

*AHELP for CIAO 3.4*

## wavdetect

Context: [tools](#)*Jump to:* [Description](#) [Examples](#) [Parameters](#) [Bugs](#) [See Also](#)

## Synopsis

Mexican-Hat Wavelet source detection (wtransform+wrecon)

## Syntax

```
wavdetect infile outfile scellfile imagefile defnbkgfile [clobber]
[kernel] [interdir] [bkginput] [bkgerrinput] [outputinfix] [sigthresh]
[bkgsigthresh] [exptime] [expfile] [expthresh] [bkgttime] [scales]
[maxiter] [literation] [xoffset] [yoffset] [eband] [eenergy] [psftable]
[log] [regfile] [ellsigma] [verbose]
```

## Description

'wavdetect' operates on its input in two stages: first (in 'wtransform'), it detects putative source pixels within a dataset by repeatedly correlating it with "Mexican Hat" wavelet functions with different scale sizes. At each scale, the original image is correlated with the wavelet. Pixels with sufficiently large positive correlation values are removed from the image (as assumed sources), and subsequent correlations are performed at the same scale. This procedure of extracting source pixels from an image is called "cleansing". Typically, when very few source candidates are being found, or when an iteration-count limit is reached, the cleansing process stops. At this point, the estimated background (estimated from the cleansed image) is used to set detection thresholds, which are applied to the initial correlation image array values to identify putative sources. A set of outputs for the given wavelet scale is generated: a table of candidate sources (identified by correlation maxima), an image of the correlation of the data with the wavelet function, an image of the normalized (or flat-field) background (the image minus the "source" pixels), and the normalized background error image. Then the tool moves on to the next scale and repeats the process with a fresh copy of the image.

The second stage ('wrecon') generates a source list from information from the first stage at each wavelet scale [correlation maxima tables, normalized (flat-field) background images (with errors), and correlation images]. Each correlation maximum from stage 1 is tested to see if it represents an independent, new, source, or a source seen at other scales. For each source, a cell is computed that contains the majority of the source flux, and within that cell, source properties (location, count flux, etc.) are computed.

For a more complete description of the theory and operation of wavdetect, please see the wavdetect sections of the [CIAO Detect Manual](#).

**NOTES:**

- The paper "A Wavelet-Based Algorithm for the Spatial Analysis of Poisson Data", by P. E. Freeman et al., which describes the theory behind wavdetect, can be found in the Astrophysical Journal Supplement Series: 2002, v138, p.185. It is also available on astro-ph as astro-ph/0108429.
- If the instrument/detector is neither Chandra nor ROSAT PSPC, wavdetect cannot compute point-spread-function sizes. The tool will still run but the source characteristics will not be reliable. If the size is wrong (as the sizes returned by the PSF lib may be), then the wrong scales may be used for the determination of source parameters, and thus the source properties may be wrongly estimated. But the \*detection\* process is unaffected.
- This tool requires a lot of memory. Data structures for a 512x512 image use up 36 Mb. A 2048x2048 images requires over 300 Mb. Data sets that do not fit in physical memory will page heavily to disk and processing will run very slowly.
- This tool takes a long time to run at each wavelet scale. The default parameter file only specifies 2 wavelet scales (2 and 4). This is fine for experimentation, but for completeness a scale list of "1 2 4 8 16" would be a reasonable default for Chandra data. The scales should encompass the approximate range of PSF radii, in pixels, found in the dataset.
- Note that wavdetect has only one parameter to specify scale size. If users want control over the xscale and yscale separately, or if they want to specify which scale sizes will be used for flux estimates, or if they want to experiment with many different runs of the second part of the process without repeating the initial phase, then they should run `wtransform` followed by (multiple) runs of `wrecon`.
- The output source list ("outfile" parameter) can be used by ds9 as a region file, using block name "srclist" (e.g. ds9 img.fits -region srclist.fits[srclist]).

