ewavelet

August 15, 2005

Abstract

The task **ewavelet** detects sources in an image using the mexican hat wavelet algorithm. The algorithm detects both point sources and extended sources provided that the source extent is not too large ($< 1/8$th the image size). The final source list will contain source positions, count rates, plus an indication of the source extent.

1 Instruments/Modes

Instrument	Mode
EPIC MOS:	IMAGING
EPIC PN:	IMAGING

2 Use

pipeline processing	yes
interactive analysis	yes

3 Description

The **ewavelet** tasks detects sources using wavelet transforms. For this task and its description we have made use of the *Chandra Detect User Guide* by Dobrzycki et al. ([1]) and the articles by Damiani et al. ([2],[3]), but the implementation will be new.

Wavelet transformations are an extension of Fourier transformations, but unlike the Fourier transformation functions (sines and cosines) wavelet functions have a well defined location in space. A wavelet function should have a zero normalization and satisfy the property:

$$W_{a,\sigma}(x) = \frac{1}{\sigma} W\Big(\frac{x-a}{\sigma}\Big), \tag{1}$$

σ is called the *scaling* or *dilation* parameter and a is the *translation* parameter. A good source about wavelets is Holschneider ([4]). For this task we have used the two-dimensional Mexican Hat (MH) function $W(\sigma, x, y)$, which can be derived from the two-dimensional Gaussian function

$$\phi(x, y, \sigma) = \frac{1}{2\pi\sigma^2} \exp\Big(-\frac{x^2+y^2}{2\sigma^2}\Big) \tag{2}$$

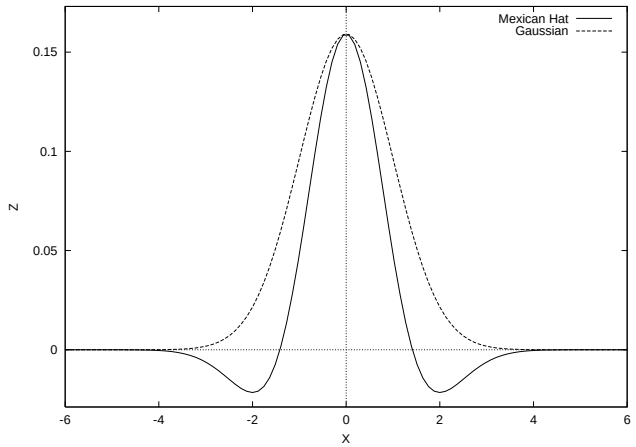

Figure 1: Solid line: The mexican hat wavelet function. Dashed line: a gaussian function with the same parameter σ.

using the following relationship:

$$W(\sigma, x, y) = [(x, y)\nabla + 2]\phi(x, y) \; = \frac{1}{2\pi\sigma^2}\Big[2 - \frac{x^2 + y^2}{\sigma^2}\Big] \exp\Big(-\frac{x^2 + y^2}{2\sigma^2}\Big). \tag{3}$$

The central part of the MH is positive and resembles a gaussian function (see Fig. 1). Outside the circle with axis length $\sqrt{2}\sigma$ the MH is negative. Since the image is convolved with the MH function the MH acts like a sort of sliding cell: the positive part being the source cell and the negative part being background area. From this it is clear that the wavelet scale should be smaller than the image itself in order to get a well defined image convolution. As a rule of thumb, the scale size should not exceed 1/8th of the image size.

Sources are detected by convolving the image with the MH function for a given scale parameter σ. Local maxima in the convolved image correspond to sources. The image is convolved using several scale parameters. In this implementation, like in most implementations, the scale size is increased for each convolution with a factor $\sqrt{2}$. Unlike the Chandra software we use circular symmetric wavelet functions. The reason is that allowing $\sigma_x \neq \sigma_y$, increases the number of convolutions, which would make the task considerably slower. Moreover, it would only increase the sensitivity for elongations in two very distinct directions, ignoring source position angles in between 0° and 90°.

