




---

 AHELP for CIAO 3.4

## wtransform

Context: [tools](#)

*Jump to:* [Description](#) [Example](#) [Parameters](#) [Bugs](#) [See Also](#)

---

## Synopsis

Obtain wavelet transform coefficients for poisson image data

## Syntax

```
wtransform infile srclist correl nbkg [bkginput] [bkgerrinput] [thresh]
[bkg] [correlerr] [nbkgerr] [bkgerr] [outputinfix] [clobber] [kernel]
[sigthresh] [bkgsigthresh] [exptime] [expfile] [expcor] [expthresh]
[bkgtime] [xscales] [yscales] [maxiter] [iterstop] [stall] [verbose]
[log]
```

## Description

`wtransform' performs detection on a dataset by repeatedly correlating it with "Mexican Hat" wavelet functions with different scale sizes. At each scale, wtransform starts with the original image, and correlates it 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.

Note: As a stand-alone tool, wtransform is only intended for use by those with a thorough understanding of the wavelet detection process. Most users should use the `wavdetect' script.

This tool requires a lot of memory. Data structures for a 512x512 image use up 36 meg. A 2048x2048 images requires over 300 meg. Data sets that do not fit in physical memory will page heavily to disk and processing will run very slowly.

Each cleansing iteration is time-consuming.

See: "WDETECT and WRECON: Wavelet Transform Techniques for Spatial Analysis of AXAF Data" Freeman, Kashyap, Rosner, Lamb. HEAD '97 Conference

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

## Example

```
wtransform pleiades.fits pleiades_src_stk.lis pleiades_correl_stk.lis
pleiades_nbkg_stk.lis xscales="1 2 4 8 16" yscases="1 2 4 8 16"
```

Read a primary image from "pleiades.fits". Write a series of source candidate list files, named in "pleiades\_src\_stk.lis". Write a series of correlation images, named in "pleiades\_correl\_stk.lis". Write a series of normalized background images, named in "pleiades\_nbkg\_stk.lis". Examine the input (and generate outputs) at pixel scales 1 2 4 8 and 16.

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>src</u> list	file	output					yes	yes
<u>correl</u>	string	output					yes	yes
<u>nbkg</u>	string	output					yes	yes
<u>bkg</u>	string	output						yes
<u>thresh</u>	string	output						yes
<u>outputinfix</u>	file							
<u>exptime</u>	real		0			seconds		
<u>expfile</u>	string	input						
<u>expcor</u>	string		fast					
<u>expthresh</u>	real		0.1					
<u>bkgtime</u>	real		0			seconds		
<u>bkginput</u>	string	input					no	
<u>bkgerrinput</u>	boolean		no				no	
<u>xscases</u>	string		2.0			pixels		
<u>yscales</u>	string		2.0			pixels		
<u>clobber</u>	boolean		no					
<u>correlerr</u>	boolean		yes					
<u>nbkgerr</u>	boolean		yes					

<u>bkgerr</u>	boolean		no					
<u>maxiter</u>	integer		2					
<u>iterstop</u>	real		0.0001					
<u>sigthresh</u>	real		1e-06					
<u>bkgsigthresh</u>	real		0.001					
<u>kernel</u>	string		default					
<u>log</u>	boolean		no					
<u>verbose</u>	integer		0	0	5			
<u>stall</u>	boolean		no					

## Detailed Parameter Descriptions

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

*Input file*

If the input data are not a FITS primary image, this name must include the extension, eg: "foo.fits[EVENTS]".

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

*File name for stack (list) of per-scale candidate source output files.*

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

**Parameter=correl (string required filetype=output autoname=yes)**

*Name of file to contain stack (list) of per-scale correlation output images.*

If auto-naming is used (correl=.), the output file will have the suffix "\_correl\_stk"

**Parameter=nbkg (string required filetype=output autoname=yes)**

*File name of stack (list) of normalized (i.e. flat-field) background output images created at each scale.*

If auto-naming is used (nbkg=.), the output file will have the suffix "\_nbkg\_stk"

**Parameter=bkg (string filetype=output default= autoname=yes)**

*Name of file to contain stack (list) of per-scale plain background images. Set to "none" to disable.*

If auto-naming is used (bkg=.), the output file will have the suffix "\_bkg\_stk".

**Parameter=thresh (string filetype=output default= autoname=yes)**

*File name of stack (list) of per-scale source detection correlation threshold output images.*

If auto-naming is used (thresh=.), the output file will have the suffix "\_thresh\_stk".