**Example 1**

```
wavdetect my_input.fits source_list.fits source_cell.fits image.fits
background.fits expfile=none
```

Read a primary image from "my\_input.fits", generating a source list "source\_list.fits", a source cell image "source\_cell.fits" a source image "image.fits" and a normalized background image "background.fits".

**Example 2**

```
wavdetect my_input.fits source_list.fits source_cell.fits image.fits
background.fits scales="1 2 4 8 16"
```

Run wavdetect with 5 scales. Note that quotes around the scale list are required.

**Example 3**

```
wavdetect "my_input.fits[events]" source_list.fits source_cell.fits
image.fits background.fits scales="2 4 8" maxiter=3
```

Run wavdetect on an event list in file my\_input.fits, blockname "events". Use 3 scales, and allow 3 cleansing iterations per scale.

**Example 4**

```
wavdetect my_input.fits source_list.fits source_cell.fits image.fits
```

```
background.fits expfile=none exptime=0
```

Read a primary image from "my\_input.fits", generating a source list "source\_list.fits", a source cell image "source\_cell.fits" a source image "image.fits" and a normalized background image "background.fits". No exposure map is input, and exposure time is not set. A value of 1.0 will be used to estimate rate parameters in the output source list.

## Example 5

```
wavdetect my_input.fits source_list.fits source_cell.fits image.fits
background.fits expfile=none exptime=10000.0
```

Read a primary image from "my\_input.fits", generating a source list "source\_list.fits", a source cell image "source\_cell.fits" a source image "image.fits" and a normalized background image "background.fits". No exposure map is input, and the exposure time is set to 10000.0 This value will be used to estimate rates in the output source list.

## Example 6

```
wavdetect my_input.fits source_list.fits source_cell.fits image.fits
background.fits expfile=expmap.fits exptime=0
```

Read a primary image from "my\_input.fits", generating a source list "source\_list.fits", a source cell image "source\_cell.fits" a source image "image.fits" and a normalized background image "background.fits". An exposure map, expmap.fits is input. Values from the exposure map at source positions will be used to estimate rates in the output source list.

## Example 7

```
wavdetect my_input.fits source_list.fits source_cell.fits image.fits
background.fits expfile=expmap.fits exptime=10000.0
```

Read a primary image from "my\_input.fits", generating a source list "source\_list.fits", a source cell image "source\_cell.fits" a source image "image.fits" and a normalized background image "background.fits". Both exposure map and exposure time are set. Rates in the output source list will be estimated from exptime, scaled by the quantity (expmap value at source location)/(max expmap value).

For other wavdetect examples, please see the CIAO Science Thread "[Detecting Sources in Imaging Observations – Using wavdetect](#)".

## Parameters

name	type	ftype	def	min	max	units	reqd	autoname
<u>infile</u>	file	input					yes	
<u>outfile</u>	file	output					yes	yes
<u>scellfile</u>	file	output					yes	yes
<u>imagefile</u>	file	output					yes	yes
<u>defnbgfile</u>	file	output					yes	yes
<u>regfile</u>	file							yes

<u>clobber</u>	boolean		no				no	
<u>kernel</u>	string		default				no	
<u>ellsigma</u>	real		3.0					
<u>interdir</u>	file	output	.				no	
<u>bkginput</u>	file	input					no	
<u>bkgerrinput</u>	boolean		no				no	
<u>outputinfix</u>	file						no	
<u>sigthresh</u>	real		1e-06				no	
<u>bkgsigthresh</u>	real		0.001				no	
<u>exptime</u>	real		0			seconds		
<u>expfile</u>	file	input						
<u>expthresh</u>	real		0.1				no	
<u>bkgtime</u>	real		0				no	
<u>scales</u>	string		2.0 4.0					
<u>maxiter</u>	integer		2				no	
<u>iterstop</u>	real		0.0001				no	
<u>xoffset</u>	integer		INDEF			pixels	no	
<u>yoffset</u>	integer		INDEF			pixels	no	
<u>eband</u>	real		1.4967			keV		
<u>eenergy</u>	real		0.393				no	
<u>psftable</u>	file	ARD						
<u>log</u>	boolean		no				no	
<u>verbose</u>	integer		0	0	5		no	

