DETECT/MAXLIK - technical documentation

DOE/MPE Feb 2000

1. Overview

DETECT/MAXLIK and its buddy COMPUTE/UPPER_LIMITS are the most refined source detection algorithms. Due to the high computational effort required for each source, the algorithm does not search the whole field-of-view for sources, but depends on a list of input source positions (e.g. a MERGE/SOURCE merged source list or an user-supplied list) and a good background model (background maps from CREATE/BG_IMAGE).

The ML algorithm defines a circle (extraction circle, cut radius) around each input source position, and extracts all photons within this circle from the events table. By optimizing the source parameters, ML finds the source parameters which gives the most likely match to the photon distribution within the extraction circle. The optimized source parameters are source position, source counts (vignetted, i.e. source counts which did actually arrive in the detector), and source extent. Sources are modeled as either point sources or as extended sources with simple Gaussian profiles.

The likelihood calculations are photon-based — i.e. each photon is treated individually taking the photon position and the point spread function (dependent on photon amplitude and arrival position in the detector) into account. This allows the application of ML to variable sources, and to observations where the optical axis of the detector is moving with respect to the sky (as is the case for both ROSAT pointing and ROSAT survey modes). The restrictions of this method are discussed in section 4.1.

Unlike the SASS ML, the EXSAS ML algorithm does not introduce a vignetting factor directly into the calculation of the likelihoods. Therefore the resulting source counts are per definition vignetted. A vignetting correction is applied afterwards to correct the source counts to unvignetted source counts. However, this correction may or may not be right, depending on the circumstances of the observation. This is discussed in section 4.9.

2. Parameter file

DETECT/MAXLIK is called with a parameter file, containing the names of the input/output files and other parameters relevant for steering the process. Parameters which are seldom modified are only accessible when creating the parameter file with the *expert* option (extended parameters). Otherwise they are properly defaulted.

Example of a template parameter file for ROSAT PSPC:

```
!
! parameter file for      DETECT/MAXLIK          TASK: DETEMAX
!
 MISSION,DETECTOR   ROSAT,PSPC                            ! mission and detector
 OBS_MODE           POINTING                              ! pointing or survey
 INPUT_DATASET      events.tbl                            ! name of events table
 AMPLITUDE_LOW         8                                  ! lower amplitude bound
 AMPLITUDE_HIGH      256                                  ! upper amplitude bound
 CUT_RADIUS               2.50                            ! cut radius in FWHM
 OFF_AX_MIN               0.00                            ! min off-axis angle
 OFF_AX_MAX              57.00                            ! max off-axis angle
 ML_THR                     10.00                         ! threshold of -ln(p)
 START_POSITION     merged_list                           ! merged_list / user_list
 MERGED_LIST        mplst.tbl                             ! merged source list
 USER_LIST          uslst.tbl                             ! user source list
 FIT_POSITION       free                                  ! fit source position
 BG_IMAGE_1         bacmp1.bdf                            ! smoothed bg image
 ML_LIST_ACCEP      solst.tbl                             ! list of accept. sources
 ML_LIST_ALL        mllst.tbl                             ! list of all sources
!
!  extended list of parameters
!
 MIN_ML_EXTENT               0.00                         ! min -ln(p) of extent
 MIN_ML_EXIST                 0.00                        ! min -ln(p) of existence
 MAX_ML_EXIST            1.0E20                           ! max -ln(p) of existence
 EXCLUDE_SIGMA       999999.00                            ! thresh.f.source exclude
 CLUSTER_THR                 0.00                         ! cluster threshold
 EXTENT_THR                  0.00                         ! extent threshold
 FLAG_BGEXTR         1                                    ! flag for bg extraction
 FULL_OUT_FLAG       F                                    ! full output flag
!
!  fixed and instrumental parameters (do not change)
!
 NUM_IMAGES           1                                   ! number of images
```

```
VIGNET_TABLE       EXSAS_CAL:vignet_pspc.tbl           ! vignetting table
BG_SCALE               1.0000                          ! scale factor bg image
MAX_ITER                  100                          !*max number iterations
ACCURACY_ITER          0.000010000000                  !*iteration accuracy
DEFAULT_AMPL           30                              ! ampl for FWHM calcul.
DEFAULT_FWHM            0                              ! default FWHM [arcsec]
AMPL_COLUMN        AMPL                                ! ampl column label
```