If we assume that the source shape has a gaussian form we can analytically derive the value of the maximum correlation at a given wavelet scale σ. So the source is described by:

$$S(r) = \frac{N_{src}}{2\pi a^2} \exp(-\frac{r^2}{2a^2}) + b, \tag{4}$$

where we have used the symbol a instead of σ for the gaussian parameter, in order to set it apart from the wavelet scale parameter σ. We have here included a constant background term, b, which is the number of background photons per pixel. N_{src} is the total number of photons, or another suitable normalization (e.g. total countrate). The maximum of the convolved source occurs when the source position matches the wavelet position, this gives for the maximum of the convolution:

$$C_{max} = [W * S]_{max} = \frac{1}{2\pi\sigma^2}\frac{N_{src}}{2\pi a^2}\int_0^\infty (2 - \frac{r^2}{\sigma^2}) \exp\big(-\frac{r^2}{2\sigma^2}\big) \exp\big(-\frac{r^2}{2a^2}\big) 2\pi r dr = \frac{N_{src}}{2\pi a^2}\frac{2}{1 + (\sigma/a)^2}\Big(1 - \frac{1}{1 + (\sigma/a)^2}\Big) \tag{5}$$

C_{max} peaks for $a = \sigma$ with $C_{max} = N_{src}/4\pi a^2$. Note that the background does not appear in eq. (5, as the integral of the wavelet function amounts, per definition, to zero. This suggests that we can estimate source counts and countrate by comparing $C_{i,j}$ for several wavelet scales and picking the wavelet scale, σ_{max} for which C_{max} peaks. This gives an estimate for the source size $a = \sigma_{max}$ and consequently we can estimate the source normalization with $N_{src} = C_{max} 4\pi\sigma_{max}^2$. This aspect of the wavelet analysis (namely source extent estimation) is neglected by the Chandra software. In section 3.1 it will be shown how one can improve this estimate.

3.1 Countrate estimation

The wavelet detection routine is most and for all a *detection* routine. Countrates can, however, be estimated in a simple way, but with less accuracy than for example a maximum likelihood fitting method.

Countrates are estimated using a countrate image. This countrate image consists of the input image divided by the exposure map. Both the input image and the countrate image are convolved with the Mexican Hat wavelet. If a maximum is found in the convolved input image, a corresponding maximum in the convolved countrate image will be searched for. Multiplying this maximum correlation with $4\pi\sigma^2$ gives an estimate for the source countrate.

In a separate routine the best correlation for a given source for all different wavelet scales will be compared and the most appropriate wavelet scale will be used, in order to obtain the source countrate. In fact, we can do a little better than that: correlation values at two different scales can be used to infer a better estimate for the scale at which the correlation peaks. From eq. (5) it can be derived that the best estimate for the scale, σ', at which the wavelet correlation peaks is:

$$\sigma' = \sqrt{\frac{\frac{\sigma_1}{\sigma_2}\sigma_2^2 - \sqrt{\frac{C_1}{C_2}}\sigma_1^2}{\sqrt{\frac{C_1}{C_2}} - \frac{\sigma_1}{\sigma_2}}}, \tag{6}$$

σ_1, σ_2 are here the two wavelet scales at which the source was detected, and C_1 and C_2 are the corresponding wavelet correlation values.

Note that this algorithm is more like described by Damiani et al. ([2]), than the algorithm in the Chandra wavelet code.

