply a map.

3.2 Scales

The whole detection process is repeated (independently) for a number of user-specified scales which are normally separated by factors of 2 or SQRT(2). While it is necessary to include the proper scale sizes of anything one needs to detect, both a large number of scales and large sized scales make heavy demands on machine memory and run time. The reconstructed image with 5 scales is shown for 3C 295: the large scales are recovered only for extended emission or strong point sources.

VG - 32wav_recon.ps

## 3.3 Source properties

The second part of the tool obtains source properties by measuring the counts within a cell that has most of the counts. The scell output map shows the area considered for each detection and each region has a value equal to the source number from the standard output file.

2 VGs
33wav.ps
33wav_snum.ps

## 3.4 The PSF library

VG - 34wavpar.ps

Reviewing the wavdetect parameter file, in addition to the key parameters of 'scales' and 'sigthresh', we see again the psflibrary file and its controlling parameters, eenergy and eband. Unlike celldetect however, the detection process of wavdetect is unaffected by the PSF size. In wavdetect, it is only the choice of the flux scale which depends on the size of the PSF so the differences spawned by a change in PSF library alters not which sources are detected, but rather the locations and sizes.

2 VGs
34wavPSF1.ps
34wavPSF2.ps

# 4.0 VTPDETECT

Unlike the previous two, vtpdetect is scale free. The process judges source location based on the density of events. Each event is assigned an area based on the density of events at that location. A detection is a collection of small areas near each other; background events have larger areas associated with them.

3 VGs
40vtpcells.eps
40vtpfield.ps
40vtpfieldcen.ps

There is a tendency for VTP to join sources together. This operation can be controlled by the threshold scaling parameter. Setting 'scale' to values greater than one will force vtpdetect to use a threshold higher than that determined automatically. The useful range is 0.8 to 3.0.

VG
40vtppar.ps

# 5.0 EVALUATION OF ALGORITHMS VIA DEEP FIELDS

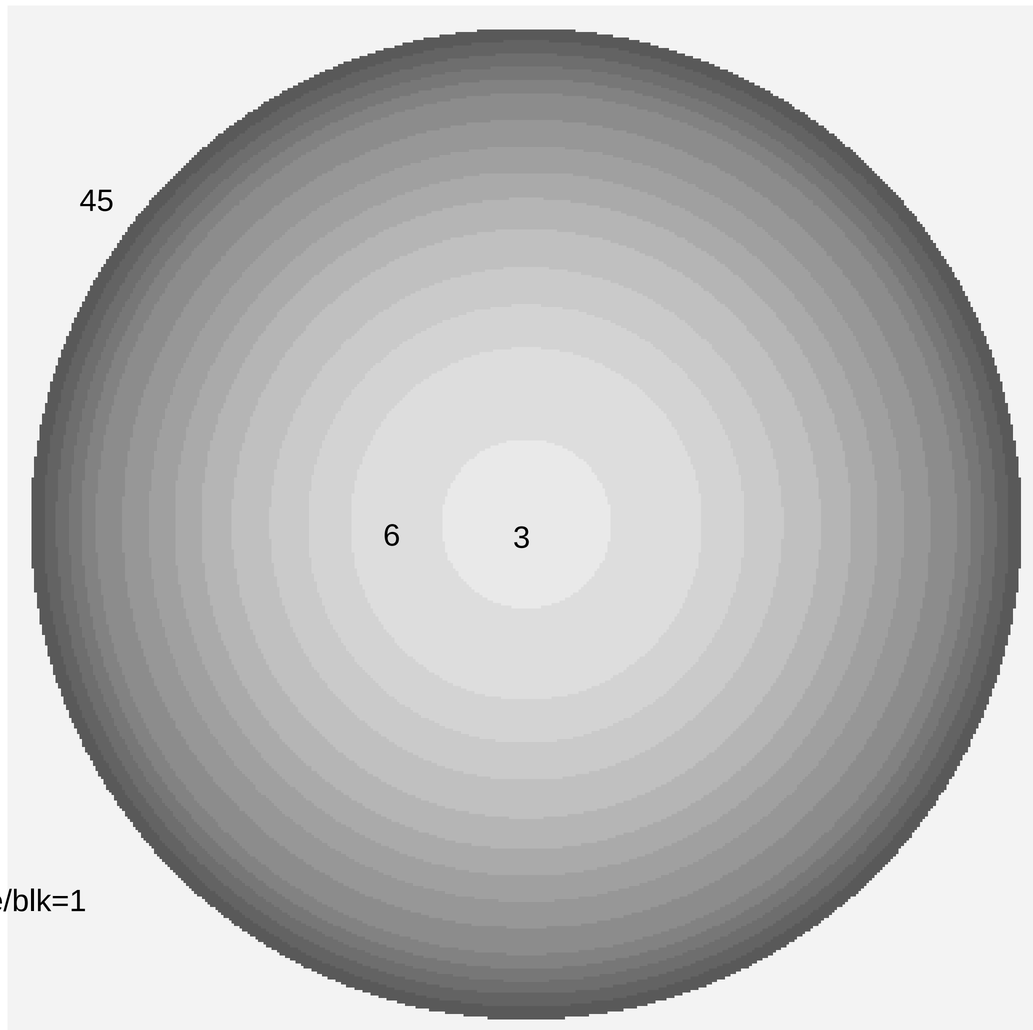
Simulations are normally used to 'test' detect algorithms but an alternative approach is afforded by the use of Chandra deep fields. We run detect on the full data with a conservative s/n threshold to identify high confidence detections and then take roughly 1/10 of the total exposure and run detect again to judge the success rate.

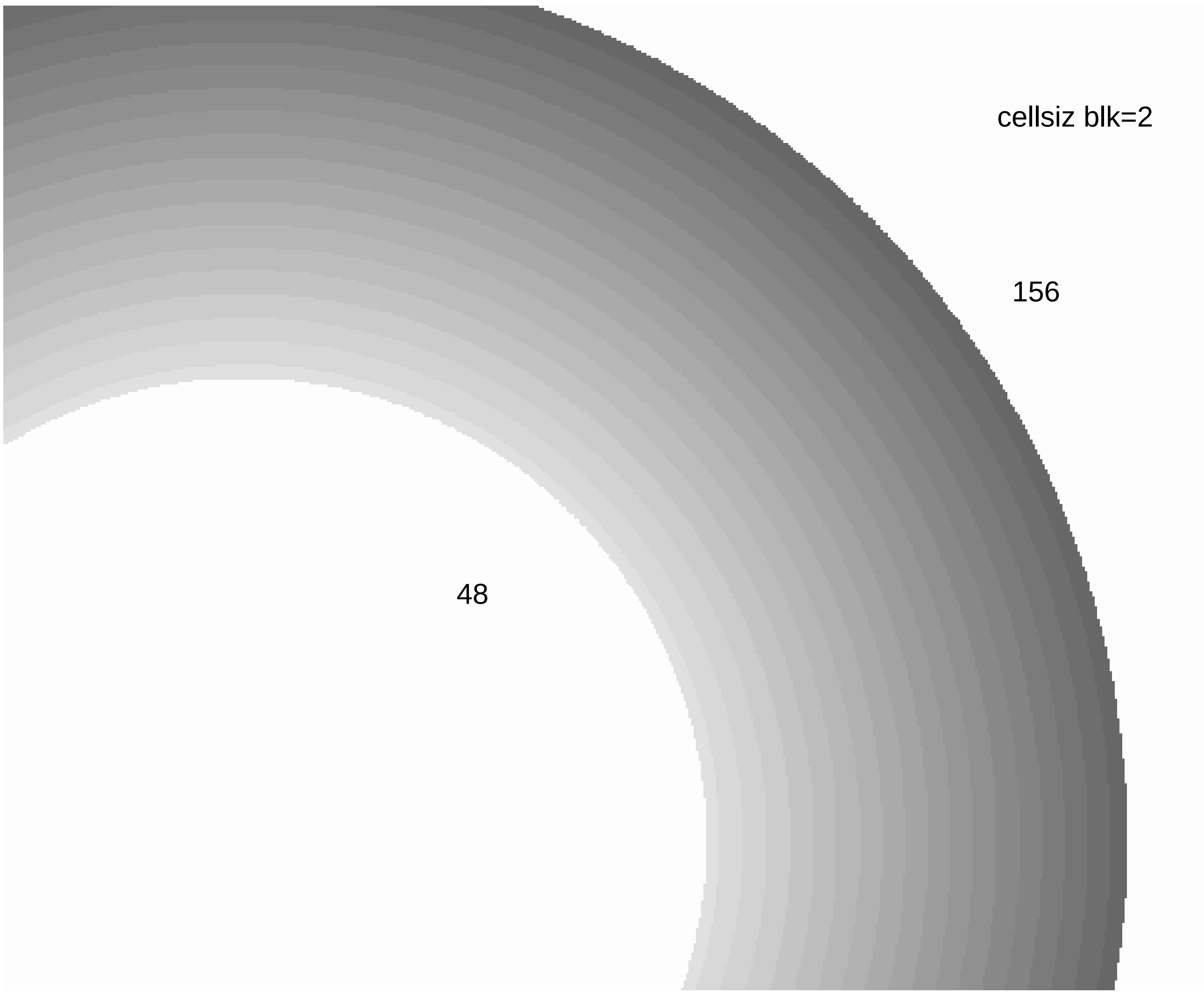
At this time we have only runs with celldetect which demonstrate that at s/n=3, the source list contains only valid sources whereas at s/n=2.5, a number of false detections occur.