## Detailed Parameter Descriptions

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

*Input image file. Depending on available memory, beware of primary images larger than 1024.*

Pixel values should be reasonable numbers. WAVDETECT probability distributions have been derived assuming Poisson statistics. WAVDETECT simply will not work with fluxed images. The calculated correlation values are so small that no detections occur.

### Parameter=outfile (file required filetype=output autoname=yes)

*File name of output source list.*

If auto-naming (outfile=.) is used, the output file will have the suffix "\_src"

### Parameter=scellfile (file required filetype=output autoname=yes)

*File name of the image showing the source cells, which delimit the image pixels which are used to estimate source properties (e.g. count rate).*

If auto-naming is used (scellfile=.), the output file will have the suffix "\_scell"

**Parameter=imagefile (file required filetype=output autoname=yes)**

*Output file containing the reconstructed source image.*

If auto-naming is used (imagefile=.), the output file will have the suffix "\_image"

**Parameter=defnbkgfile (file required filetype=output autoname=yes)**

*Default normalized background.*

During its second stage, wavdetect makes one normalized background estimate from the stack of per-scale normalized backgrounds produced during the first stage. It writes this computed background to this file. If auto-naming is used (defnbkgfile=.), the output file will have the suffix "\_nbkg"

**Parameter=regfile (file default= autoname=yes)**

*File for ASCII region output.*

If auto-naming is used (regfile=.), the output file will have the suffix "\_reg". NB: autonaming for regfile is not currently operational.

**Parameter=clobber (boolean not required default=no)**

*If set to "yes", existing output files will be overwritten.*

**Parameter=kernel (string not required default=default)**

*output filetype: (fits | iraf | default)*

The datamodel will support more than one output format (eg: "fits", "iraf"). This parameter selects among them. Use "default" to set the output kernel to be the same as that for the input file.

**Parameter=ellsigma (real default=3.0)**

*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=interdir (file not required filetype=output default=.)**

*Directory for intermediate results.*

**Parameter=bkginput (file not required filetype=input default=)**

*Pre-determined background input image.*

Use a previously computed background in the specified file in place of constructing a new default normalized background. When using bkginput, enter "none" for the default normalized background file.

**Parameter=bkgerrinput (boolean not required default=no)**

*If yes, use background error in file bkginput[2].*

**Parameter=outputinfix (file not required default=)**

*If needed to avoid file naming collisions with other wavdetect users, set to a unique [string] to give the set of outputs differing names. The string will be embedded in the file names.*

**Parameter=sigthresh (real not required default=1e-06)**

*Threshold for identifying a pixel as belonging to a source.*

*\*After\* the final background estimate B is made, the threshold correlation is recomputed by solving for C<sub>o</sub>:*

$$\text{sigthresh} = \text{integral}(C_o \text{ dC}) \text{PSD}(C(B))$$

Here,  $C = \text{avg}[W*D]$ , where D are the *\*raw\** data. (The iterations are simply to compute background. Once we have that, we go back to the beginning.)

If C is less than or equal to C<sub>o</sub> in a pixel, then that pixel is considered to be associated with a source. If the pixel is also the location of a local maximum in C-space, the location of that maximum is output to a FITS table.

sigthresh should not be significantly larger than one over the number of pixels in the image; for a 1024x1024 image, that means sigthresh  $\sim 1.e-6$ . If, in this case, sigthresh = 1.e-5, there will be approximately 10 strong background fluctuations detected as sources.

sigthresh should not be smaller than *\*roughly\** 1.e-9 – 1.e-10; even at this point, the accuracy of the computed detection thresholds is unknown (though probably fine for most applications).

