Chandra X-ray Center

AHELP for CIAO 3.4   **vtpdetect**   Context: tools

Jump to: Description Examples Parameters CHANGES IN CIAO 3.0 Stack Overflow Bugs See Also

Synopsis

Voronoi Tessellation and Percolation (VTP) source detection

Syntax

```
vtpdetect infile expfile outfile scale limit coarse maxiter [edge]
[maxbkgflux] [mintotflux] [maxtotflux] [mincutoff] [maxcutoff] [fittol]
[fitstart] [clobber] [verbose] [logfile] [kernel] [superdo] [regfile]
[ellsigma]
```

Description

`vtpdetect' locates connected sets of photons above a threshold density and groups each set into a source. Events are read from the event list and reduced to unique X,Y pixel locations (possibly with a "multiplicity" greater than 1, i.e. more than 1 event). An exposure time is assigned to each pixel location.

If the input file is an image, then the image is read–in and a pseudo event list is created; events are added with a multiplicity equal to the pixel amplitude.

An area is assigned to each pixel location by computing the Delaunay triangulation from which the Voronoi tessellation is computed (the triangulation information is saved for the percolation step).

The cumulative distribution (cumulative density function, CDF) for non–source pixels is fitted on the low end (up to maxbkgflux) with a Poisson distribution. For the first pass, all pixels are considered to be background.

The minimum of the residual between the total (all pixel locations) CDF and the estimated background CDF (from above) is found between the mintotflux and maxtotflux ranges by two methods: absolute minimum and a fitted quadratic function.

A flux cutoff is determined from these two values and the values of the mincutoff and maxcutoff parameters and scaled by the user–input scale factor.

Below this flux cutoff, pixel locations are assumed to be background pixels. Above the flux cutoff, the pixels are grouped into sources based on the percolation (friends of friends) using the triangulation information to know

which pixel locations are adjacent to every other pixel.

Given the total area of the source region (sum of the area of all pixel locations in source region), the minimum number of events needed to exceed the tolerated false source limit of background fluctuations is computed. If the number of events in the source (correcting for expected background) exceeds this minimum, then the source is significant. If not, then the pixel location with the lowest flux is eliminated from the source region and the percolation and significance testing is recomputed. (The percolation needs to be recomputed since the "trimmed" event may cause the source to split into more than 1 source region.)

After all the pixel locations have been searched and the sources found significant or not (they revert back to background events), the threshold is recomputed using newly found background events and the source list is recomputed from scratch. This process continues until the number of background events converges or maxiter iteration have occurred.

Super Voronoi Cell update: After the final iteration, the Voronoi tessellation is computed from just the set of source pixel locations and the edge pixel locations (to constrain the tessellation). To better account for flux in the wings of a source profile, the total flux is taken to be the total flux contained in these "super" Voronoi cells around each source. Then the background is estimated from this larger source area.

This option is currently implemented; however, since it has not been sufficiently tested, it does not produce output other than to STDOUT at the end in lines that have "BEFORE" and "AFTER" comments.

Finally the properties of the source regions are computed and output to the FITS source list.

For a more complete description of the theory and operation of vtpdetect, please see the vtpdetect sections of the CIAO Detect Manual.

NOTES:

- See H. Ebeling and G. Wiednmann, Phys. Rev E, 47, 704 (1993).
- The output source list ("outfile" parameter) can be used by ds9 as a region file if the block name is renamed to "REGION" (e.g. dmcopy srclist.fits[srclist][region] regfile.fits).
- NB: IF THE INPUT FILE IS AN EVENTS FILE, THE FILTER "[cols x,y]" must follow the filename.

Example 1

```
vtpdetect "my_evts.fits[STDEVT][pi=0:50][cols x,y]" my_exp_0_50.fits
my_srclist.fits
```

Read an event list filtered on the PI column from 0 to 50. Use the exposure map and output the source list to my_srclist.fits.

Example 2

```
vtpdetect "my_evts.fits[STDEVT][bin x=::32,y::32]" none output.fits
```

Bin an event list into an image, using a binning factor of 32.

For other vtpdetect examples, please see the CIAO Science Thread "Detecting Sources in Imaging Observations – Using vtpdetect".

Parameters

name	type	ftype	def	min	max	reqd
infile	file	input				yes
expfile	file	input	none			
outfile	file	output				yes
regfile	file		none			
ellsigma	real		3			
scale	real		1	0		
limit	real		1e−06	0	1	
coarse	integer		10	0		
maxiter	integer		10	0	100	
edge	integer		2	0		
superdo	boolean		no			
maxbkgflux	real		0.8	0		
mintotflux	real		0.8	0		
maxtotflux	real		2.6	0		
mincutoff	real		1.2	0	10	
maxcutoff	real		3	0	10	
fittol	real		1e−06	0		
fitstart	real		1.5	0.9	2	
clobber	boolean		no			
verbose	integer		0	0	5	
logfile	file	output	stderr			
kernel	string		default			

Detailed Parameter Descriptions

Parameter=infile `(file required filetype=input)`

Input file name. This can either be an event list or an image as specified by the datamodel syntax. NB: IF THE INPUT FILE IS AN EVENTS FILE, THE FILTER "[cols x,y]" must follow the filename.