MISSION,DETECTOR: mission and detector identifier, separated by comma. Internal variable *EXPDET*.

OBS_MODE: keywords **POINTING** or **SURVEY** Internal variable *OBSMOD*.

INPUT_DATASET: name of events table. The events table should be small-map-sorted (filetype PEM). That means it must not be time-sorted, amplitude-sorted, etc. Internal variable *INPUT0*.

AMPLITUDE_LOW: lower amplitude limit. Must match the amplitude range given in the background map (descriptor **AMP_SEL**, or **RAW_SEL** for HRI). Internal variable *IECHL(1)*.

AMPLITUDE_HIGH: upper amplitude limit. Must match the amplitude range given in the background map (descriptor **AMP_SEL**, or **RAW_SEL** for HRI). Internal variable *IECHH(1)*.

CUT_RADIUS: determines radius of extraction circle for each source. The radius of the extraction circle is
CUT_RADIUS * FWHM
where FWHM is the full-width-half-maximum of the PSF at the mean source position. Internal variable *SP_CUT*. This parameter is by default active, but can be overridden by setting a general (source- or PSF-independent) cut radius with extended parameter **DEFAULT_FWHM** (see below).

OFF_AX_MIN: lower limit for off-axis angle (arcmin). Only input sources with a mean off-axis angle larger than this limit are fitted. Rarely used. Internal variable *SP_EPSL*.

OFF_AX_MAX: upper limit for off-axis angle (arcmin). Only input sources with a mean off-axis angle smaller than this limit are fitted. Should be set to prevent sources at the extreme edge of the field-of-view from being fitted. Internal variable *SP_EPSU*.

ML_THR: Maximum likelihood threshold. Only sources with a ML result greater than this threshold are written to the output source list *solst*. Internal variable *SP_THR*.

START_POSITION: defines the type of the input source list. The two possible settings are (**m**)**erged_list** (take a merged source list from MERGE/SOURCES as input) or (**u**)**ser_list** (take a user-supplied source list with RA,DEC coordinates as input). Internal variable *DT-MODE*.

MERGED_LIST: filename of merged list. Active if **START_POSITION** = **merged_list**. Internal variable *MPLSX*.

USER_LIST: filename of user-supplied list. Active if **START_POSITION** = **user_list**. Internal variable *USLST*.

FIT_POSITION: Flag with the options **free** or **fixed**. If set to **fixed**, the source position is kept fixed during the ML fit. Internal variable *POSFLA*.

BG_IMAGE_1: filename of background map. Internal variable *BACMP(1)*.

ML_LIST_ACCEP: filename of output source list containing all sources with a likelihood of existence above **ML_THR**. Internal variable *SOLST*.

ML_LIST_ALL: filename of output source list with all sources, disregarding whether the fit was successful or whether the likelihood was above the threshold. Internal variable *MLLST*.

MIN_ML_EXTENT: Threshold for fitting source extent. If the fitted maximum likelihood of source existence is above this threshold, the algorithm tries to fit a source extent. Internal variable *SP_SNRE*.

MIN_ML_EXIST: Rarely used parameter. The likelihood analysis is only applied to sources with input likelihood between **MIN_ML_EXIST** and **MAX_ML_EXIST**. This input likelihood is read from the input merged list. Internal variable *SP_SNRL*.