**Parameter=bkgsigthresh (real not required default=0.001)**

*Significance threshold for cleansing data during iterations.*

Significance threshold for cleansing data during iterations. In each pixel, during each iteration, the background, and correlation of the wavelet function and the current data ( $C' = \text{avg}[W*D']$ ), are estimated. The background estimate B' implies a probability sampling distribution (PSD) for C', i.e. the distribution that C' would have if there was locally no source in the image and D' were sampled from background. A threshold C<sub>o'</sub> is calculated from this PSD, using the supplied value of bkgsigthresh (e.g.  $10^{*-2}$ ) by solving the following for C<sub>o'</sub>:

$$\text{bkgsigthresh} = \text{integral}(C_o' \text{ dC}') \text{PSD}(C'(B'))$$

If C' is greater than or equal to C<sub>o'</sub>, the data in the pixel is replaced with the background estimate. In this way, putative sources are eliminated from the data image, along with weak (but otherwise undetectable) sources and background fluctuations, so that a better background estimate can be made.

This value should be no larger than 0.05 (the usual 5 per cent, or 95 per cent, statistical criterion for rejecting the null hypothesis, which is that the pixel in question has data sampled solely from the background); it should not be smaller than sigthresh.

**Parameter=exptime (real default=0 units=seconds)**

*Time of the exposure, in seconds.*

If set to zero and no exposure file is used, the program sets the exposure time to 1.0 when estimating source rates.

If set to a non-zero value and no exposure file is used, the program uses the input value when estimating source rates.

If set to zero and an exposure file is used, the program uses the value from the exposure file when estimating source rates.

If set to a non-zero value and an exposure file is used, the program uses the quantity  $\text{exptime} * (\text{expfile value at source} / \text{max expfile value})$  when estimating source rates.

**Parameter=expfile (file filetype=input default=)**

*Exposure file. Image of the data set exposure time.*

**Parameter=expthresh (real not required default=0.1)**

*If relative exposure is less than the parameter expthresh, then the pixel in question is not analyzed*

For each pixel, a relative exposure is calculated (pixel exposure value over maximum value of exposure in map). If this relative exposure is less than the parameter expthresh, then the pixel in question is not analyzed: the background, correlation, et al., are set to zero.

expthresh should not be less than 0.1 (or else the accuracy of normalization will be too low); if set close to 1, only very limited regions of the FOV will be considered when source correlation maxima are listed. Typical values would be 0.1–0.2.

**Parameter=bkgtime (real not required default=0)**

*Livetime of predetermined background.*

Livetime of predetermined background. This parameter should only be used *if* the user has provided a normalized background image (rather than having wavdetect calculate it). A value of 0 will make the exposure time used for background time. If bkgtime is non-zero and no background map is supplied, the source parameters will be adversely affected (i.e. wrong).

**Parameter=scales (string default=2.0 4.0)**

*The wavelet radii, in pixels, are given by this list of numbers separated by spaces. Quotes are required around the list of scales. NB: normal ciao syntax is to use commas for separating a list of variables; here one must use spaces.*

Wtransform produces a complete set of outputs for each scale. Small scales tend to detect small features, and larger scales larger features.

Values are typically  $2^{**n}$ , where n is an integer, and n\_lo and n\_hi are chosen with respect to instrumental PSF sizes. For the ROSAT PSPC, for instance, this could mean n\_lo = 1 pixel, n\_hi = 16 pixels. Note that larger scales might need to be used to characterize (i.e. derive good source property estimates for) extended sources.

$2 \times n$  should not be larger than the size of the image (in pixels) divided by 5 (in which case the extent of the wavelet function is larger than the image itself, which could lead to strange results).