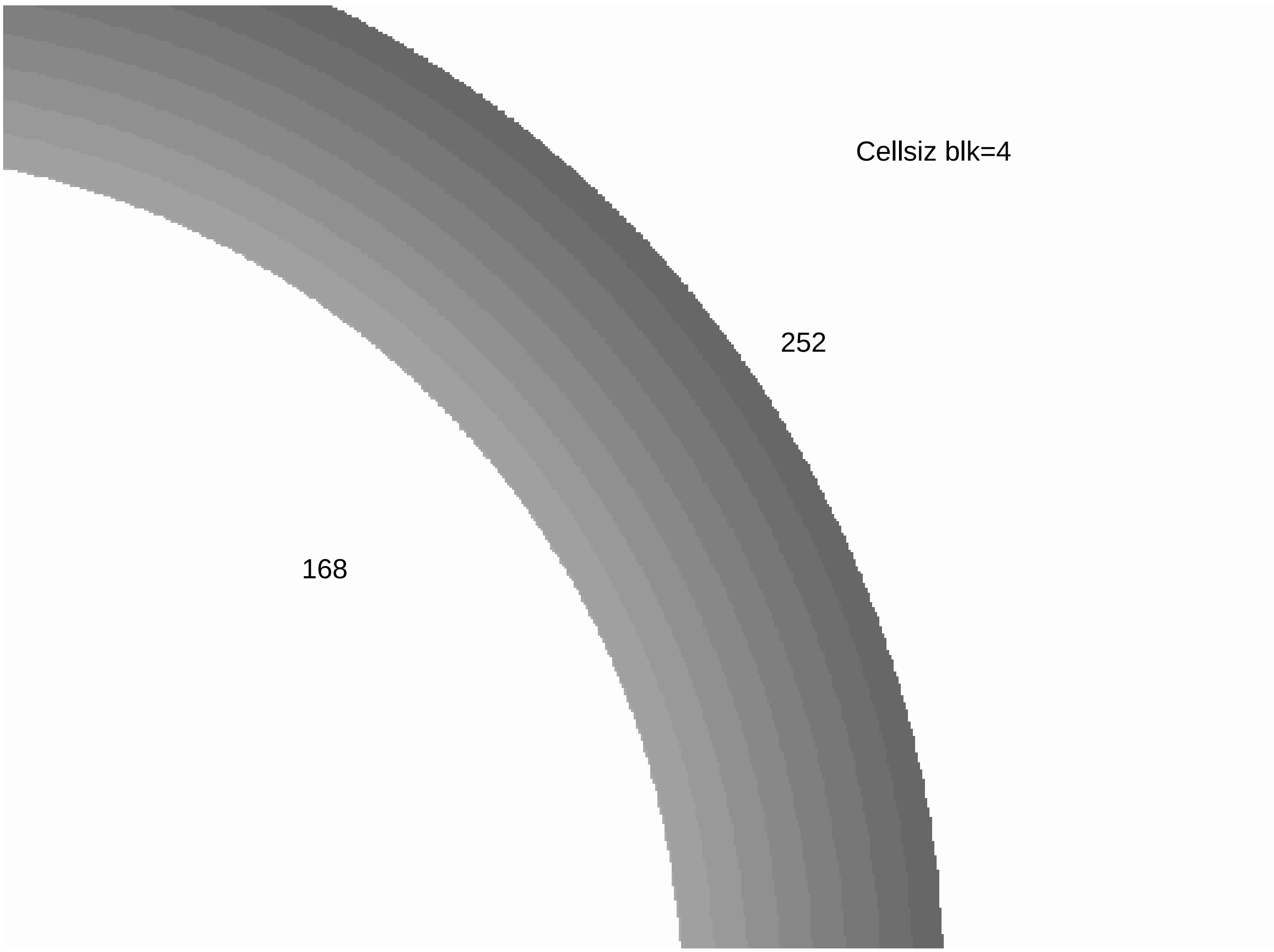
5 VG
50celltot.ps
50celltot1.ps
50cell3.ps
50cell2pt5.ps
50cell2pt5B.ps

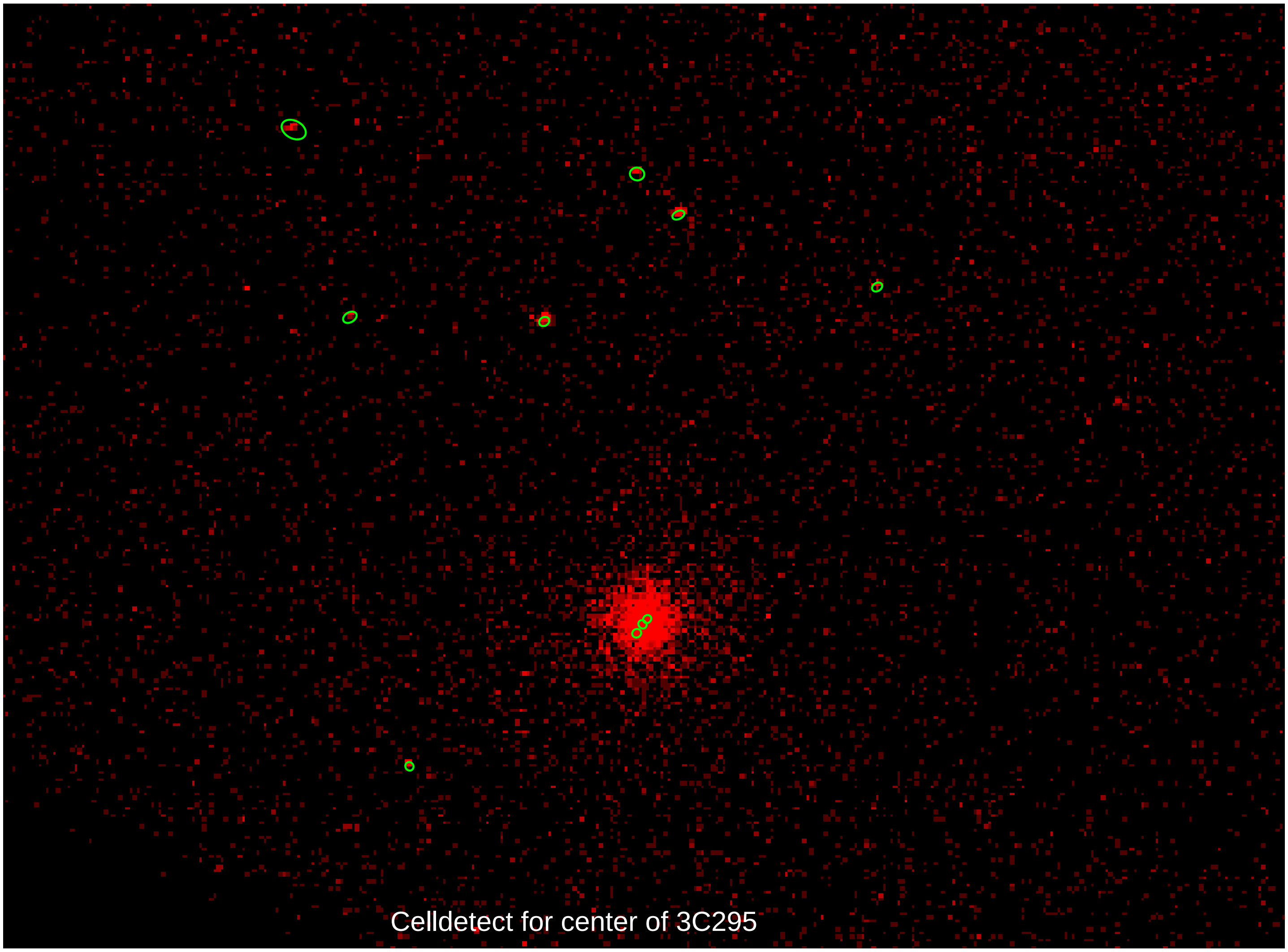
## 6.0 SUMMARY

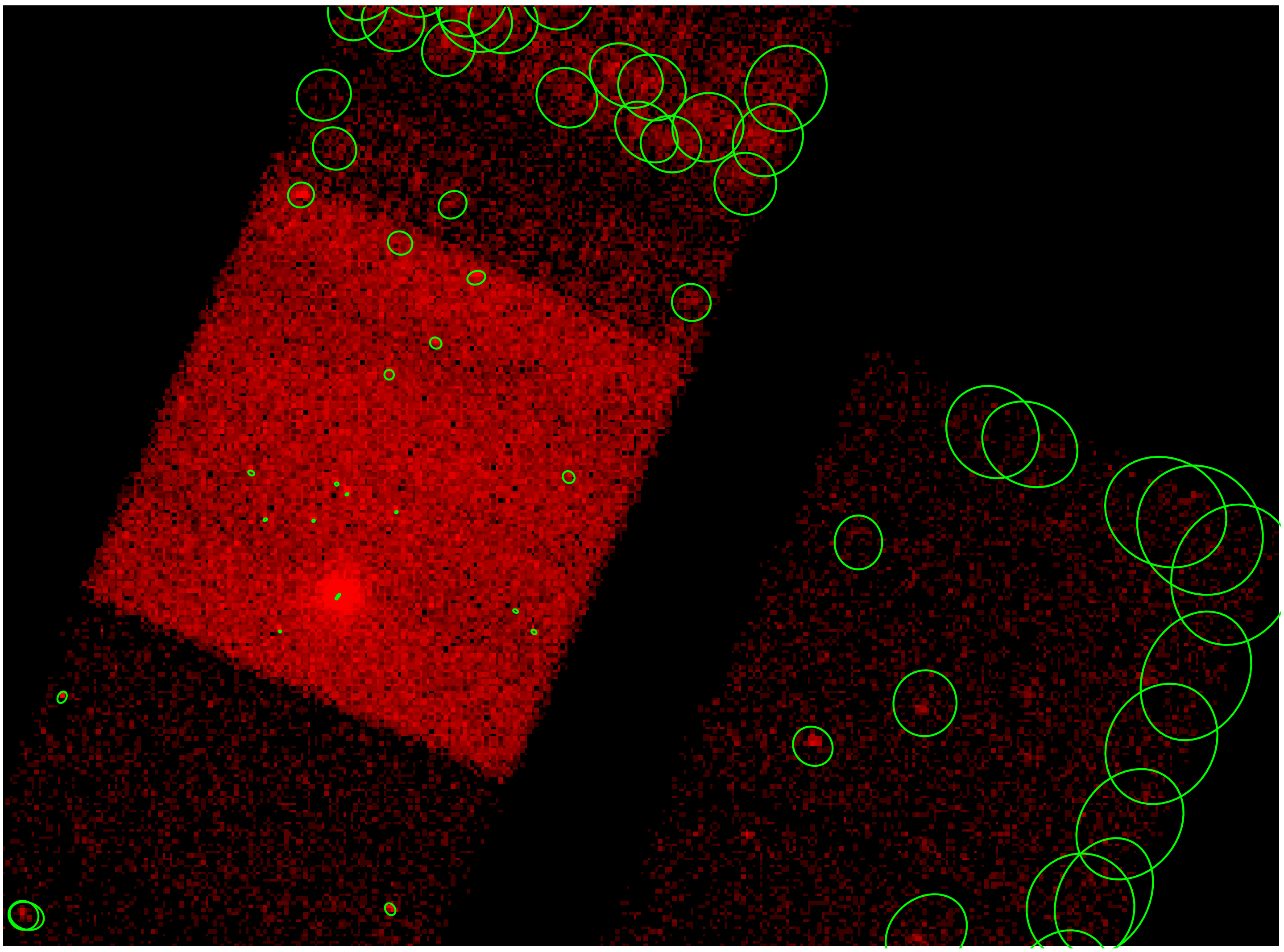
Each tool has specific advantages and each tool can be used differently by adjustment of its parameters.  Some degree of experimentation and testing are warranted, and remember to start with a reasonably sized array before trying more demanding data areas.