MAX_ML_EXIST: Rarely used parameter. The likelihood analysis is only applied to sources with input likelihood between **MIN_ML_EXIST** and **MAX_ML_EXIST**. This input likelihood is read from the input merged list. Internal variable *SP_SNRU*.

EXCLUDE_SIGMA: Number of σ's for another source within the extraction circle, above which the source exclusion mechanism is triggered. A large value of **EXCLUDE_SIGMA** like the default of 999999 effectively disables the problematic source exclusion mechanism (see section 4.4). Internal variable *SP_EXC*.

CLUSTER_THR: Threshold for cluster size. Only sources with a smaller cluster size in the input source lists are included in the ML analysis. Since there is no cluster analysis in the preceding source detection steps, this parameter is practically inactive. Internal variable *SP_THCL*.

EXTENT_THR: Threshold for sources extent. Only sources with a smaller extent in the input source lists are included in the ML analysis. Since there is no extent analysis in the preceding source detection steps, this parameter is practically inactive. Internal variable *SP_THEX*.

FLAG_BGEXTR: determines the method for background extraction. If set to **0**, the old method for background extraction is used (extract background only at input source position). If set to **1** (default), the new method is activated (extract background for each photon individually). For details see section 4.5.

FULL_OUT_FLAG: full-output-flag. If set to **T**, some useful files for debugging are created (see section 3). Internal variable *SP_FOF*.

NUM_IMAGES: number of energy ranges. Since DETECT/MAXLIK can only handle one energy range at once, **NUM_IMAGES** must stay set to **1**. Any other value crashes the program. Internal variable *ICT_ECH*.

VIGNET_TABLE: filename of table containing the vignetting factors dependent on photon amplitude and off-axis angle. Detector-specific. Internal variable *VIGTAB*.

BG_SCALE: scale factor of the background image. Some years ago the background maps were still stored in INTEGER*4 format. To ensure sufficient accuracy, the background maps were multiplied with a scale factor which was usually 100. With the introduction of REAL*4 format for the background maps, this scale factor became obsolete. For compatibility reasons it is still accessible here and defaulted to 1. Newer background maps have a descriptor **BG_SCALE**, overriding the **BG_SCALE** factor specified in the parameter file. Internal variable *BSCAL*.

MAX_ITER: Maximum number of fit iterations for both the position/flux fit and for the extent fit. Internal variable *SP_ITER*.

ACCURACY_ITER: Precision limit for both the position/flux fit and for the extent fit. The iterations are stopped when the likelihoods converge within this value.

DEFAULT_AMPL: Default amplitude for the PSF-related calculations. Internal variable *DEFAMPL*.

DEFAULT_FWHM: Default FWHM in units of arcsec. If set to a non-zero, positive, value, this constant FWHM supersedes the source-dependent calculation of the FWHM. For pointed observations it should be left at zero, to trigger the individual FWHM calculation. But for e.g. survey observations the individual FWHM calculation does yield wrong values, and so **DEFAULT_FWHM** must be set. For the ROSAT survey it is by default set to 60 arcsec, so that every source is cut out with the same radius of
CUT_RADIUS * 60 arcsec.
Internal variable *DEFFWHM*.

AMPL_COLUMN: label of the amplitude column in the events table. Usually the amplitude column is labelled *AMPL*. For HRI, the amplitudes must be taken from column *RAW_AMPL*, so parameter **AMPL_COLUMN** must be set to **RAW_AMPL**. Internal variable *AMPLCOL*.

Template parameter files for each of the possible missions can be found in the *detemap* source code directory.

The default name for the parameter file is *detemax.epf*. Other names are possible though. If at execution time no parameter file is found, a default one for ROSAT,PSPC will be created.

To ensure, that each parameter is defaulted properly, even if missing in the parameter file, the routine *sppari.for* reads the first two parameters **MISSION,DETECTOR** and **OBS_MODE**. Based on this, the other parameters are initialized in routine *spini.for*. After that, *sppari.for* looks for the other keywords in the parameter file and overwrites the default values for the found parameters.

