

# aperture

David Makovoz, 30/01/2006

Version 1.0

## Table of Contents

aperture .....	1
1 Overview .....	2
1.1 Input Image Requirements .....	2
2 aperture .....	2
2.1 Input .....	2
2.2 Processing .....	4
2.3 Output Table .....	6
3 aperture.pl .....	7
4 References .....	8

## 1 Overview

Included in MOPEX is a program for computing aperture photometry - *aperture*. It is used in the script *apex.pl* and *apex\_1frame.pl* as part of computing point source fluxes. It can be also used as a stand-alone program to compute aperture photometry for a list of points (not necessarily point sources) specified in the input table in a single input image. Another option is using it to compute aperture photometry for a single point in a set of images specified in an input list. This capability is encapsulated in the perl script *aperture.pl*. This document describes the input requirements, processing, and output of *aperture* and *aperture.pl*.

### 1.1 Input Image Requirements

The input images have to be in the FITS format. The data type of the images are single precision floating point numbers, i.e. BUNIT = -32. The following keywords are required in the headers of the input files for the software to work: BITPIX, NAXIS, NAXIS1, NAXIS2, CRVAL1, CRVAL2, CRPIX1, CRPIX2, CTYPE1, CTYPE2. CD-Matrix elements or/and CDELT1, CDELT2, CROTA2 should be also present.

## 2 aperture

### 2.1 Input

The program *aperture* takes its input either on the command line or from the namelist (configuration file). The command line takes precedence over the namelist. All the keywords are case-sensitive. The namelist **does not have** the default location in the cdf subdirectory. If the namelist is located in the cdf subdirectory it has to be specified explicitly:

```
aperture -n cdf/aperture.nl
```

Table 1 contains all the input parameters, their namelist names and command line switches.

Parameter Name		Required	Default
Namelist	Command line		
Input_Image_Filename	-i	y	-
Input_List_Filename	-il	y, if SinglePoint_X(Y), are not given	-
Output_Table_Filename	-o	y	-
N_Apertures	-napers	y	-
Aperture_Radius_#	-aper#	y	-
N_Masks	-nmask	n	0
Mask_Value_#	-mv#	y, if N_Masks > 0	-

Mask_Filename_#	-m#	y, if N_Masks > 0	-
X_Column_Name	-xname	n	'x'
Y_Column_Name	-yname	n	'y'
TransformationType	-tr	n	'NoTransform'
FIF_FileName	-fif	n	-
SinglePoint_X	-x	n	-
SinglePoint_Y	-y	n	-
Use_Refined_Pointing	-refpoint	n	0
Mode	-mode	n	'image'
Use_Annulus	-ann	n	0
Annulus_Compute_Type	-comp	n	'median'
Inner_Radius	-innr	y, if Use_Annulus=1	-
Outer_Radius	-outr	y, if Use_Annulus=1	-
Min_Number_Pixels	-minpix	n	9
HighPrecision	-high	n	0

**Table 1.** The list of namelist names and the command line switches for the input parameters for the program *aperture*.

The aperture photometry is measured in the input fits file (*Input\_Image\_FileName*). The aperture photometry is measured at a set of points listed in the input list (*Input\_List\_FileName*). The list is an IPAC table which should have two required columns with the coordinates of the points. The column names can be changed from their default by the user (*X\_Column\_Name*, *Y\_Column\_Name*).

The program can compute aperture photometry for a single point. In order to do so the coordinates of the point are given as input to the program (*SinglePoint\_X*, *SinglePoint\_Y*). In this case the input list name is ignored.

The coordinates could be either pixel coordinates in an image or the sky coordinates. The coordinate transformation from the coordinates in the table to the pixel coordinates in the input image should be performed. The type of coordinate transformation (*TransformationType*) depends on the coordinates in the input list. If the point coordinates are pixel coordinate in the input file, then the coordinate transformation is set to 'NoTransform'. If the point coordinates are pixel coordinates in a different fits file, then the coordinate transformation is set to 'PlaneToPlane'. In this case the pointing information of the fits file should be given in the FIF table (*FIF\_FileName*). If the point coordinates are sky coordinates, then the coordinate transformation is set to 'SkyToPlane'.

The number of apertures is given by the namelist keyword *N\_Apertures*. The number of aperture keywords should match the value of *N\_Apertures* and they should be numbered in order. For example, if *N\_Apertures* = 2 is set in the namelist, or *-napers* 2 on the command line, then there should be two keywords *Aperture\_Radius\_1* and *Aperture\_Radius\_2* set in the namelist, or alternatively *-aper1* and *-aper2* on the command line.