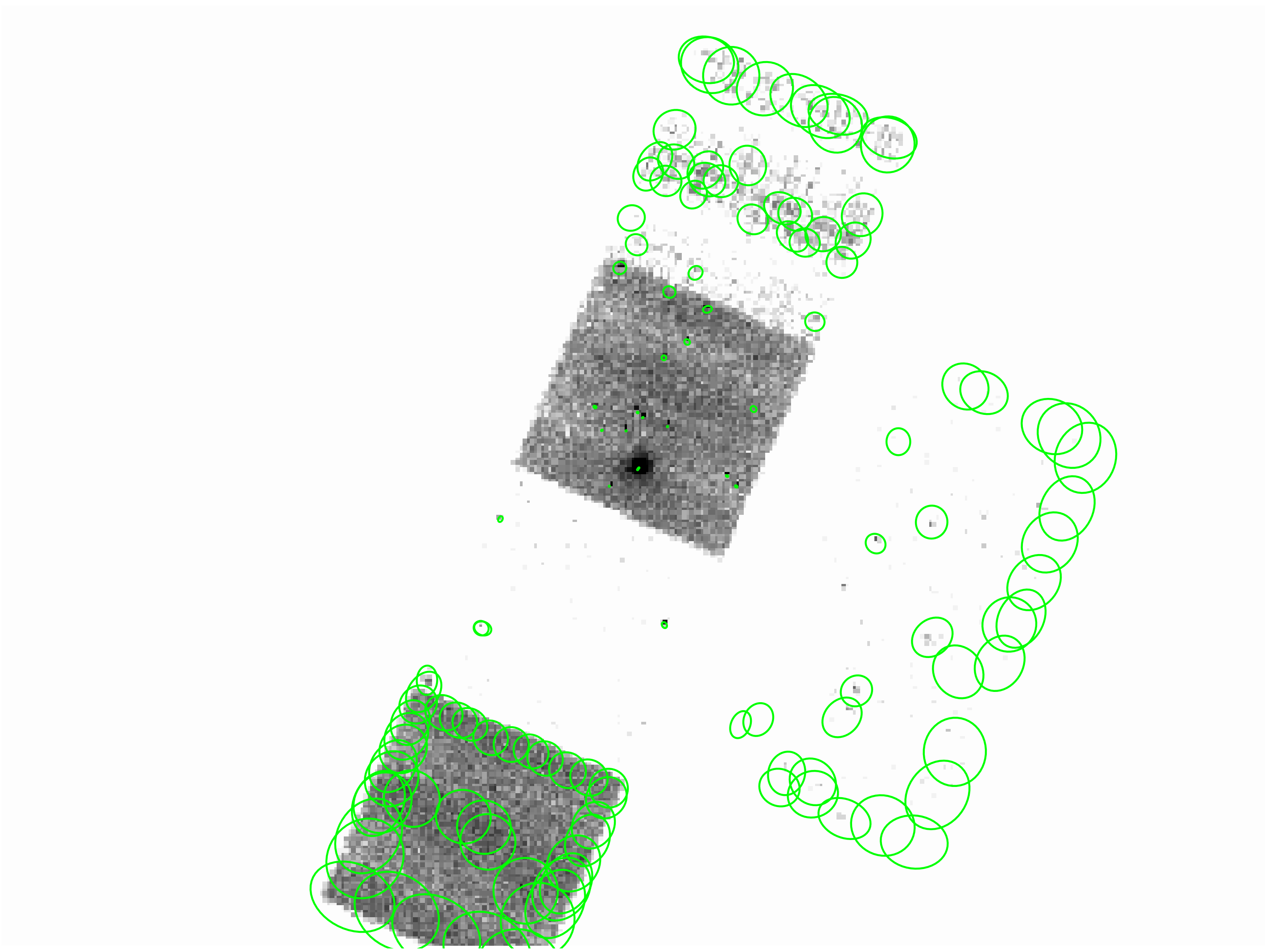

45
6
3
Cellsize/blk=1


cellsiz blk=2
156
48

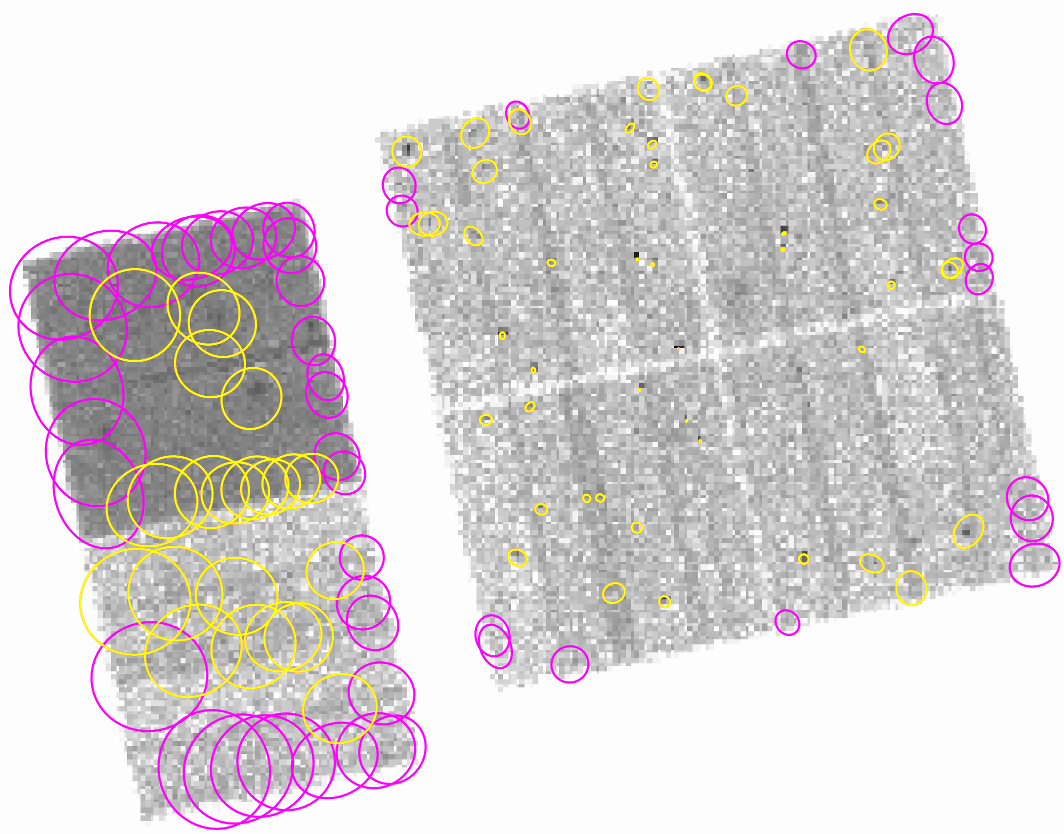

Cellsiz blk=4
252
168


Celldetect for center of 3C295

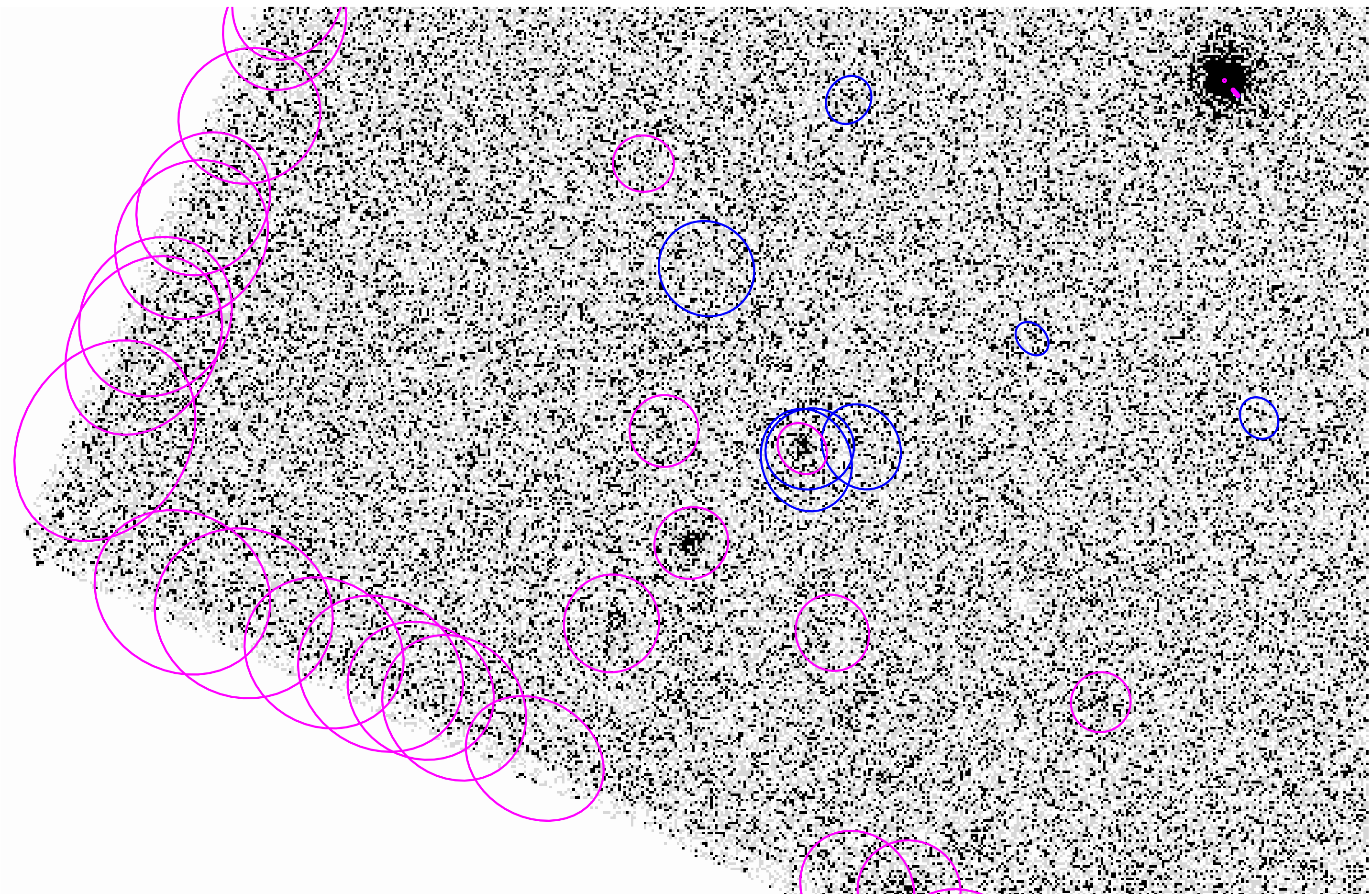


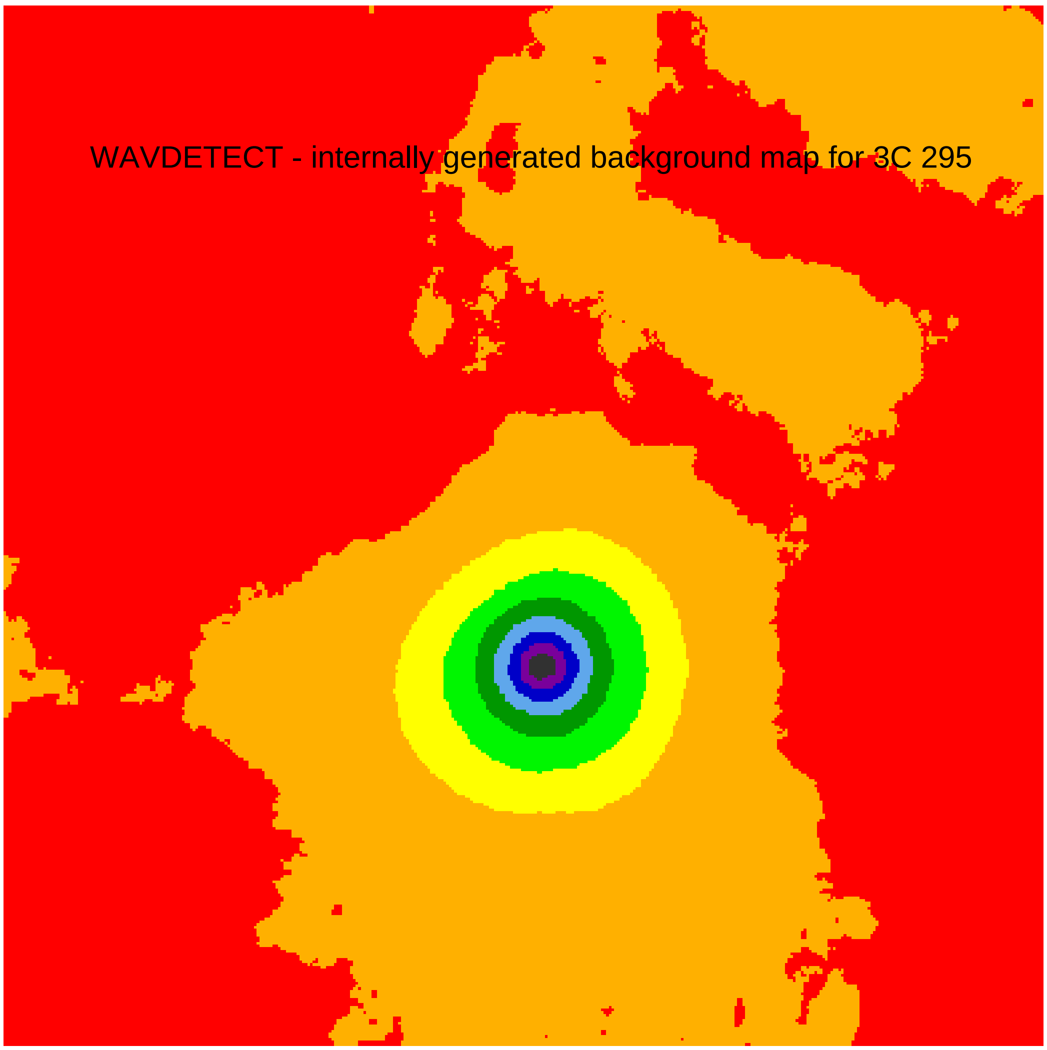


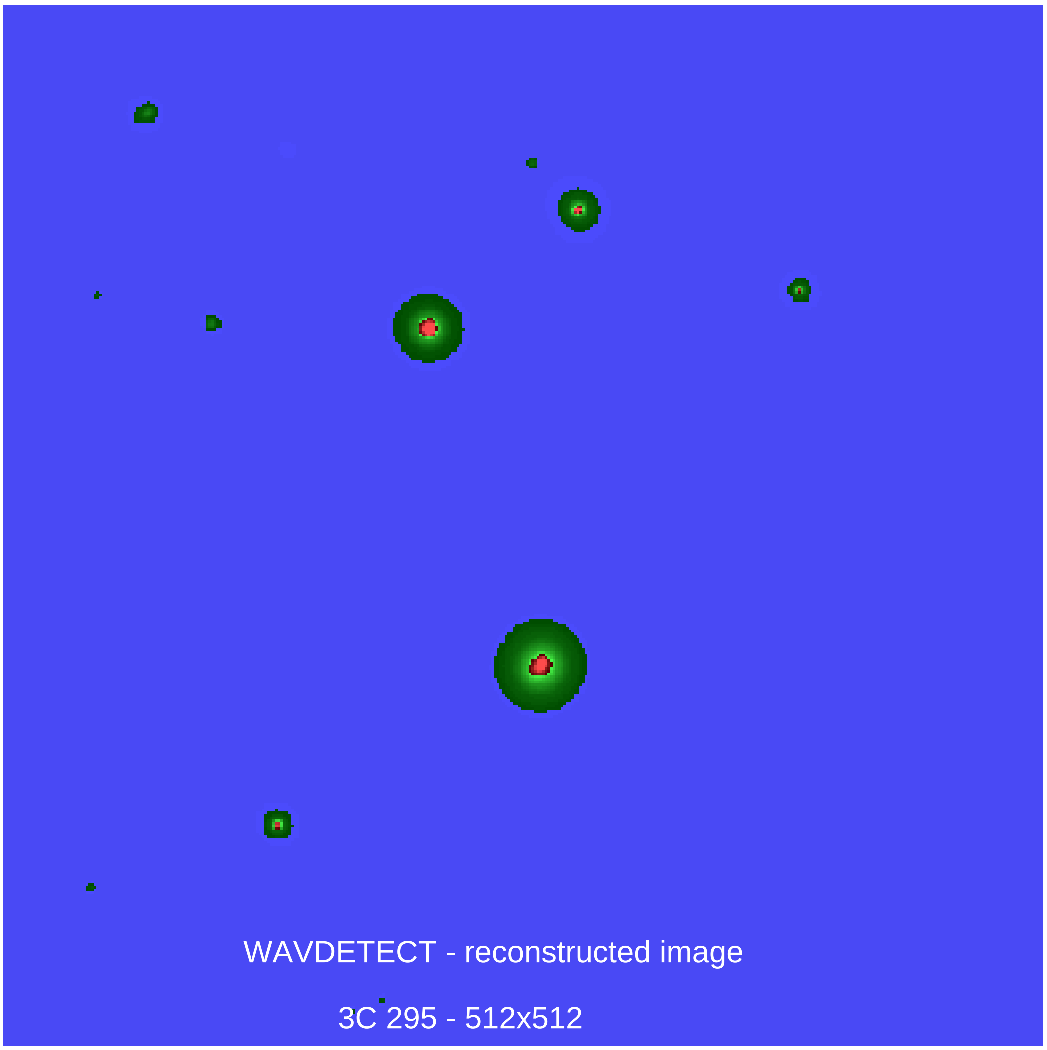
Yellow marks detections with expratio > 0.9; magenta, those rejected.



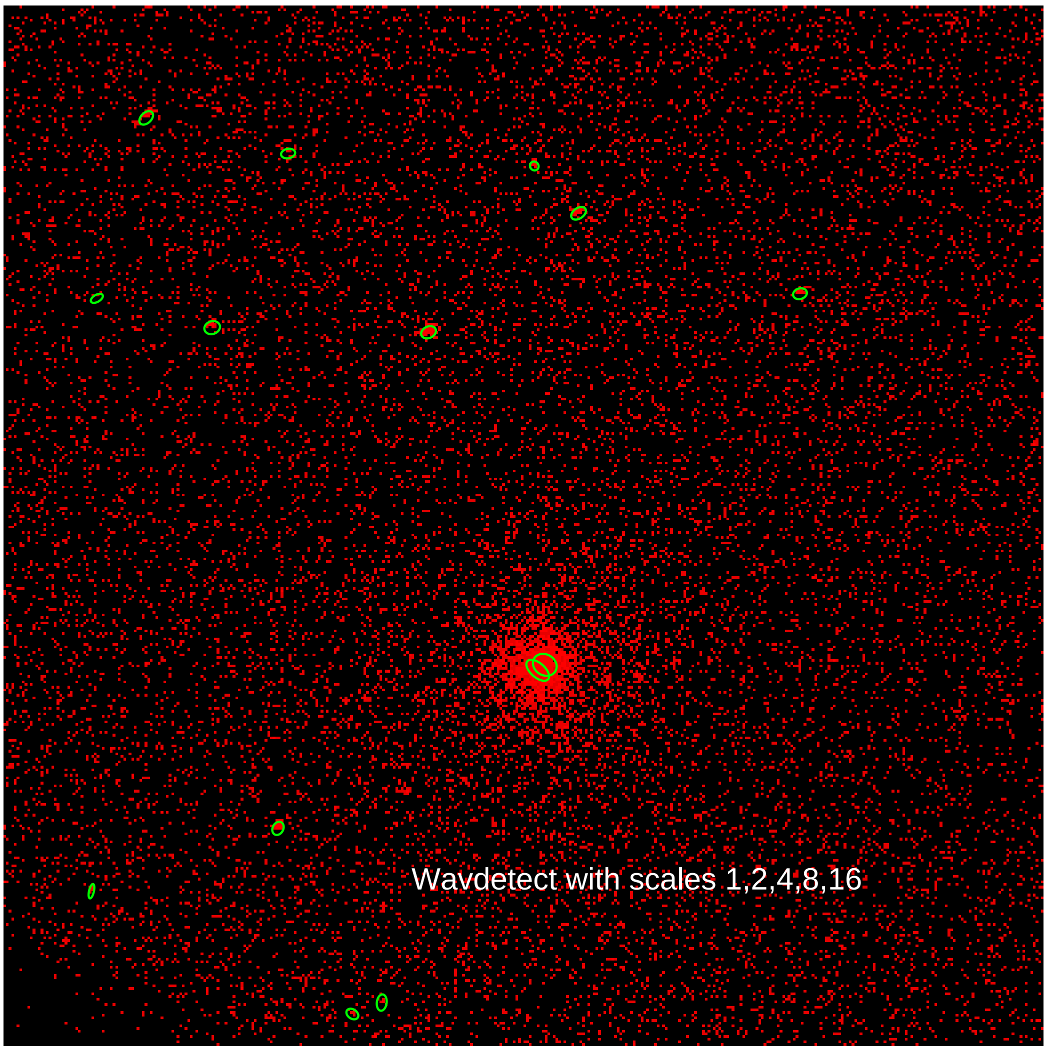
```
Parameters for /home/harris/cxcds_param/celldetect.par

#
#   celldetect parameter file
#
#
#   input
#
        infile = ../hrcm00461N000_evt2.fits Input file
#
#   output
#
       outfile = cell_stdpsf_out.fits Output source list
       (expstk = )                list of exposure map files
      (regfile = cell_stdpsf.reg) ASCII regions file
#
#   output options
#
       (kernel = default)         Output file format
      (clobber = yes)             Overwrite exiting outputs?
#
#   output content/format options
#
       (thresh = 3)               Source threshold
    (findpeaks = yes)             Find local peaks?
     (centroid = yes)             Compute source centroids?
     (ellsigma = 5)               Size of output source ellipses (in sigmas)
     (expratio = 0)               cutoff ratio for source cell exposure variation
#
#   detect cell size parameters
#
    (fixedcell = 0)               Fixed cell size to use (0 for variable cell)
      (xoffset = INDEF)           Offset of x axis from data center
      (yoffset = INDEF)           Offset of y axis from data center
        (eband = 1.4967)          Energy band
      (eenergy = 0.8)             Encircled energy of PSF
     (psftable = ))echo $ASCDS_CALIB/psfsize_20000830.fits -> /proj/cm/installs/cm.install.A
pr17/data/psfsize_20000830.fits) Table of PSF size data
     (cellfile = cell_stdpsf_cellsiz.fits) Output cell size image stack name
#
#   background parameters
#
      (bkgfile = )                Background file name
     (bkgvalue = 0)               Background count/pixel
  (bkgerrvalue = 0)               Background error
#
#   using defaults is recommended here
#
     (convolve = no)              Use convolution?
      (snrfile = )                SNR output file name (for convolution only)
#
#   run log verbosity and content
#
      (verbose = 0)               Log verbosity level
          (log = no)              Make a celldetect.log file?
#
#   mode
#
         (mode = ql)
```