To estimate the error in the number of counts we have to calculate the variance, and for the error in eq. (6) also the covariance, related to eq. (5). We assume here that the image to be analysed is a photon count image, i.e. the errors per pixel follow a poisson distribution. For a gaussian source shape we obtain for the covariance between wavelet scales σ_1 and σ_2:

$$\begin{aligned}
&[(W_1 W_2) * S]_{max} = \\
&\frac{N_{src}}{(2\pi)^3\sigma_1^2\sigma_2^2 a^2}\int_0^\infty \Big(2 - \frac{r^2}{\sigma_1^2}\Big)\Big(2 - \frac{r^2}{\sigma_2^2}\Big)\exp\big(-\frac{r^2}{2\sigma_1^2} - \frac{r^2}{2\sigma_2^2}\big)\Big(N_{src}\exp\big(-\frac{r^2}{2a^2}\big) + 2\pi a^2 b\Big)2\pi r dr = \\
&\quad \frac{N_{src}}{2\pi^2(a^2\sigma_1^2 + a^2\sigma_2^2 + \sigma_1^2\sigma_2^2)}\Big(2 - \frac{s^2}{\sigma_1^2} - \frac{s^2}{\sigma_2^2} + \frac{s^4}{\sigma_1^2\sigma_2^2}\Big) + \frac{4\sigma_1^2\sigma_2^2 b}{\pi(\sigma_1^2 + \sigma_2^2)^3},
\end{aligned} \tag{7}$$

where $s^2 = 2a^2\sigma_1^2\sigma_2^2/(\sigma_1^2\sigma_2^2 + \sigma_2^2 a^2 + a^2\sigma_1^2)$.

To calculate the variance we can simply set $\sigma_1 = \sigma_2 = \sigma$:

$$\begin{aligned}
&[W^2 * S]_{max} = \frac{1}{2\pi\sigma^2}\frac{1}{2\pi\sigma^2}\frac{1}{2\pi a^2}\int_0^\infty \Big(2 - \frac{r^2}{\sigma^2}\Big)^2\exp\big(-\frac{r^2}{\sigma^2}\big)\Big(N_{src}\exp\big(-\frac{r^2}{2a^2}\big) + 2\pi a^2 b\Big)2\pi r dr = \\
&\quad \frac{1}{2\pi\sigma^2}\frac{N_{src}}{2\pi a^2}\frac{1}{1 + \frac{1}{2}(\sigma/a)^2}\Big(2 - \frac{2}{1 + \frac{1}{2}(\sigma/a)^2} + \frac{1}{(1 + \frac{1}{2}(\sigma/a)^2)^2}\Big) + \frac{b}{2\pi\sigma^2}.
\end{aligned} \tag{8}$$