**Parameter=outputinfix (file default=)**

*Leave blank. This parameter controls extra text to insert in output filenames. for internal use only.*

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

*Time of the exposure, in seconds. If set to zero, the program estimates the exposure time from the exposure map.*

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

*Image of exposure time for dataset.*

**Parameter=expcor (string default=fast)**

*Fast or Full or None. Correct for obstructions and vignetting in the telescope. "none"=no corrections. If in doubt, use the "fast" correction.*

"full": assume  $C_{correct} = C_{measured} - \text{avg}(W*(EB\_norm)) + \text{avg}(W*B\_norm)$   
E C = correlation, at pixel in question  
E = exposure  
B\_norm = normalized background  
W = wavelet function  
 $\text{avg}(W*(EB\_norm))$  equal to or greater than correlation of wavelet function and product of exposure and normalized background, at pixel in question  
 $\text{avg}(W*B\_norm)$  equal to or greater than correlation of wavelet function and normalized background  
This correction is applied during each iteration, since B\_norm changes.  
"fast": assume  $C_{correct} = C_{measured} - B\_norm * \text{avg}(W*E)$

**Parameter=expthresh (real default=0.1 min= max=)**

*For each pixel, a relative exposure is calculated (pixel exposure time over detector livetime). 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 default=0 min= units=seconds)**

*Livetime of predetermined background.*

This parameter is only used \*if\* the user has provided a normalized background image. A value of 0 means use the exposure time as the background time.

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

*Pre-determined background input image.*

Use a previously computed background in the specified file in place of constructing new per-scale normalized backgrounds.

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

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

**Parameter=xcales (string default=2.0 units=pixels)**

*The wavelet scale sizes in the x direction, given by pixel values in this list of numbers.*

The number of scale sizes (a series of numbers separated by spaces) indicated by xcales must be the same as yscales. The scale sizes themselves for a particular run do not have to be equal, though initial usage of asymmetric wavelets is not advised.

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.

$2^n$  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=yscales (string default=2.0 units=pixels)**

*The wavelet scale size in the y direction, given by pixel values in this list of numbers.*

The number of scale sizes indicated by yscales must be the same as xcales. See additional notes under 'xcales'.

**Parameter=clobber (boolean default=no)**

*If "yes", existing outputs will be overwritten.*

**Parameter=correlerr (boolean default=yes)**

*If set to "yes", wtransform will include an error image as part of the correlation image.*

**Parameter=nbgerr (boolean default=yes)**

*If set to "yes", wtransform will include an error image as part of the normalized background images (these outputs are needed by wrecon).*

**Parameter=bkgerr (boolean default=no)**

*If set to "yes", wtransform will include an error image as part of the plain background image.*

**Parameter=maxiter (integer default=2 min=)**

*Maximum number of cleansing iterations per scale pair. 4 is usually adequate. If the maxiter is greater than necessary, the program will usually stop on its own (see iterstop).*

**Parameter=iterstop (real default=0.0001 min= max=)**

*The user specifies how many iterations the program should go through to calculate the background (parameter iters).*

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 mask 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 0.0001: stop iterating when only one of every ten thousand pixels is being cleansed. This parameter should not be larger than 1, nor smaller than one over the number of pixels in the image/mask.

**Parameter=sigthresh (real default=1e-06 min= max=)**

*\*After\* the final background estimate B\_final is made, the threshold correlation is recomputed by solving for C\_o: sigthresh = integral(C\_o dC) PSD(C(B\_final))*

*\*After\* the final background estimate B\_final is made, the threshold correlation is recomputed by solving for C\_o: sigthresh = integral(C\_o dC) PSD(C(B\_final))*

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 greater than or equal to C\_o 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/mask; for a 1024x1024 image, that means sigthresh ~ 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 default=0.001 min= max=)**

*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\_o' is calculated from this PSD, using the supplied value of bkgsigthresh (e.g. 0.01) by solving the following for C\_o':

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

If

$$C' \text{ is greater than or equal to } C_o'$$

, the data in the pixel is replaced with the background estimate. In this way, putative sources are eliminated from

## Ahelp: wtransform – CIAO 3.4

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%, or 95%, 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 `sighthresh`.

**Parameter=kernel (string default=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=log (boolean default=no)**

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

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

*Level of debugging output*

**Parameter=stall (boolean default=no)**

*Special mode to rendezvous a debugger with wtransform. Leave set to "no".*

## 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](#), [vtpdetect](#), [wavdetect](#), [wrecon](#)

---

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/wtransform.html>  
Last modified: December 2006