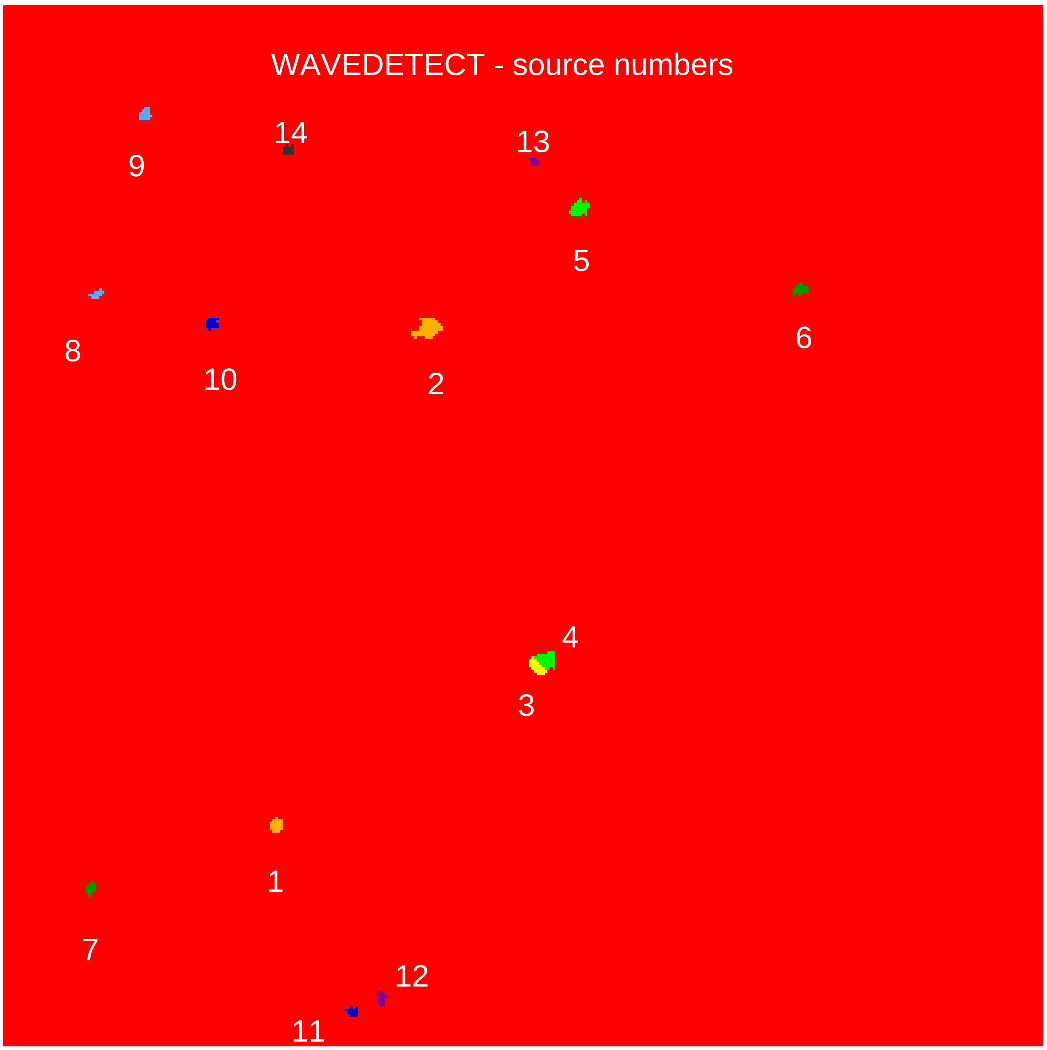

CELLDETECT on HRC: 2 different PSF files
blue: Aug2000 vs.  magenta: revised PSF


WAVDETECT - internally generated background map for 3C 295



WAVDETECT - reconstructed image
3C 295 - 512x512


Wavdetect with scales 1,2,4,8,16

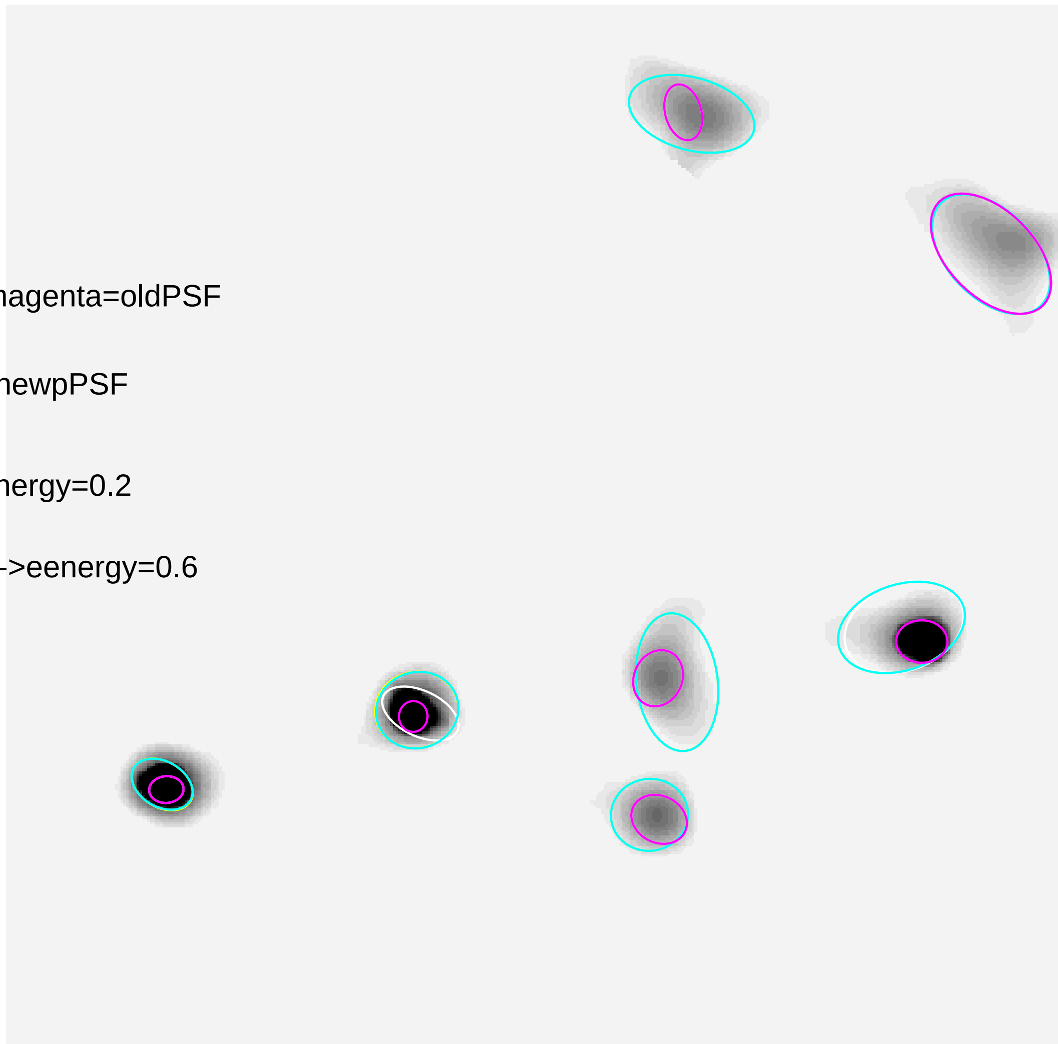

WAVEDETECT - source numbers
14
13
9
5
6
8
10
2
4
3
1
7
12
11

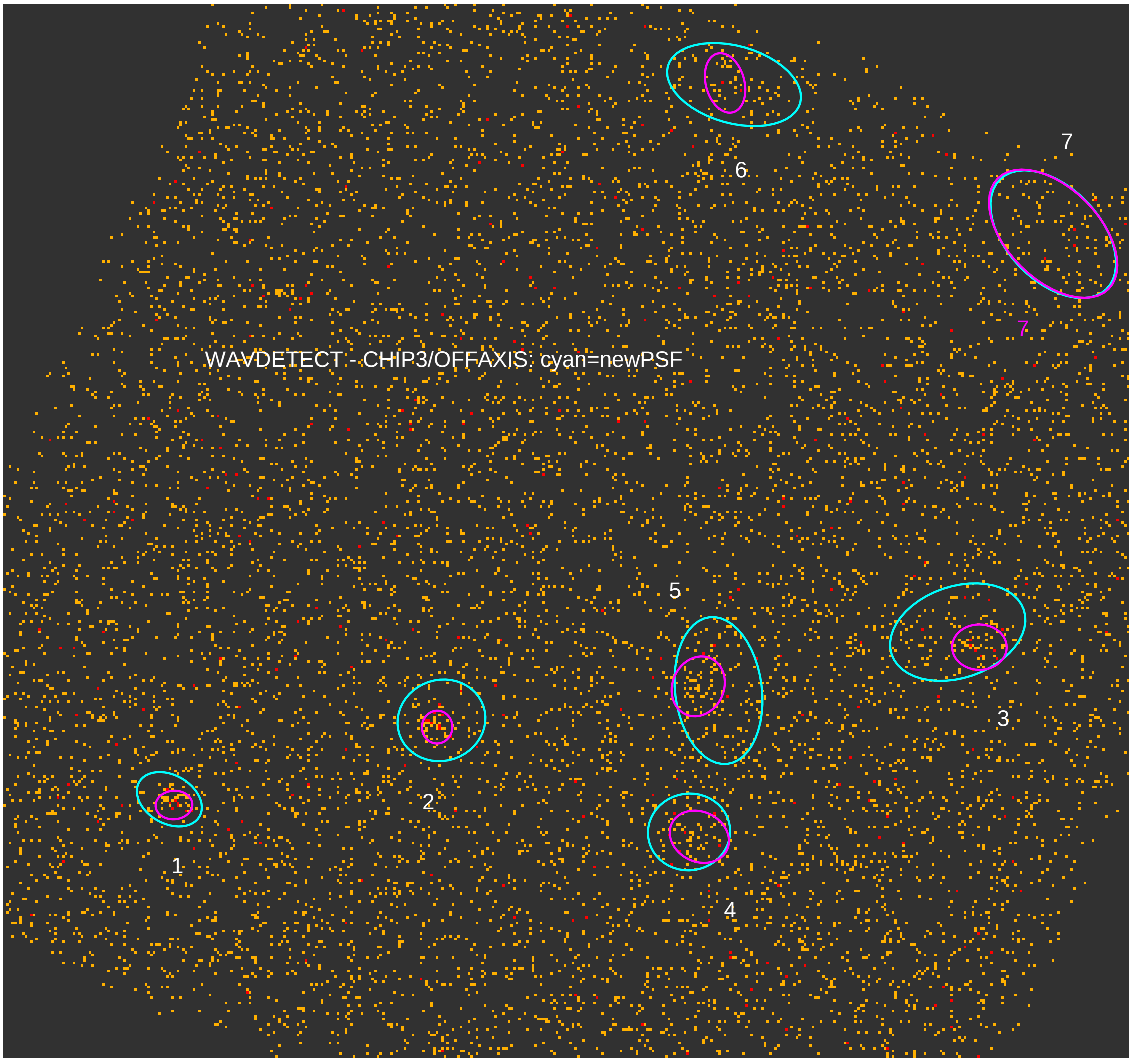
Chip3: magenta=oldPSF

cyan=newpPSF

white->eenergy=0.2

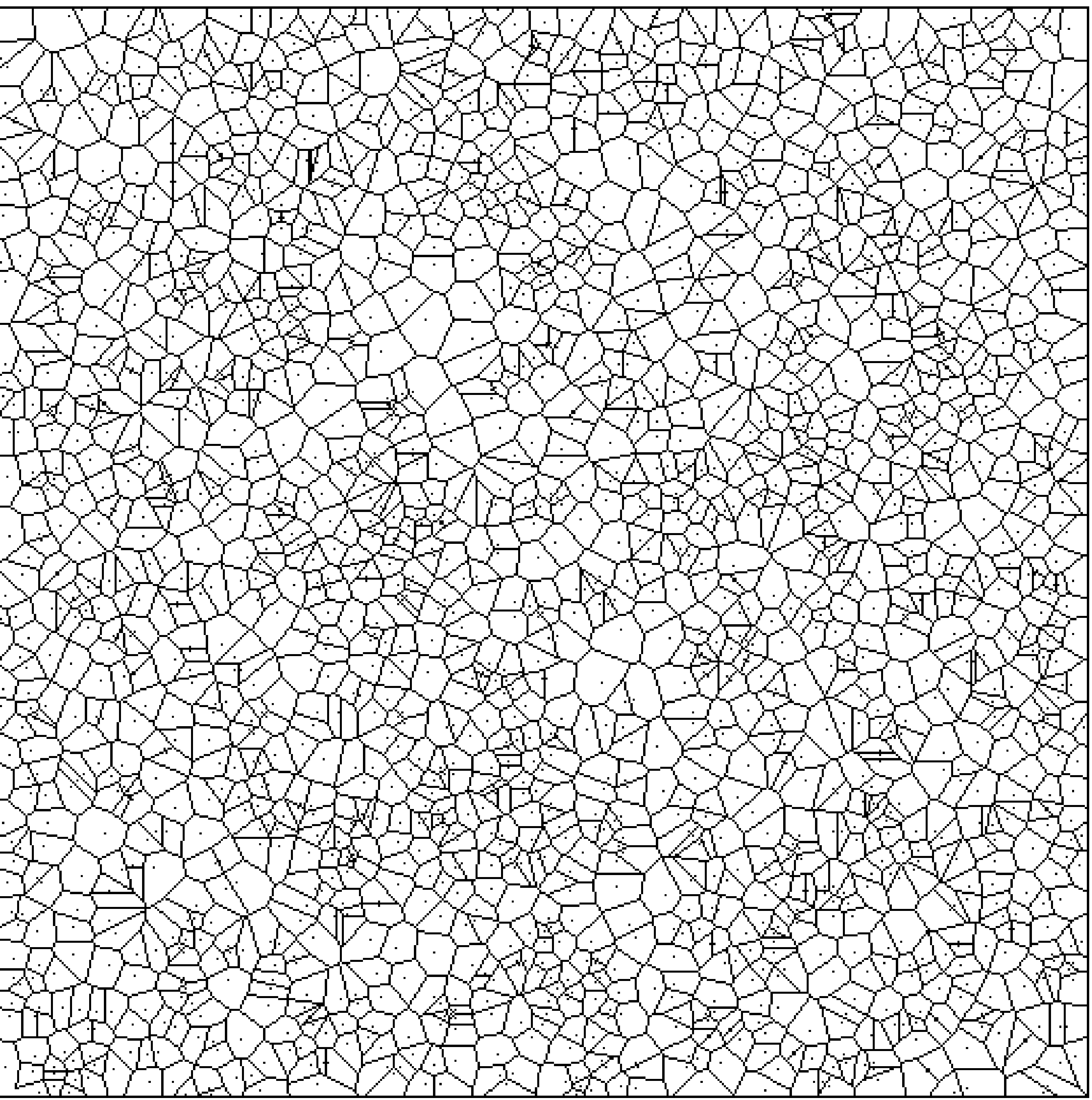
yellow->eenergy=0.6

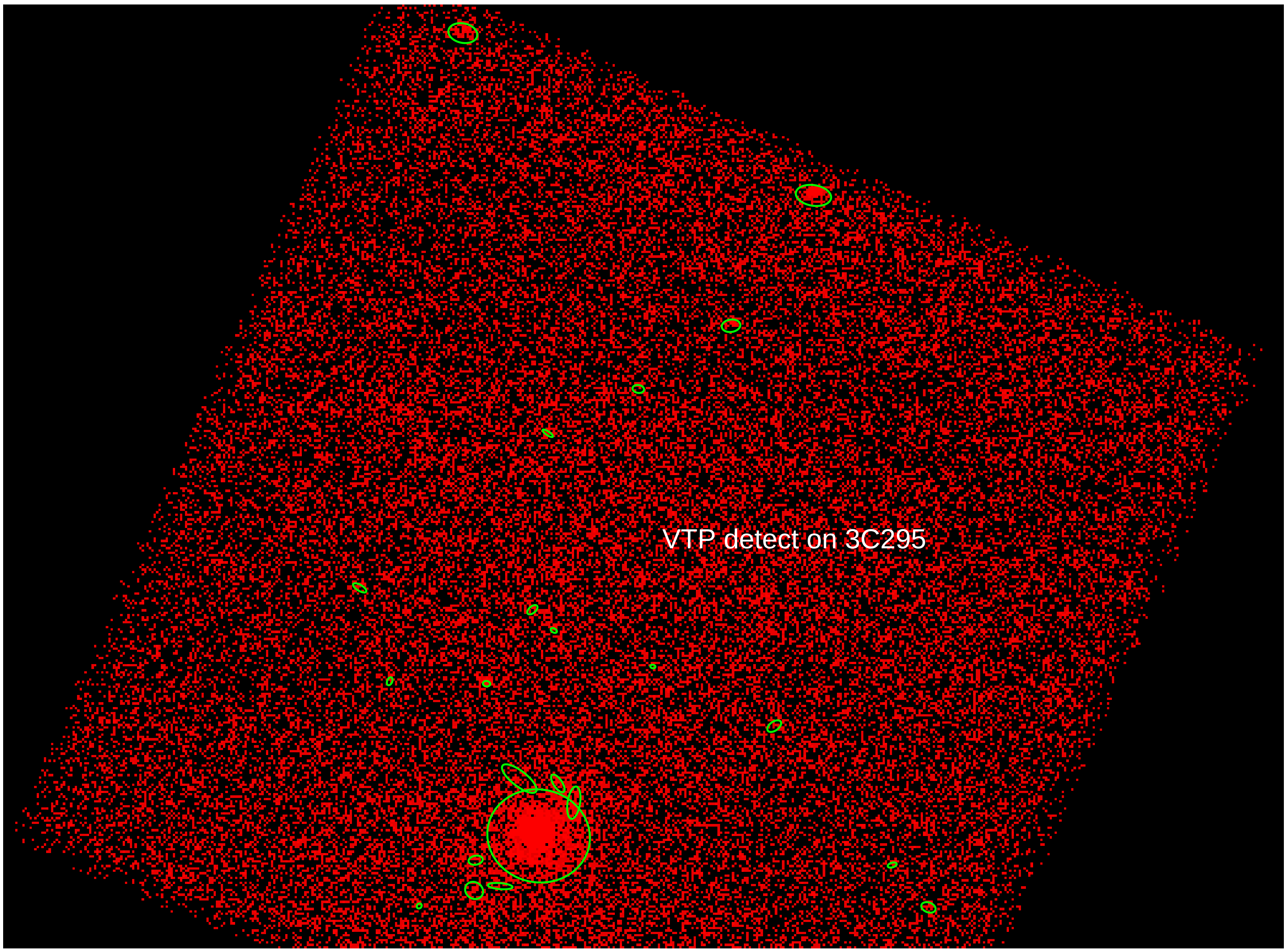


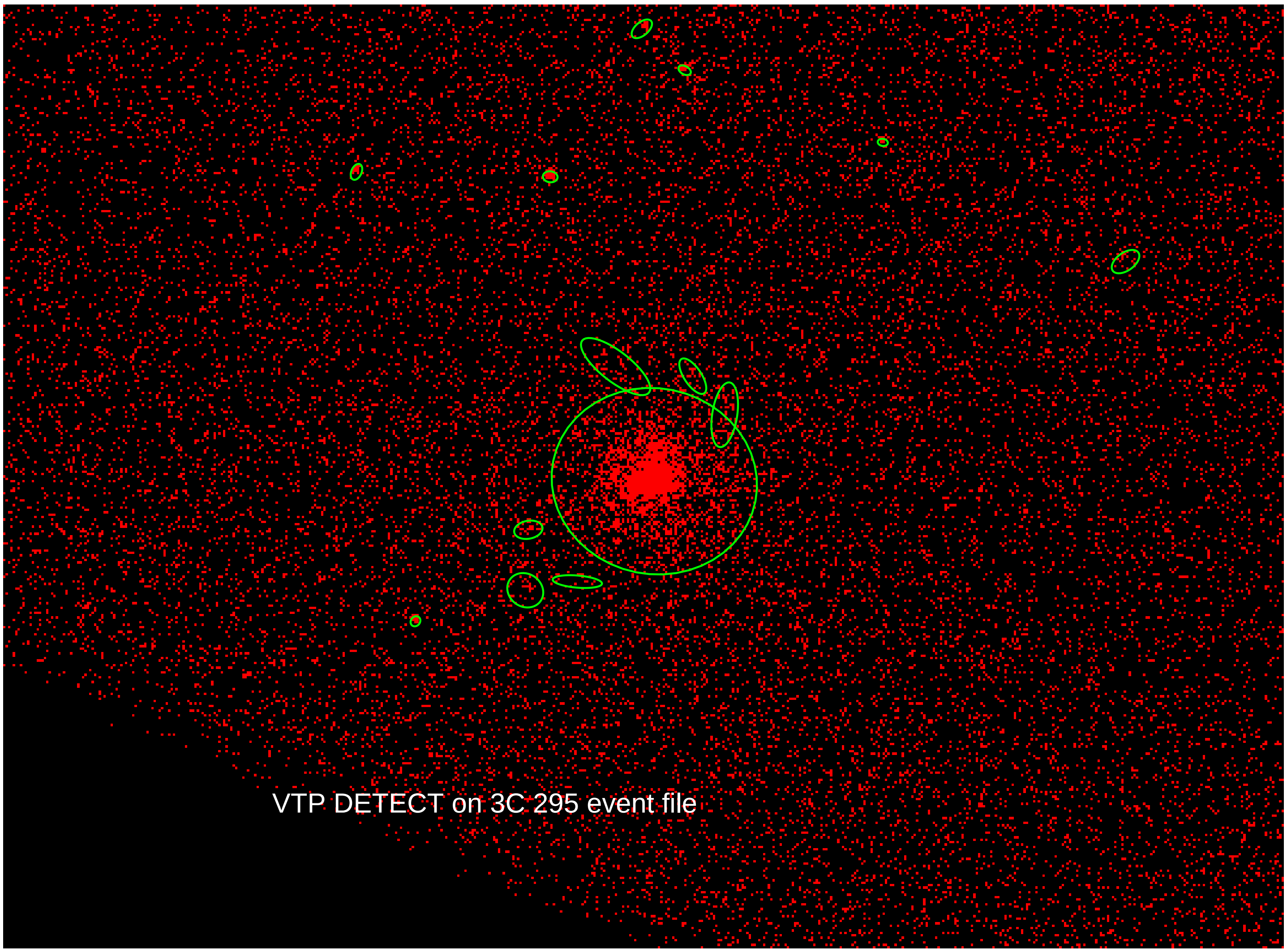

WAVDETECT - CHIP3/OFFAXIS: cyan=newPSF
6
7
7
5
3
2
1
4

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

#
#   parameter file for wavdetect
#
#
#   input
#
        infile = ../bin32.fits    Input file name
#
#   output
#
       outfile = wav_bin32new_out.fits Output source list file name
     scellfile = wav_bin32new_num.fits Output source cell image file name
     imagefile = wav_bin32new_img.fits Output reconstructed image file name