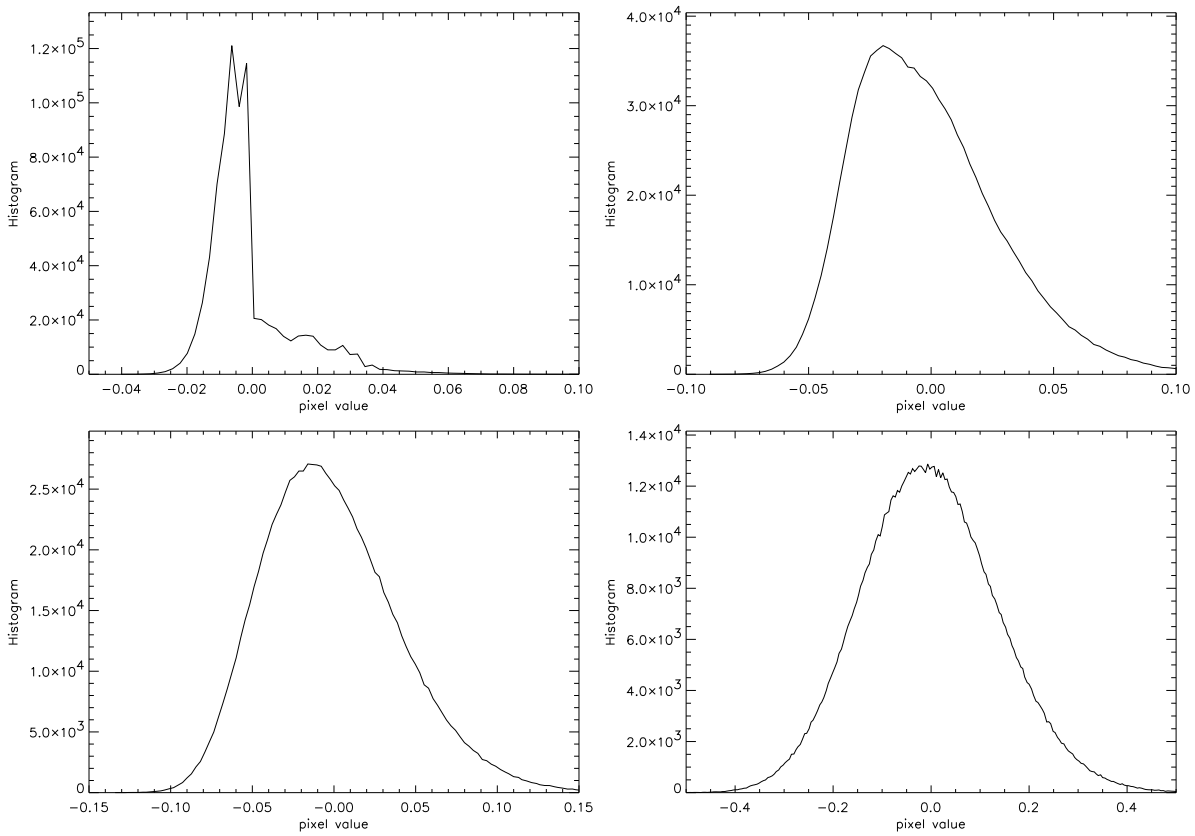

Figure 2: Histograms of wavelet convolved images. The wavelet scale was $2\sqrt{2}$ and the background value were respectively $b = 0.01/q \simeq 0.5$, $b = 0.05/q \simeq 2.5$, $b = 0.1/q \simeq 5.0$ and $b = 1/q \simeq 50$ (cf. Damiani et al. [2]).

The error in the correlation is the square root of this expression.

3.2 The detection threshold

In order to determine the significance of a detection, we have to calculate what the chance is that we find a peak in the wavelet convolved image in the absence of real sources. The fluctuations in the wavelet map is given by Eq. (8). For large number of (background) photons we may expect that the fluctuations in W approach a gaussian distribution, with standard deviation $\sqrt{\frac{b}{2\pi\sigma^2}}$:

$$P(C_{max} > C_0) = \frac{\sigma}{\sqrt{b}} \int_{C_0}^{\infty} \exp\Big(-\frac{1}{2}\frac{2\pi\sigma^2 C^2}{b}\Big) dC. \tag{9}$$

Because we used a different normalization constant in eq. (5), this equation is different from Damiani et al. ([2]) and Dobrzycki et al. ([1]). However, using the transformation $q = 2\pi\sigma^2 b$ and $C' = 2\pi\sigma^2$ we obtain:

$$P(C'_{max} > C'_0) = \frac{1}{\sqrt{2\pi q}} \int_{C'_0}^{\infty} \exp\Big(-\frac{1}{2}\frac{C'^2}{q}\Big) dC' = \frac{1}{2}\Big[1 - \mathrm{erf}\Big(\frac{C'}{\sqrt{2q}}\Big)\Big]. \tag{10}$$

The transformation is useful, as both Dobrzycki et al. ([1]) and Damiani et al. ([2]) have made Monte Carlo simulations to approximate the statistical behaviour in case the gaussian approximation is not valid. They observed that this is the case for $\log q < 3.25$. To determine significant thresholds for the case the