3. I/O

The primary input file is the parameter file, either specified on the command line or by default *detemax.epf*. The other I/O files are specified in the parameter file.

The parameter file is read with routine *sppari.for*.
The reading of the background map is done in routine *spprep.for*.
The opening of the input source lists is done in *espin.for*. The source-by-source input of the source lists is handled in *rmplsx.for* (merged list) or *ruslst.for* (user-supplied list). Closing is done in *exspcl.for*.
Creation of the output source list is done also in *espin.for*, the source-by-source output is handled in routine *espout.for*, and the closing is done in *exspcl.for*.

Input files

BG_IMAGE_1: The background maps must have the same amplitude range as specified in the parameter file as parameters *AMPLITUDE_LOW* and *AMPLITUDE_HIGH*. If not, the program stops with an error message. The background map must also have the usual EXSAS header.

MERGED_LIST: Usually the output of MERGE/SOURCES. It must have the usual EXSAS header, and the descriptors *START*, *STEP*, *NPIX*, *SKY_CEN_X*, *SKY_CEN_Y*, and *SKY_PIX_SIZE* must be present, referring to the image used for the map and local detection. These descriptors must match the corresponding descriptors of the background map **BG_IMAGE_1**, otherwise the program stops with an error message.
Alternatively also the output lists of DETECT/LOCAL or DETECT/MAP could be specified here. In this case column *COFL* must be set to 1 by hand. *COFL* is the continuation flag identifying unmerged sources.

USER_LIST: Must contain the columns *RA_DEG* and *DEC_DEG* with equatorial coordinates in units of fractional degrees. The necessary information for converting these coordinates to sky coordinates (nominal pointing, etc.) is read from the background map (see routine *uslstr.for*).

VIGNET_TABLE: vignetting table to be used. Must contain a vignetting factor grid, dependent on the photon amplitude (not photon energy!) and the off-axis angle. If such a table is not available (like for ROSAT WFC), parameter **VIGNET_TABLE** must be set to **none**. In this case, subroutine *spextr.for* bypasses the reading of the vignetting table and sets the vignetting factors to 1.

Output files

ML_LIST_ACCEP (*SOLST*): Source list with all detected sources with a likelihood of existence above the ML threshold **ML_THR**. The format and content of the source lists is described in section 4.10.

ML_LIST_ALL (*MLLST*): Source list with all sources. The format and content of the source lists is described in section 4.10.

Optional output files

These files are created only if the full-output flag **FULL_OUT_FLAG** has been set to **T**. They are used for debugging issues only.

diagnosis.tbl: Contains some useful numbers for each source, describing the immediate output of the fit procedures. Should be at least partially self-explaining.

usphot.tbl: Table containing all extracted photons for the first source. Useful for checking whether the photon selection and extraction worked properly.

mlbld.bdf: Image with ML values for detected sources.

4. Maximum Likelihood source detection

4.1 Maximum Likelihood equation

The following is a derivation of the maximum likelihood equation as used in EXSAS. Although I do not use the same terminology of the derivation given in Cruddace, Hasinger, & Schmitt

(Astronomy From Large Databases, Garching 1988, eds. Murtagh & Heck), the following is essentially identical to their derivation.

Initial assumptions

Assume a celestial source with S_0 total counts arriving in front of the detector during an observation. Of those S_0 source counts, S counts are actually recorded in the detector. Whether the photons are actually recorded or not depends on the mirror vignetting V and detector efficiency ϵ. Since at this stage we are only interested in S, we can forget about V and ϵ for now.

The spatial resolution of the detector is described by the point spread function PSF, which is a two-dimensional function normalized to 1. The shape of the point spread function is variable over the detector plane, and depends moreover on the photon amplitude. Due to the spacecraft wobble or slew, the source position moves over the detector plane during the observation interval. Thus the effective PSF will have a different shape for each of the observed photons.