If the image is in surface brightness units (FITS keyword *BUNIT* = *MJy/Sr* or *microJy/square\_arcsec*), it computes the results in the units of  $\mu\text{Jy}$ . Otherwise the output is in the units of the input image.

Optionally an unlimited number of mask fits files can be specified along with the fatal bit pattern for each mask. The number of `Mask_Filename_#` and entries should match the value of `N_Masks`. E.g if `N_Masks = 2` is set in the namelist, or `-nmask 2` on the command line, then there should be two keywords `Mask_Filename_1` and `Mask_Filename_2` set in the namelist, or alternatively `-m1` and `-m2` on the command line. In addition there should be two keywords `Mask_Value_1` and `Mask_Value_2` set in the namelist, or alternatively `-mv1` and `-mv2` on the command line.

If `HighPrecision` is set, the output is written with 9 significant digits (formatting `%9.7e`). Otherwise it is written with 5 significant digits (formatting `%5.3e`).

The refined pointing keywords `RARFND`, `DECRFND`, `CT2RFND` are read optionally instead of the standard keywords `CRVAL1`, `CRVAL2`, `CROTA2`, if `Use_Refined_Pointing` is set.

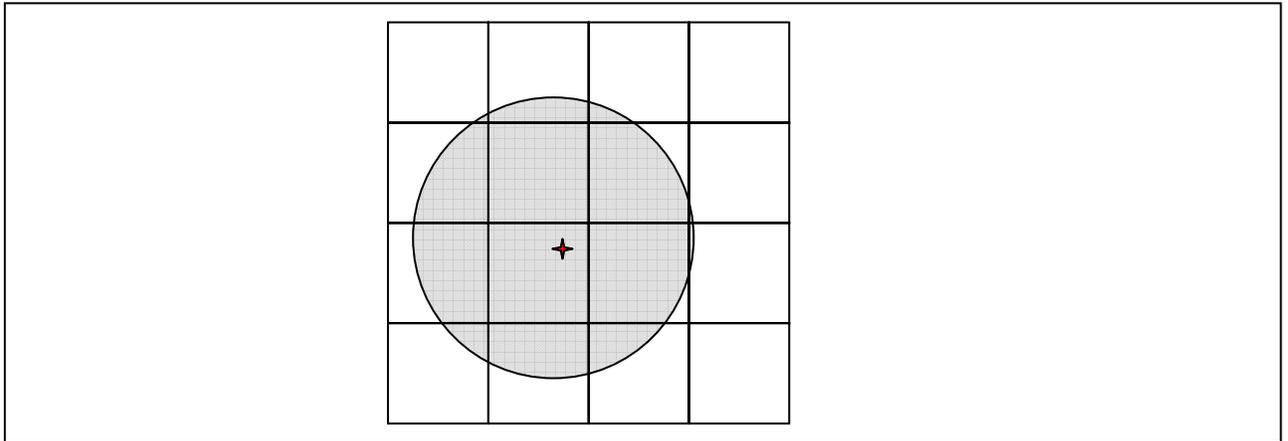
## 2.2 Processing

By default the aperture photometry AP is computed as

$$AP = \sum_i a_i \cdot I_i,$$

which is the sum of the products of the pixel values  $I_i$  weighted with the exact area overlap  $a_i$ . If `Mode` is set to 'sigma', the pixel contributions are added in quadratures:

$$AP = \sqrt{\sum_i (a_i \cdot I_i)^2}$$



**Figure 1.** A circle of a specified radius centered on the point source (red) is overlaid on the image and aperture photometry is computed by summing the pixel values weighted by the overlap area.