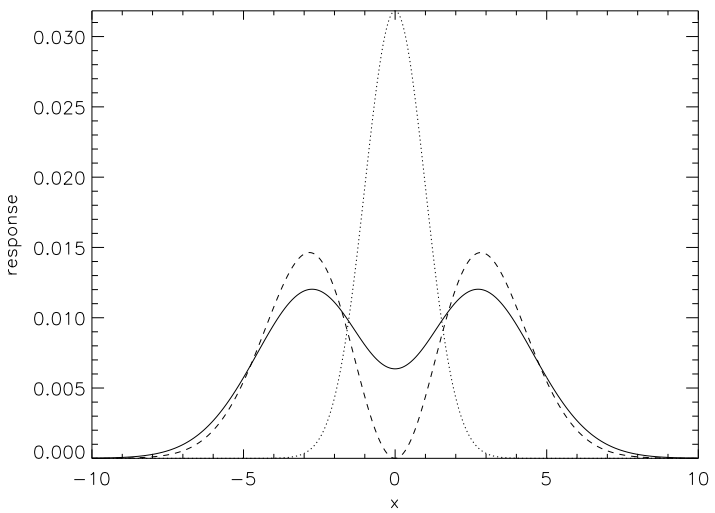

Figure 3: The response of a gaussian source (dotted line) to the convolution of the image with the bakcground smoothing function. The dashed line shows the function itself and the solid line shows the response to a gaussian ($\sigma_{bgd} = 2\sigma_{gauss}$). The gaussian has been multiplied by a factor 1/5.

gaussian approximation is not valid Damiani et al. ([2]) fitted the results of Monte Carlo simulations to the following formula:

$$C_0'(k,q) = C_0 2\pi\sigma^2 = k\sqrt{q} + (c_1 + c_2 k + c_3 k^2), \tag{11}$$

where C_0' and C_0 are the detection threshold for MH wavelets with and without normalization constants respectively, and k is the number of gaussian sigma's. The numerical constants were found to be: $c_1 = -0.2336$, $c_2 = 0.0354$ and $c_3 = 0.1830$. Clearly, this fit has the correct assymptotic behaviour.

Note that all the above apply to correlation values of individual pixels. In reality the probabilities to detect spurious sources above the background is a little bit smaller, as detections are defined as *local maxima* above the threshold. Furthermore, since we are only interested in maxima not in minima eq 11 refers to the "single tail" probability of a gaussian distribution. As an example, in a 1024×1024 image, $1363 = 0.0013 \times 1024^2$ pixels are expected above the 3σ threshold.

3.3 The convolution

From version 2.0 on **ewavelet** uses the fast fourier transform (FFT) to convolve images with the wavelet filter and the background images with other smoothing filters. For small wavelet scales direct convolutions are still used. The FFT routines are those of the external *fftw* package. Care is taken that the input images are only once transformed and stored, in order to avoid unnecessary calls to the FFT routines. The Fourier Transforms of the mexican hat function and the background smoothing functions are calculated directly using the analytical expressions for those functions (rather than using a FFT).

The direct convolution (small wavelet scales only) uses a window size of approximately 4.5σ. A smaller window size decreases the computational effort at the expense of the convolution accuracy. (Note that for a two dimensional gaussian function 3σ does not correspond to a fraction of 99.73%, but to 98.9%.)

3.4 Background estimation

The wavelet detection algorithm itself does not use any separate background estimate. However, for the determing the significance of a candidate source detection the background value at a given pixel coordinate has to be known.

At the moment the background is calculated by smearing the image with a function of the form

$r^2/\sigma^2 \exp(-0.5r^2/\sigma^2)$.

This function is related to the mexican hat function through its Fourier Transform. It has the nice property that it weighs more the outside ring, so that possible sources do not give too much of a strong signal at the source position. For the σ parameter I choose to use a value twice the size of the wavelet scale under investigation (so for every scale a different background map is generated).

After one iteration the input for the background estimate is the original image minus the sources, detected at scales smaller or equal to the wavelet scale under investigation. In this way sources with a large extent are modelled in the background map. This prevents spurious detections of small sources within the envelope of extended sources.

3.5 Avoiding spurious detections near edges in the image

The algorithm used to prevent the detection of spurious sources near sharp gradients in the exposure (e.g. chip edges) has been changed as of version **ewavelet** v3.3 .