The detector has a background (both internal and sky background). which is described by the number of background counts per sky pixel B_i (i denoting the background B for a given sky pixel i). B is usually obtained by fitting a bicubic spline to the observed data after blending out the sources. Note that B is thus the number of actually detected background counts (after vignetting).

Derivation of the exact Likelihood function

In the following, the index k always denotes a photon k, the index i denotes a sky pixel i.

Let's look at a single sky pixel i and a single photon k, which has an amplitude a, and is detected at time t. The probability p_{ik} that photon k hits pixel i is given by:

$$p_{ik} = psf(a,t) \tag{1}$$

where $psf(a,t)$ is the integral of the normalized point spread function $PSF(a,t)$ over the area of sky pixel i, valid at the time t and for amplitude a.

Assuming the normalized amplitude distributions $A(a,t)$ and the normalized distribution of arrival times $T(t)$ is known, the probability that any photon at any time hits pixel i is

$$p_i = \int\limits_a \int\limits_t psf_{a,t}\, A(a,t)\, T(t)\, da\, dt \tag{2}$$

p_i is thus something like an time- and spectrum-averaged PSF value for pixel i.

It should be noted, that p_i is impossible to calculate exactly, since $A(a,t)$ (spectrum and spectral variability) and $T(t)$ (source intrinsic lightcurve and varying vignetting) are a priori unknown to ML. Anyhow, let's continue.

Over the whole observation the **expected** number of photons recorded in sky pixel i is

$$N_i = B_i + S \cdot p_i \tag{3}$$

The number of events actually recorded in pixel i is n_i. Counts are independent events and underly the Poisson statistics. The probability for n_i events in a Poissonian distribution with expectation value N_i is given by

$$P(N_i, n_i) = e^{-N_i} \cdot \frac{N_i^{n_i}}{n_i!} \tag{4}$$

The probability that one gets exactly the observed distribution of events for all pixels i is given by the product of the individual probabilities:

$$P = \prod_i e^{-N_i} \cdot \frac{N_i^{n_i}}{n_i!} \tag{5}$$

Expressing the probability as the likelihood (defined as $L = -2 \ln P$) one gets

$$\begin{aligned} L &= 2 \sum_i \left(N_i - n_i \cdot \ln N_i + \ln \left(n_i! \right) \right) \\ &= 2 \sum_i N_i - 2 \sum_i \left(n_i \cdot \ln N_i \right) + 2 \sum_i \ln \left(n_i! \right) \end{aligned} \tag{6}$$

Replacing N_i using eq. 3 yields:

$$L_S = 2 \sum_i B_i + 2 \cdot S \sum_i p_i - 2 \sum_i n_i \cdot \ln \left(B_i + S \cdot p_i \right) + 2 \sum_i \ln \left(n_i! \right) \tag{7}$$

The corresponding likelihood for no source ($S = 0$) is

$$L_0 = 2 \sum_i B_i - 2 \sum_i n_i \cdot \ln \left(B_i \right) + 2 \sum_i \ln \left(n_i! \right) \tag{8}$$

The likelihood of existence of a source is given as the difference between L_S and L_0:

$$\begin{aligned} L &= L_S - L_0 \\ &= 2 \cdot S \sum_i p_i - 2 \sum_i n_i \cdot \ln \left(B_i + S \cdot p_i \right) + 2 \sum_i n_i \cdot \ln \left(B_i \right) \end{aligned} \tag{9}$$

The sum over pixels i in the second and third term can be replaced with a sum over the detected photons k (because $\sum_i n_i = \sum_k$):

$$L = 2 \cdot S \sum_i p_i - 2 \sum_k \ln \left(B_i + S \cdot p_i \right) + 2 \sum_k \ln \left(B_i \right) \tag{10}$$

Likelihood function in EXSAS

Up to now the derivation was exact. Now let's look at the approximations and simplifications that EXSAS uses to make the whole thing computable.