   defnbkgfile = wav_bin32new_bkg.fits Output normalized background file name
      (regfile = wav_bin32new.reg) ASCII regions output file
#
#   output options
#
      (clobber = yes)             Overwrite existing outputs?
       (kernel = default)         Output file format (fits|iraf|default)
     (ellsigma = 5)               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-07)           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
#
#   scales
#
       (scales = 1 2 4 8 16)      wavelet scales (pixels)
#
#   iteration info
#
      (maxiter = 3)               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 = /pool14/mk/PSFSIZE/psfsize4_16_2001.fits) Table of PSF size data
#
#   end of wrecon parameters
#
#########################################################################
#
#   run log verbosity and content
#
          (log = no)              Make a log file?
      (verbose = 0)               Log verbosity
#
#   mode
#
         (mode = ql)
```

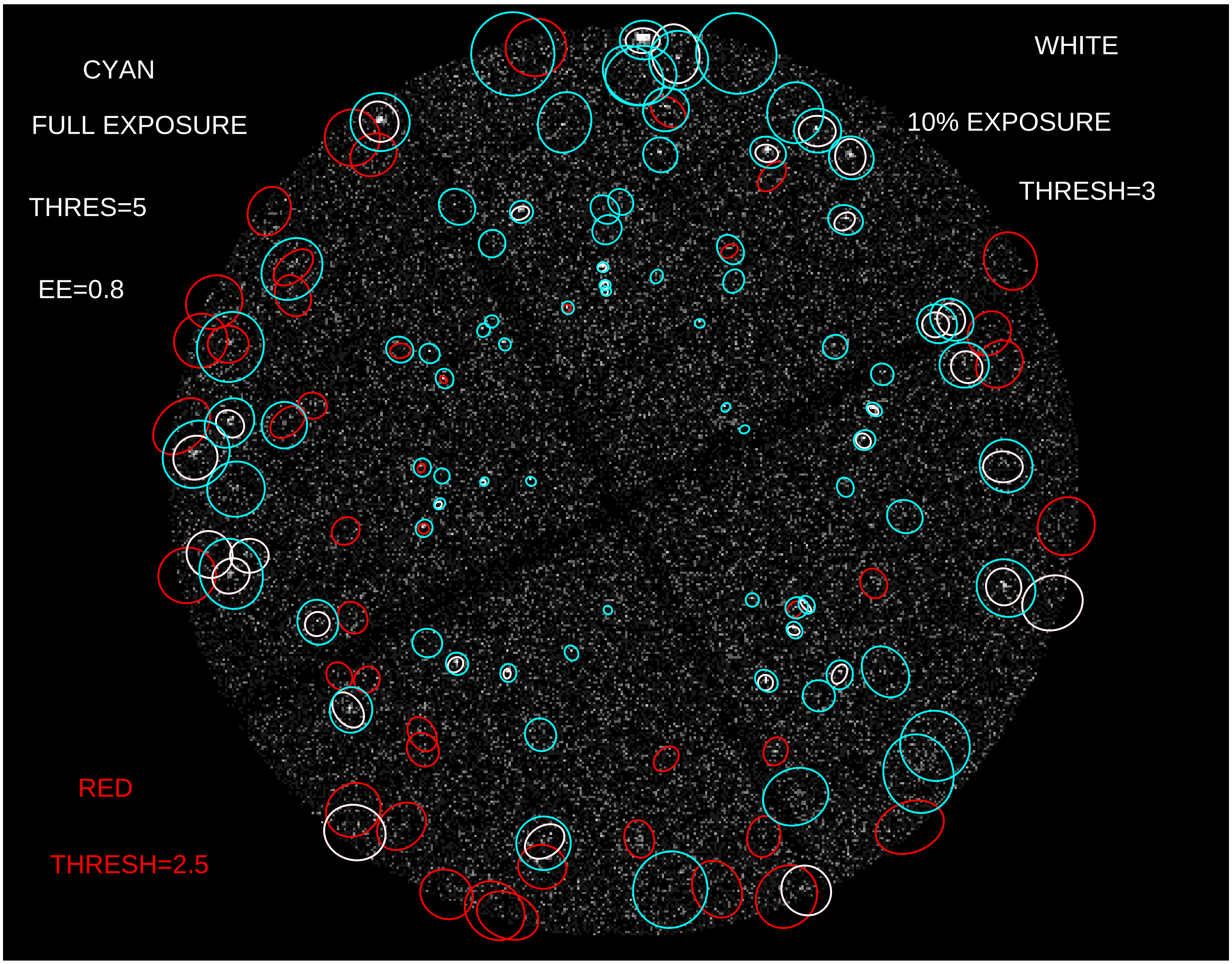


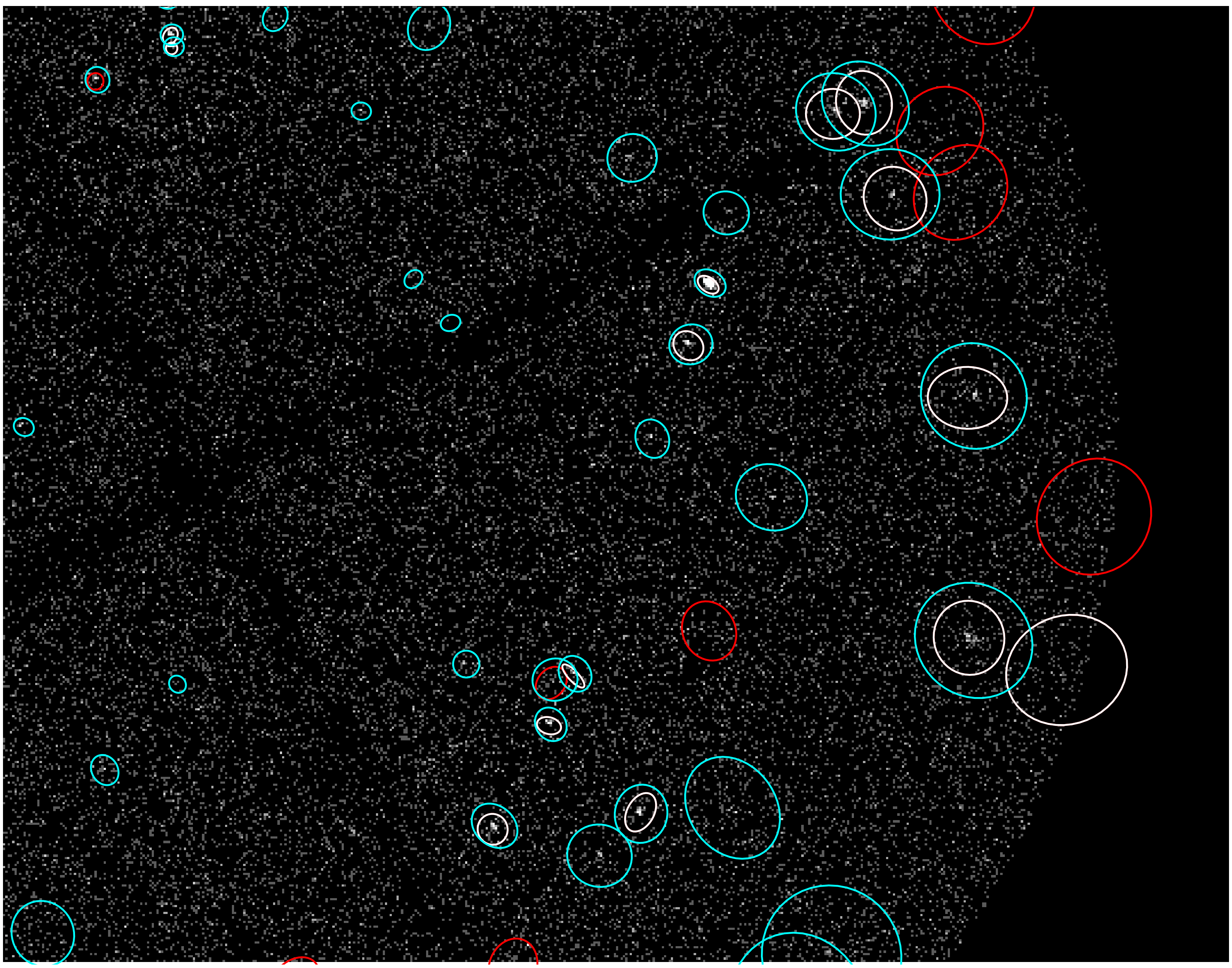