The background map is convolved with the current wavelet filter. This convolved image should show similar features as the convolved input image, if no real sources were present. We only retain sources whose value of C_{max} exceeds the corresponding pixel value in the convolved background map by at least a factor of `edgethreshold`.

3.6 Creating reconstruction image and background image from an input source list

If the parameter `withinputlist` is set to "true", the reconstruction and/or background image will be created using source parameters from an input source list. The filename of the input list is specified by the parameter `inputlist`. Use parameters `makerecon` and `savebkgrnd` to specify which of the output images you want to create. Only sources originally detected at scales up to `maxscale` will be used for to calculate the images. The value of `maxscale` will also control the scale of the kernel used to filter the background image.

3.7 Examples

1. Run **ewavelet** on MOS1 image, saving reconstruction image and background image. Use all permitted convolution scales and a detection threshold of 4.0.

```
ewavelet imageset=M1_IMAGE_8000.FIT expmapset=M1_EXPMAP8000.FIT
         srclistset=ewavlist_m1.fits minscale=1 maxscale=32 threshold=4.0
         makerecon=yes recimageset=m1_rec.fits
         savebkgrnd=yes bkgrndset=m1_bkg.fits
```

2. Use **ewavelet** to create a reconstruction image and background image based on the existing ewavelet source list "my_ewavlist_m1.fits". Use `maxscale=32` in order to include sources detected at all scales. In this mode no source detection is performed.

```
ewavelet imageset=M1_IMAGE_8000.FIT expmapset=M1_EXPMAP8000.FIT
         withinputlist=yes inputlist=my_ewavlist_m1.fits
         makerecon=yes recimageset=my_m1_rec.fits
         savebkgrnd=yes bkgrndset=my_m1_bkg.fits
         maxscale=32
```

3.8 Implementation

This task is written in C++ and uses an image class containing the image data and the FITS parameters associated with the world coordinates. FFT routines are taken from the FFTW package, which is fast.

4 Parameters

This section documents the parameters recognized by this task (if any).

Parameter	Mand	Type	Default	Constraints
imageset	yes	string	`image.fits`	

Name of the input fits image.

srclistset	no	string	`ewavlist.fits`	

Output data set containing the wavelet source list.

expmapset	no	string	`expmap.fits`	

Name of the input fits image containing the exposure map.

useexpmap	no	boolean	true	

Use an external exposure map or not (in case no proper exposure map is available).

minscale	no	real	2.0	1.0–16.0

Minimum wavelet scale.

maxscale	no	real	8.0	2.0–32.0

Maximum wavelet scale.

threshold	no	real	5	3–7

Detection threshold in σ, only sources with a significance larger than this threshold will be stored. The choice of the threshold will also influence the parameters of sources above the threshold, since the background estimate depends on the number of detected sources.

edgethreshold	no	real	0.1	0.001–1.0

Controls the supression of spurious detection near sharp edges in the exposure mask. The exposure mask is convolved with the MH function and only regions, where the resulting image value is less than `edgethreshold`, are used for detection. Small values lead to better supression of spurious sources, but also real sources near edges might be missed. With `edgethreshold`=1.0, no sources will be rejected.

niter	no	integer	2	2–4

Number of iterations to be used.

makerecon	no	boolean	false	

Boolean controlling whether an image with detected sources represented by gaussians will be saved as a fits image.

recimageset	no	string	reconstruction.fits	

Name of output fits image containing the reconstructed input image. See also parameter makerecon.

savebkgrnd	no	boolean	false	

Boolean controlling whether the final background image will be saved as a fits image.

bkgrndset	no	string	background.fits	

Name of the fits output image containing the final background image.

withinputlist	no	boolean	false	

Boolean controlling whether reconstruction and/or background image will be derived from an input source list (parameter `inputlist`) . If set to "true", no source detection will be performed.

inputlist	no	string	ewavlist.fits	

If `withinputlist`="true", this file will be used as input source list.