Assumption 1: The sum of the p_i over the pixels within the extraction radius equals 1. This is a bad approximation when the radius of the extraction circle is chosen too small and approaches the FWHM of the point spread functions or the extent of the source, or when parts of the extraction circle are blended out by the software due to interfering other sources.

$$L = 2 \cdot S - 2 \sum_k \ln\left(B_i + S \cdot p_i\right) + 2 \sum_k \ln\left(B_i\right) \tag{11}$$

Assumption 2: The mean probability p_i can be replaced by the individual probabilities $p_{ik} = psf_{ik}$ for the individual detected photons. psf_ik is the normalized point spread function valid for photon k and for the exact photon arrival point on the detector plane i. The approximation is good if the sky and detector pixel sizes are smaller than the typical point spread function, and if the distribution of p_{ik} is a reasonably well-behaved statistical sample of the average value p_i. The former is no problem, and the latter is usually also more or less fulfilled except for weird cases, where for example the period of a variable source comes near to the wobble period.

$$L = 2 \cdot S - 2 \sum_k \ln\left(B_i + S \cdot psf_{ik}\right) + 2 \sum_k \ln\left(B_i\right) \tag{12}$$

This is the equation for the likelihood. To find the best-fitting source counts, position, and extent, the likelihood is maximized by iteration. For this iteration, the last, constant term and the factor 2 can be skipped for simplicity:

$$L' = S - \sum_k \ln\left(B_i + S \cdot psf_{ik}\right) \tag{13}$$

Note, that the best-fitting source counts obtained by maximizing this likelihood equation are still vignetted counts. To obtain the unvignetted source flux S_0, the vignetting correction has to be applied.
SASS uses a slightly different approach in their ML version, since they include the vignetting in their ML equation to obtain S_0 directly. However, this approach requires one additional approximation is thus more unstable for ill-behaved sources or large vignetting variations.

4.2 Source models

Sources are modelled either as point sources or as extended sources. All sources are folded with the point spread function. For reasons of computability, both point spread functions and source extents are treated as Gaussians, so that the Gaussian σ's of point spread function and source extent simply add to a modified point spread function:

$$psf'_{ik} = e^{-\left(\frac{d}{(\sigma_{psf} + \sigma_{ext})}\right)^2}$$

σ_{psf} is the σ of the Gaussian approximation to the instrument-induced point spread function, dependent on the photon amplitude a and the photon's location on the detector.

σ_{ext} is the source extent to be fitted. It is zero for a point source.
d is the linear distance between the photons arrival point and the nominal source position to be fitted.

In reality neither the point spread function nor the shape of any extended source are pure Gaussians, so there is another approximation involved here. The poorer the approximation is, the less reliable the fit results will be.

4.3 Input positions

Input positions are read from the input source lists. This can either be a merged list from MERGE/SOURCES, or a user-supplied list.

From a merged list, the columns **X_POS** and **Y_POS**, containing image coordinates, are evaluated and converted to sky coordinates using the image parameters of the background map. In hindsight it would have been better to read the sky coordinates directly, since the image coordinates depend on the image sizes and require additional crosschecking and computing effort.

The user-supplied list is supposed to contain RA and DEC values. These RA and DEC values are deprojected to sky coordinates using the relevant info stored in the background map (e.g. nominal pointing direction, coordinate system used, method of projection) since this is the only image read by DETECT/MAXLIK. Therefore it is essential that the background map coordinate and projection conventions matches those valid for the events table — i.e. the background map should ideally stem from the same events table from which the photons for ML are extracted.