Parameter=expfile `(file filetype=input default=none)`

There is preliminary support for exposure map files. This is assumed to be a 2 D image with WCS matching the input file.

Parameters 3

Parameter=outfile `(file required filetype=output)`

Name of the output file.

Parameter=regfile `(file default=none)`

Filename for ASCII region output.

Parameter=ellsigma `(real default=3)`

Size, in sigmas, to make the elliptical source regions.

ellsigma is a multiplicative factor applied to sigma, the standard deviation of the distribution, to scale the major and minor axes of the ellipses for each source. ellsigma affects both the outfile and the ASCII region file (regfile). This feature is included so that the graphics overlay will be more visible and under the user's control. Often a value greater than 3 is helpful.

Parameter=scale `(real default=1 min=0)`

Background flux cutoff scale.

`vtpdetect' determines the background flux cutoff. This parameter allows the user to scale that value. Setting this to something greater than 1 will tend to de–blend sources while losing faint sources. Setting it to less than 1 will tend to blend source while picking out fainter sources.

Parameter=limit `(real default=1e-06 min=0 max=1)`

Number of false source events per number of background events.

Parameter=coarse `(integer default=10 min=0)`

Minimum number of events to consider to be a real source.

Parameter=maxiter `(integer default=10 min=0 max=100)`

Maximum number of iterations to allow.

Maximum number of iterations to allow. `vtpdetect' may converge before this. `vtpdetect' may also enter a state where the background estimate cycles between 2 or 3 values. Setting this to different values in this case will allow the user to control the exit level.

Parameter=edge `(integer default=2 min=0)`

The Voronoi tessellation at the edges is unbounded. This parameter controls how deep into the field to ignore events.

Parameter=superdo `(boolean default=no)`

Perform the super Voronoi cell update algorithm at the end of the program. Since this is a time consuming and uncalibrated routine this option is optional.

Parameter=maxbkgflux `(real default=0.8 min=0)`

Maximum normalized background flux level to fit with the cumulative Poisson distribution.

Parameter=mintotflux `(real default=0.8 min=0)`

Minimum normalized total flux level to fit with parabola.

Parameter=maxtotflux `(real default=2.6 min=0)`

Maximum normalized total flux level to fit with a parabola.

Parameter=mincutoff `(real default=1.2 min=0 max=10)`

Minimum absolute normalized flux level to find background cutoff.

Parameter=maxcutoff `(real default=3 min=0 max=10)`

Maximum absolute normalized flux level to find background cutoff.

Parameter=fittol `(real default=1e-06 min=0)`

Fit tolerance on Poisson fit.

Parameter=fitstart `(real default=1.5 min=0.9 max=2)`

Initial guess at Poisson fit.

Parameter=clobber `(boolean default=no)`

Remove output file if it already exists?

Parameter=verbose `(integer default=0 min=0 max=5)`

Level of informative output as the program is running. WARNING: verbose=4 and verbose=5 can produce hundreds of pages of output. Use only if needed to diagnose a problem during the percolation.

Parameter=logfile `(file filetype=output default=stderr)`

File where all messages will be directed. There are two special cases "STDOUT" and "STDERR" to indicate to use the C standard output (terminal) and standard error file.

Parameter=kernel `(string default=default)`

Selects format of output file (e.g. FITS, IRAF). "default" sets output kernel to be the same as that for the input file.

CHANGES IN CIAO 3.0

Additional Extension in Source List

A second extension, SRC_REGION, has been added to the source list fits file, that outlines the exact source regions that vtpdetect found rather than just the circumscribing ellipses. Regions in this extension are polygons (possibly with excluded polygons).

Stack Overflow

`vtpdetect' uses several recursive routines to find sources; specifically the 'P'ercolation steps. If there is a large object in the field; there may be many event positions that are grouped together. This may cause a problem on some computers where the 'stack' size is small/limited. As more and more events are added to the source the stack gets deeper and deeper and more and more stack memory is used up. The stack–size is fixed at run–time and unlike other memory cannot be grown beyond the system limits. The problem is that by the time the stack overflows no error handling can be done as the program has exhaused the available resources.

If vtpdetect is run with 'verbose=3' or higher (not recommended to run higer than 3); and the tool reports a 'Segmentation violation' at the "DEBUG: doing percolation." step; you can issuing the command 'ulimit stacksize' ((t)csh shell) or 'ulimit –s unlimited' for sh/bash/ksh users. See 'man limit' for more options. Then try re–running vtpdetect.

Bugs

See the bugs page for this tool on the CIAO website for an up–to–date listing of known bugs.

See Also

tools

celldetect, tgdetect, tgidselectsrc, tgmatchsrc, wavdetect, wrecon, wtransform

---