5 Errors

This section documents warnings and errors generated by this task (if any). Note that warnings and errors can also be generated in the SAS infrastructure libraries, in which case they would not be documented here. Refer to the index of all errors and warnings available in the HTML version of the SAS documentation.

paramError *(error)*
A minimum wavelet scale was requested, larger than the maxium wavelet scale

inputImError *(error)*
The task encountered a serious error when trying to read the input image.

inputMapError *(error)*
The task encountered a serious error when trying to read the input exposure map.

reconError *(error)*
The task encountered problems making a model image, containing gaussians for all detected sources. This map is used to subtract sources from the input image in order to generate a better background image.

bgdCalc *(error)*
Error generating background map.

wavCalc *(error)*
Wavelet analysis error. Should be accompanied by warnings, which give more specific information.

srclistError *(error)*
Error generating final source list. No sources may have been found.

unsupportedDataType *(warning)*

corrective action: The task uses single floating point arrays to store images, doubles are automatically changed to single floating point values, but this may cause errors.

noExpMap *(warning)*
In some cases an exposure map may not be available, this is, however, not recommended and will result in this warning.
corrective action: A flat exposure map is made, using the `EXPOSURE` keyword of the input image

noMatch *(warning)*
One of the following keywords in the input image/exposure map did not match: `OBS_ID`,`EXP_ID` and/or `INSTRUMENT`.
corrective action:

reconError *(warning)*
This will useally mean that the source list is empty, so that a "reconstructed image" could not be made
corrective action: No reconstructed image will be written/or in some cases a proper background map cannot be made, which generates an error.

diffCoord *(warning)*
World coordinates of two images (e.g. exposure map and input image) are different
corrective action:

noSources *(warning)*
No source are present in the source list.
corrective action:

coordError *(warning)*
Signifies an error in the image world coordinates system.
corrective action: The columns `RA` and `DEC` will remain empty in the final source list

6 Input Files

1. PPPs product (from task **evselect**): EPIC FITS images (one broad band image)
2. PPS product (from task **eexpmap**): Exposure image (for the broad band image)

7 Output Files

1. EPIC **ewavelet** source list. The source list will contain the following columns: `SRC_NUM`, `IMAX`, `JMAX`, `X_IMA`, `Y_IMA`, `X_IMA_ERR`, `Y_IMA_ERR`, `RA`,`DEC`, `RADEC_ERR`, `SCTS`, `SCTS_ERR`, `WSCALE`, `WCORR`, `XYCORR`, `RATE`, `EXTENT`, `EXT_ERR`, `EXP_MAP` and `BG_MAP`.

   Most of them can also be found in the tasks **eboxdetect** and **emldetect**. New are:

   (a) `IMAX`, `JMAX`: pixels (integer) where the maximum correlation value was found (`X_IMA`, `Y_IMA` give a refined meausurement of the position)

   (b) `WSCALE`: the wavelet scale at which the source was found. This gives an indication of the source extent (in gaussian σ). Note that if the value is comparable to the point spread function one is probably dealing with a point source.

   (c) `WCORR`: the maximum correlation value ($C_{i,j}$ in this description).

   (d) `EXTENT` and `EXT_ERR`: gives the source extent and its error (not correcting for the spread function). It is defined as $\sigma' \sqrt{8 \ln 2}$ (see eq. 6). This corresponds to the FWHM for a gaussian.

   (e) `XYCORR`: Correlation value between X and Y. Large values are an indication of an elongated source with a position angle around 45°.

2. FITS image containing the final background map generated and used by **ewavelet**.
3. A fits image with the "reconstructed image": the detected sources are represented by gaussians with σ =wavelet scale and normalization the number of counts found by the algorithm. This image will make it easy to compare results of the task with the input image.

8 Algorithm

The wavelet detection involves the following steps:

- Read the input image and the exposure map.
- Generate the countrate image from the input image and exposure map.
- Generate the initial background map by scaling the exposure map in such a way that the background map contains as many "counts" as the input image.