The input positions determine the center of the extraction circle. If the input positions are bad, the source is not centered in the extraction circle. A small offset is no problem for the fit. If the offset is on the order of the cut radius, two potential problems arise. First, on one side of the circle there will be too much area of pure background, unneccessarily increasing the fit errors. Second, and more serious, assumption 1 from section 4.1 may be violated, leading to fit results more or less in error. Another secondary, but nasty effect is described in the next section.
To avoid these problems, it has been proposed, to check the input against the fit output source position. If they differ too much, the ML fit should be repeated with a better centered extraction circle. Such an algorithm would for sure be useful, but it was never implemented.

4.4 Photon extraction

Extraction circle

The radius of the extraction circle (cut radius) is determined by parameter **CUT_RADIUS** (internal variable *SP_CUT*) in the parameter file, multiplied with the FWHM of the PSF at the average off axis angle ϵ and for a default amplitude given by parameter *DEFAMPL*:

$$cutradius = SP\_CUT * FWHM(\epsilon, DEFAMPL)$$

DEFAMPL is adjustable via parameter **DEFAULT_AMPL** in the parameter file.
The average off axis angle is determined by computing the distance of the input source position from the image center. This is clearly wrong for off-centered observations, merged fields, and

especially survey data where the off-axis angle variation of a source is extreme. In these cases parameter **DEFAULT_FWHM** in the parameter should be set to a positive value, specifying a constant FWHM in units of arcsec, which overrides the individual FWHM calculations:

$$cutradius = SP\_CUT * DEFFWHM$$

For the ROSAT survey, *DEFFWHM* is defaulted to 60 arcsec.

Photon extraction

Once the extraction circle is determined, the events table is searched for photons whose sky coordinates fall into the extraction radius (in subroutine *spextr.for*). For each photon, the sky coordinates *sky_x*,*sky_y*, the detector coordinates *det_x*,*det_y*, and the photon amplitude a is read and stored into internal arrays. All of these values are required later for calculating the psf_{ik}.

The label of the table column in the events table, from which the amplitude is read, is defined by parameter **AMPL_COLUMN** in the parameter file. This was necessary since for the ROSAT HRI the amplitude must be read from column **RAW_AMPL** instead of **AMPL**.

Avoiding array overflows

The arrays holding the photon positions and amplitudes, are confined to a size of 4000 elements (constant *IPA_SP* in common block *CSPIO*). Bright sources easily exceed this limit. To account for this, a count of all photons within the extraction circle is done (*coucir.for*) before the actual extraction. If the ratio $FRAACC = \frac{totalcounts}{4000}$ exceeds 1, the photon extraction routine in *spextr.for* is adjusted so that it reads only each $FRAACC$th photon ($FRAACC$ is a REAL value!). This ensures that the number of extracted photons does not exceed 4000 while being uniformly distributed over their positions in the events table avoiding any bias effects.

Of course the background read for each photon and the fitted source counts and count errors must later be corrected by this fraction.

Source exclusion mechanism

After the extraction is done, the algorithm *spexcl.for* looks for contamination by secondary nearby sources within the extraction circle. To do this, the extraction circle is divided into an inner and an outer ring. The outer ring is subdivided in 8 sectors, and the photons in each sector are summed up. If one of the sector counts is significantly higher than the others, the sector and the two adjacent sectors is marked bad. All photons within the bad sectors are then deleted from the photon lists. The fraction of not excluded area is calculated and used later to correct fitted source counts and count errors. It does also show up in the final source table as column *FA*.

The criterium for bad sectors is a sector count which is a certain number of statistical σ's higher than the average sector count. This 'certain number' is given by parameter **EXCLUDE_SIGMA** (internal variable *SP_EXC*) in the parameter file.

The source exclusion mechanism has a nasty problem. If one or more sectors are excluded, assumption 1 and 2 of section 4.1 are seriously violated. The fit results are at best unreliable.

Another problem of this mechanism was the phenomenon we called 'self exclusion'. A source (preferrably a bright one) with a bad input position causing it to sit off-center in the extraction circle, often triggered the source exclusion mechanism without any other source present, making things go from bad to worse.