VTP detect on 3C295

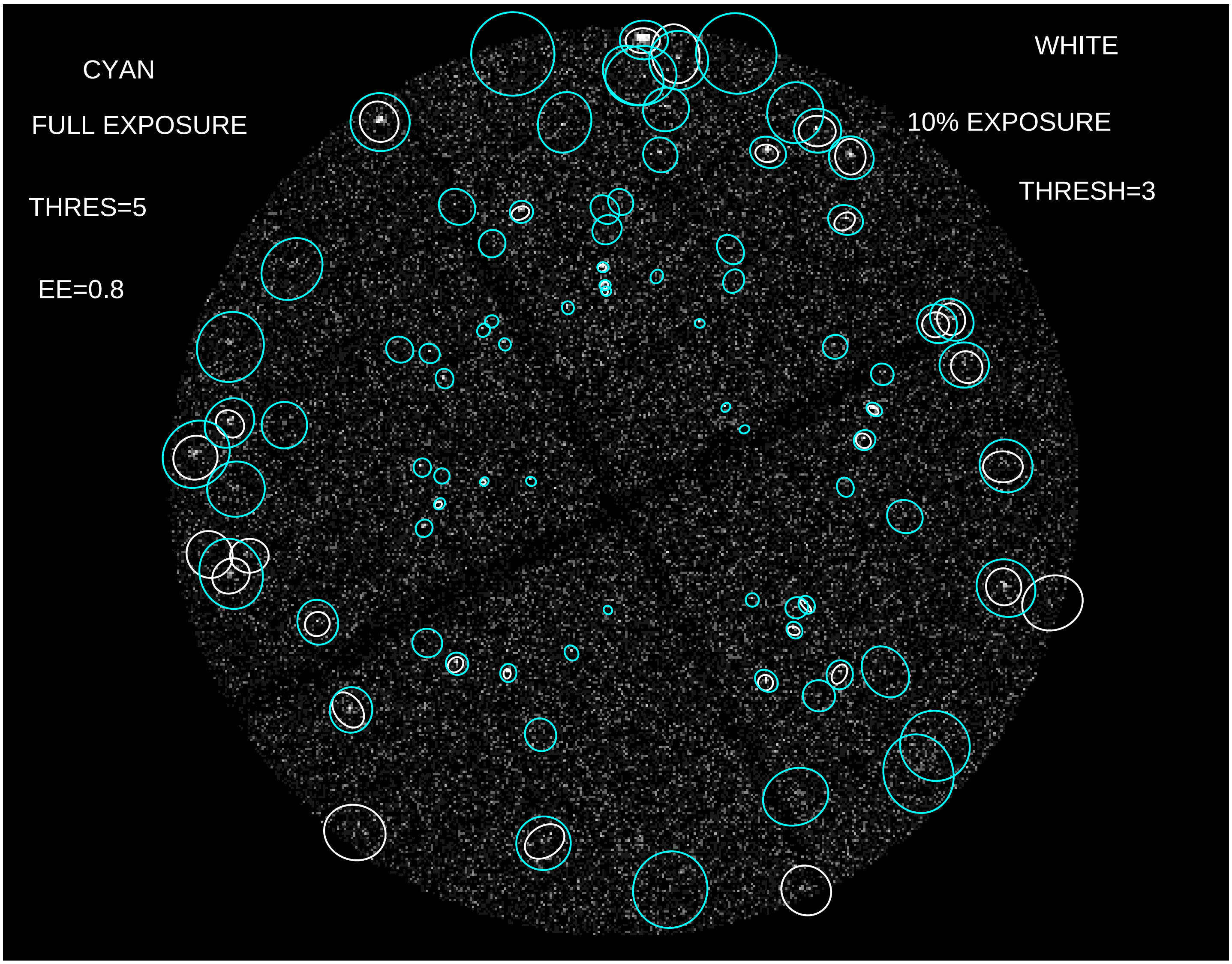

VTP DETECT on 3C 295 event file

Parameters for /home/harris/cxcds_param/vtpdetect.par

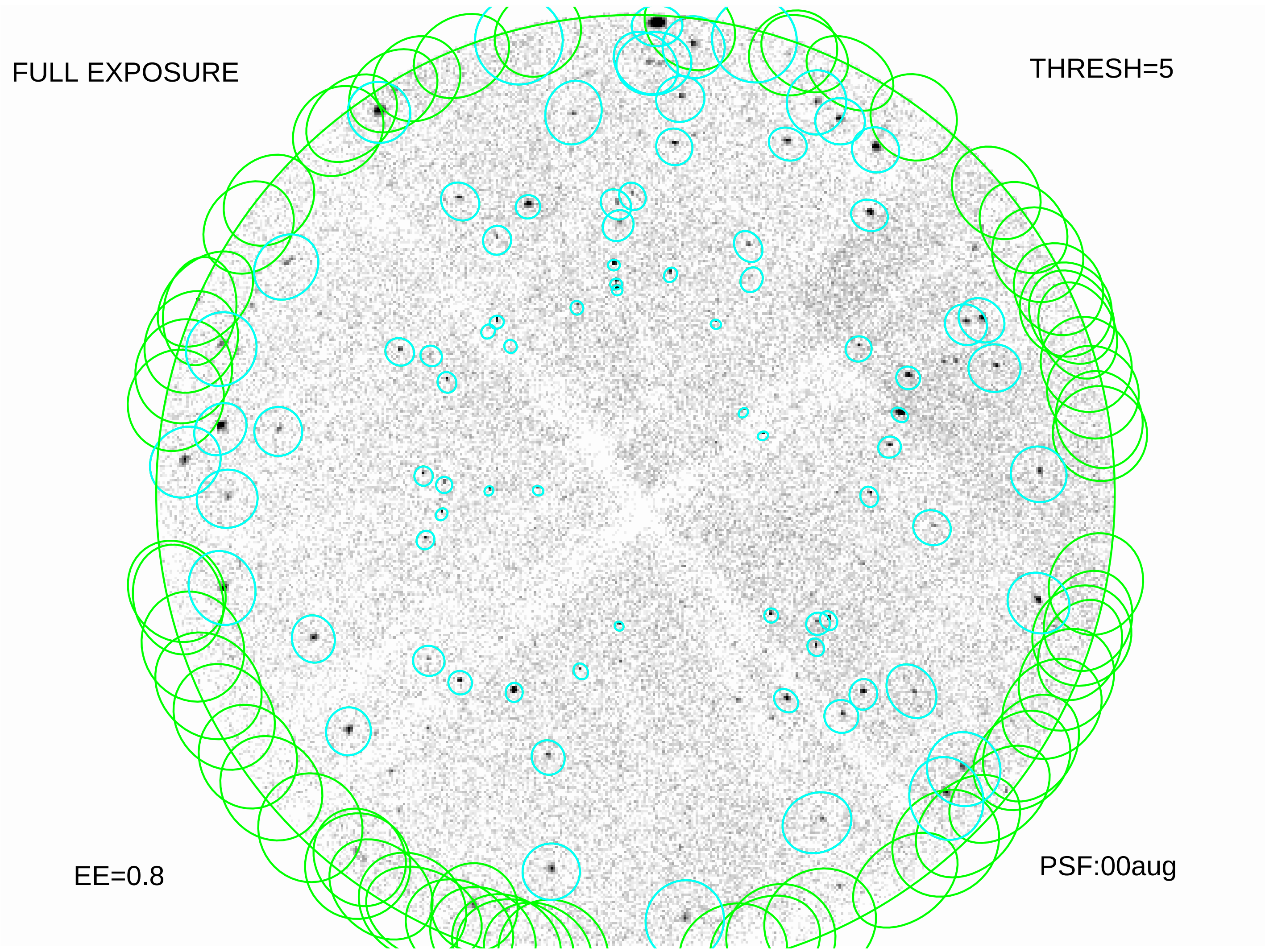
```
#
# parameters for vtpdetect
#
#
# inputs -- can either be an image or table
#
        infile = 578chip7_evt2.fits[EVENTS][cols x,y] Input file name
       expfile = none             Exposure map file name
#
# output
#
       outfile = vtp578chip7.fits Source list output file name
#
# processing parameters
#
         scale = 1                Threshold scale factor
         limit = 1e-06            Max. probability of being a false source
        coarse = 10               Minimum number of events per source
       maxiter = 10               Maximum number of iterations to allow
#
# SAOImage regions
#
      (regfile = vtp578chip7.reg) name for ASCII output region files
     (ellsigma = 3)               Size of output source ellipses (in sigmas)
         (edge = 2)               How close to edge of field to reject events
      (superdo = no)              Perform Super Voronoi Cell procedure