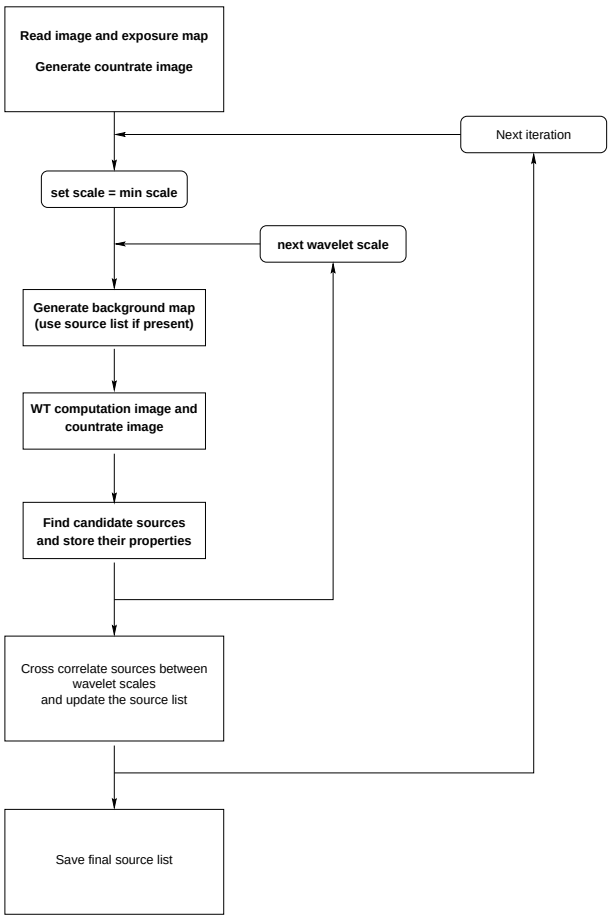

Figure 4: Block diagram of the wavelet source detection algorithm.

- Do the following for several wavelet scales:
  1. generate a background map for this particular scale
  2. convolve image with the MH wavelet function
  3. find local maxima in the convolved image
  4. Check if the local maximum is significant
  5. If the maximum is significant, add the point to the source list and calculate the number of counts from the wavelet correlation value ($C_{i,j}$)
- Determine if some sources were detected at several wavelets scales. If so, choose the wavelet scale with the highest local correlation value.
- Use the detections of the same source at several wavelet scale to get some information on the source properties.
- Make a new iteration or generate the final source list.

9 Comments

10 Future developments

- FFT routines may still need some performance optimization, as not all the functionality of the FFTW package is currently used.
- Galactic coordinates are not yet calculated, but this can change in the near future.
- Currently preference is given to small sources over extended sources if they are present near the same coordinates. It would be nice to have a good selection criterium to include also extended sources in the final source list if it is clear that the signal is not caused by the small scale source.
- More work has to be done on the error estimates of source parameters.
- Countrates are estimated assuming that the source shape is a gaussian. Unresolved sources have the shape of the PSF. This means that for point-like source we can in future calibrate the true countrates versus the wavelet countrates, for small wavelet scales. However, this involves a lot of calibration work and/or numerical simulations.

References

[1] K. Glotfelty et al. A. Dobrzycki, H. Ebeling. CHANDRA detect 1.0 User Guide. Technical Report Version 1.0, Chandra Science Center, February 1999.

[2] G. Micela S. Sciortino F. Damiani, A. Maggio. A method based on wavelet transforms for source detectionin photon-counting detector images. I Theory and general properties. *Ap. J.*, 483:350–369, 1997.

[3] G. Micela S. Sciortino F. Damiani, A. Maggio. A method based on wavelet transforms for source detectionin photon-counting detector images. II Application to ROSAT PSPC images. *ApJ.*, 483:370–389, 1997.

[4] M. Holschneider. *Wavelets: An Analysis Tool.* Oxford University Press, 1995.