Therefore parameter **EXCLUDE_SIGMA** has been defaulted to the very high threshold of 999999, effectively disabling the source exclusion mechanism. However, there is still a warning on screen and also a warning remark in the output source list, if one of the sectors deviates by more than 5 σ from the others. An appearance of this warning should caution the user that there may be another source within the extraction radius, or that possibly the sources is strongly decentered, or may be the background is very uneven in this region.

4.5 Background map

The background values for each photon (appearing as B_i in the maximum likelihood equation, see section 4.1) are read from the background map (subroutine *getbkg.for*). There is an old method and a new method which has been implemented only recently.

The old method (triggered by setting parameter **FLAG_BGEXTR** in the parameter file to **0**) reads only the background at the input source position and assigns the same value to each photon. This method did not only discard information, it had also often difficulties where the background map near the source was rough or uneven.

The new method (**FLAG_BGEXTR** = **1**, default) reads the background for each photon individually using the photon's sky position, thus avoiding the problems with the old method and improving the ML fit quality. Simultaneously, there is an average background calculated. This average is never used in the ML algorithm, but it serves as output for the final source list.

If there were events falling out of the borders of the background map, a warning is issued, and the background for these events is set to the background value at the input source position.

If the input source position itself falls outside the borders of the background map, the source is flagged bad and the ML fit for this source is not executed. In the *mllst* output list such sources are flagged with the text 'no bgr avail' in the *REMARKS* column.

4.6 Maximum Likelihood fit

description of flux and position iteration, to be done...

4.7 Source extent

description of source extent iteration, to be done...

4.8 Error estimation

description of error estimation algorithm, to be done...

4.9 Vignetting correction

math behind vignetting correction, to be done...

4.10 Output source list

format and content of output lists, to be done...

5. Code status

The code is fully ftools-compatible. Executables for both ftools and MIDAS environment are available and running. The memory for the two internal image arrays is allocated dynamically (*spmemo.for*), and the routines are rewritten to handle any image sizes up to hardware limits. Images can also be non-square, and the sky center is not required to coincide with the image center. The current version supports all ROSAT experiments. Lots of the code are shared with the related command COMPUTE/UPPER.

The ftools version is still untested!

What remains to be done:

- For the ML detection an iterative approach would be useful for sources with bad input positions. I.e., if the found position differs too much from the input position, a second (or even third) run might be done with the found position as new input position, with the new extraction circle determined by this improved position. Should be implemented optionally, and steerable by the user.
- The input and output routines (in particular the reading and writing of source lists) are still a mess. They could be simplified greatly.
- Throw out unneccessary or inactive stuff like parameters **CLUSTER_THR**, **EXTENT_THR** or **NUM_IMAGES** (see sect. 2).
- Amplitude ranges could be read from the background map instead of having to specify them in the parameter file.
- Full documentation of the error calculation stuff.

6. Implementation of other experiments

Required calibration files:

- Vignetting table containing a grid with vignetting factors dependent on photon amplitude (not photon energy!) and off-axis angle. Must be readable by subroutine *getvig.for* in library *xmakepro*. See for example *vignet_pspc.tbl* in the EXSAS calibration area. If none is available, set parameter **VIGNET_TABLE** in the parameter file templates to **none**. In this case the program sets all vignetting factors to 1.

Software changes:

- Prepare template parameter files with proper defaults, with special attention to parameters **AMPLITUDE_LOW**, **AMPLITUDE_HIGH**, **OFF_AX_MAX**, **VIGNET_TABLE**, **DEFAULT_AMPL** and **DEFAULT_FWHM**.
- Edit subroutine *spini.for* and add the default values for the new experiment. The defaults should be the same as in the template parameter files.
- Extend point-spread-function routine *fwhm_g.for* in library *xpsfnew* to deliver approximate Gaussian FWHMs also for the new experiment's PSF — or provide a dummy constant FWHM via extended parameter **DEFAULT_FWHM**.