#
# probably use defaults for these...
#
   (maxbkgflux = 0.8)             Maximum normalized background flux to fit
   (mintotflux = 0.8)             Minimum total flux fit range
   (maxtotflux = 2.6)             Maximum total flux fit range
    (mincutoff = 1.2)             Minimum total flux cutoff value
    (maxcutoff = 3)               Maximum total flux cutoff value
       (fittol = 1e-06)           Tolerance on Possion fit
     (fitstart = 1.5)             Initial background fit starting scale factor
#
# user setable parameters
#
      (clobber = no)              Overwrite if file exists
      (verbose = 0)               Debug level
      (logfile = stderr)          Debug file name
       (kernel = default)         Output format
#
# mode
#
         (mode = ql)
```

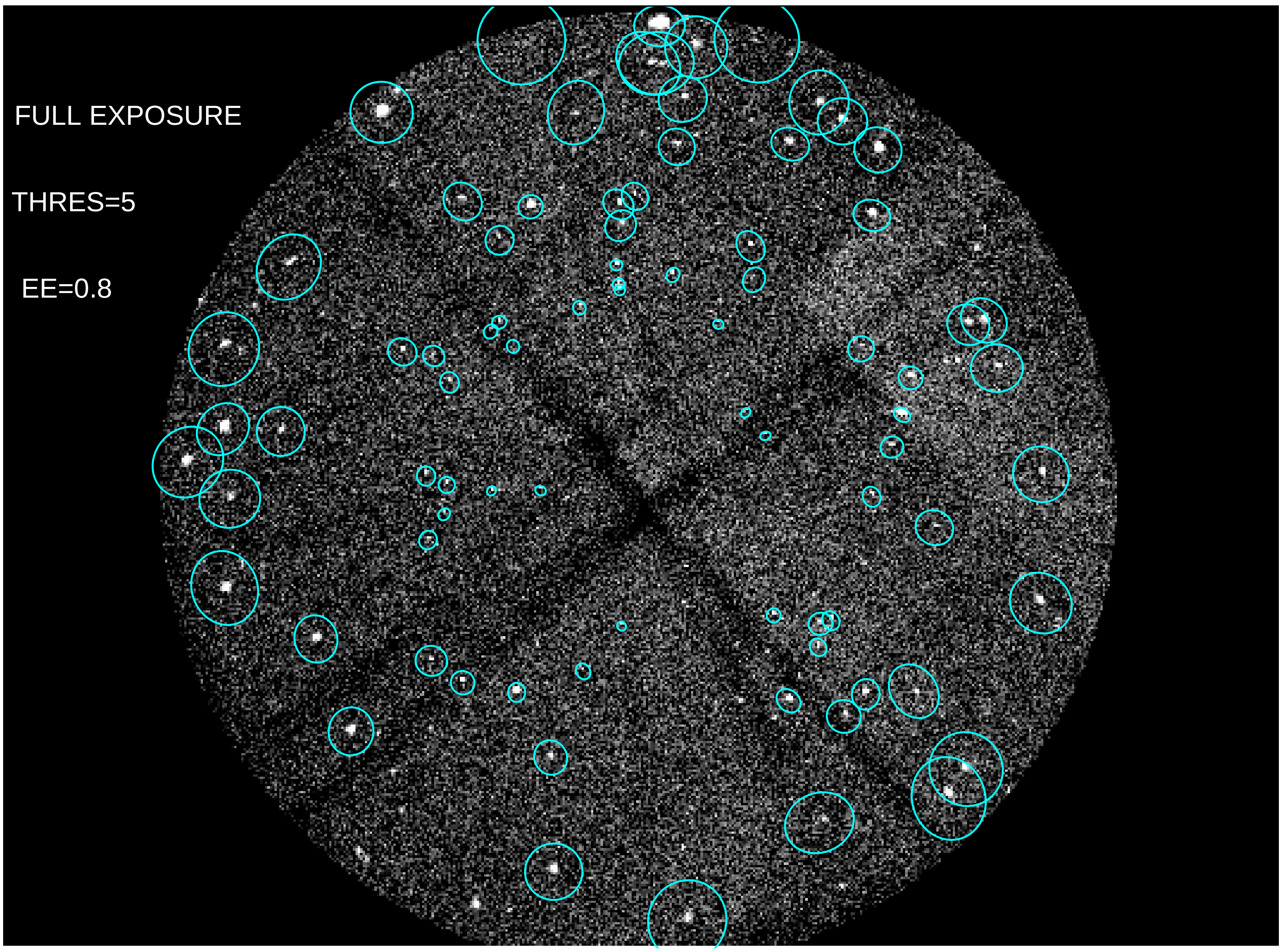

CYAN
FULL EXPOSURE
THRES=5
EE=0.8
WHITE
10% EXPOSURE
THRESH=3
RED
THRESH=2.5





CYAN
FULL EXPOSURE
THRES=5
EE=0.8
WHITE
10% EXPOSURE
THRESH=3


FULL EXPOSURE
THRESH=5
EE=0.8
PSF:00aug



FULL EXPOSURE
THRES=5
EE=0.8