**Parameter=maxiter (integer not required default=2)**

*Maximum number of cleansing iterations per scale pair. If maxiter is greater than necessary, the program will usually stop on its own (see iterstop).*

**Parameter=iterstop (real not required default=0.0001)**

*If the ratio of \*newly\* cleansed pixels to the overall number of pixels in the dataset is less than the parameter iterstop, the program quits iterating and uses the current background estimate as the final background estimate.*

The user specifies how many iterations the program should go through to calculate the background. However, the background is for all intents and purposes calculated if there are very few new pixels being cleansed (see bkgsthresh). If the ratio of \*newly\* cleansed pixels to the overall number of pixels in the dataset is less than the parameter iterstop, the program quits iterating and uses the current background estimate as the final background estimate.

A typical value is 0.001: stop iterating when only one of every thousand pixels is being cleansed. This parameter should not be larger than 1, nor smaller than one over the number of pixels in the dataset.

**Parameter=xoffset (integer not required default=INDEF units=pixels)**

*Offsets of x- and y-axis for the calculation of the off-axis angle.*

By default (when both offsets are set to INDEF), wavdetect calculates the off-axis angle using the nominal pointing of the data file as the origin. Users may change this behavior by specifying offsets to any numerical values; the offsets provide the location of the optical axis with respect to the center of the data file.

A typical scenario in which the users may select to use offsets is when their data file is a sum of two or more observations, with different pointings. The nominal pointing in such data file is usually poorly defined and the off-axis angle could be calculated from an undesired origin, leading to suboptimal selection of the detect cell size. The ability to set the origin explicitly with x/yoffset solves this problem.

**Parameter=yoffset (integer not required default=INDEF units=pixels)**

*See "xoffset".*

**Parameter=eband (real default=1.4967 units=keV)**

*Energy, in keV, at which the PSF size should be computed.*

For ROSAT PSPCB, eband and eenergy parameters are ignored. The values 1 keV and .393 are used.

**Parameter=eenergy (real not required default=0.393)**

*Percentage of encircled energy, expressed as a fraction of 1.0, which the PSF size should contain.*

Percentage of encircled energy, expressed as a fraction of 1.0, which the PSF size (see "eband") should contain. Suggested values: .393 (one-sigma percentage threshold for a two-dimensional Gaussian), or 0.500. The efficiency of the algorithm is reduced if this fraction is made too high (say, 0.900) or too low.



The encircled energy fraction of the PSF is used to estimate the PSF size, in pixels, at the location of each detected source. The default value (0.393) is the one-sigma integrated volume of a normalized two-dimensional Gaussian, thus the estimated PSF size should be approximately one sigma. The PSF size is used only to select a flux image; a source cell is computed using this image, and source properties are estimated using the data within the computed cell. NB: Unlike celldetect, the PSF library and eenergy have no effect on the detection process in wavdetect.

If the value of eenergy is set too high, there is the risk that the computed source cell will contain more than one source, rendering source property estimates moot. On the other hand, if the value is set too low, then the computed cell may be too small and contain only a fraction of the source counts.

For ROSAT PSPCB, eband and eenergy parameters are ignored. The values 1 keV and .393 are used.

**Parameter=psftable (file filetype=ARD default=)**

*Table of PSF size data.*

FITS file of PSF information. A file for Chandra is included with the release. The user may not specify "CALDB" for this parameter.

**Parameter=log (boolean not required default=no)**

*If set to "no", log information will go to stdout. If set to "yes", file "wavdetect.log" will be created.*

**Parameter=verbose (integer not required default=0 min=0 max=5)**

*Level of log output. 0=none, 5=highest.*

## 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](#), [tgmatsrc](#), [vtpdetect](#), [wrecon](#), [wtransform](#)

---

The Chandra X-Ray Center (CXC) is operated for NASA by the Smithsonian Astrophysical Observatory.  
60 Garden Street, Cambridge, MA 02138 USA.  
Smithsonian Institution, Copyright © 1998–2006. All rights reserved.

URL:  
<http://cxc.harvard.edu/ciao3.4/wavdetect.html>  
Last modified: December 2006