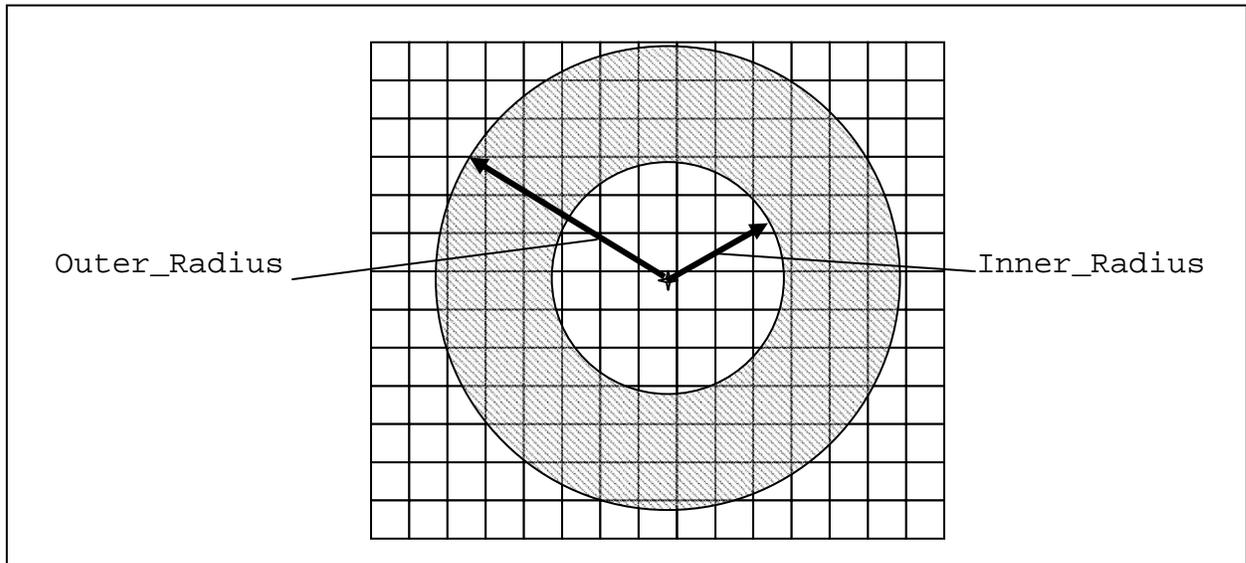
Optionally, if `Use_Annulus` is set, the background  $B$  can be estimated and subtracted from the aperture photometry for each point. It is estimated within an annulus of a user specified size.

$B = \text{Operation}(\text{Pixels with } R \geq \text{Inner\_Radius} \text{ and } R \leq \text{Outer\_Radius})$ .  $R$  is the distance from the point position to the center of the pixel, fractional pixel computation is not done, pixel is either in or out.

**Operation** is defined by the `Annulus_Compute_Type` keyword and can be of three types: 'mean', 'median'(default), and 'mode'.

`Min_Number_Pixels` is the minimum number of good (non-NaN) pixels in the annulus to compute the background. If the number of good pixels < `Min_Number_Pixels`, then the background is not estimated and the aperture photometry is written in the output table without background subtraction. The background subtracted aperture photometry is computed as

$$AP = \sum_i a_i \cdot I_i - B \sum_i a_i$$



**Figure 2.** The geometry of the annulus used for background estimation.

For MOPEX releases 030106 and later, the pixel area is computed taking the Spitzer distortions into account, if such are present in the fits headers.

### 2.3 Sample Running Options

One can specify all the input parameters on the command line:

```
aperture -i mosaic.fits -il detect.tbl -o aperture.tbl -xname RA -yname Dec
-napers 1 -aper1 6 -tr SkyToPlane
```

Alternatively, as was mentioned above, one can run the program with only the namelist on the command line:

```
aperture -n cdf/aperture.nl
```

or a combination of the namelist and command line options:

```
aperture -n cdf/aperture.n -i mosaic_minback.fits -o aperture_minback.tbl
```

Below is a sample namelist.

```
&APERTURE
  Comment = 'Namelist file for APERTURE',
  Input_Image_Filename = 'mosaic.fits',
  Input_List_Filename = 'detect.tbl',
```

```

Output_Table_Filename = 'aperture.tbl',
X_Column_Name = 'RA',
Y_Column_Name = 'Dec',
TransformationType = 'NoTransform',
N_Apertures = 3,
Aperture_Radius_1 = 6.0,
Aperture_Radius_2 = 10.0,
Aperture_Radius_3 = 15.0,
Use_Annulus = 1,
Min_Number_Pixels = 10,
Annulus_Compute_Type = 'mode',
Inner_Radius = 15,
Outer_Radius = 25,
&END

```

## 2.4 Output Table

All the input parameters are written in the header of the output table. The columns of the output table are listed in Table 2.

#	Column Name	Description
1	srcid	The index of the point
2	x	The x coordinate of the point in the input image
3	y	The y coordinate of the point in the input image
4...n+3	aperture#	N_Apertures columns giving the aperture photometry
n+4...2n+3	bad_pix#	N_Apertures columns giving unusable portion of the aperture
2n+4	bckgrnd/pxl	Annulus background per pixel $B$ (if Use_Annulus = 1 was set)

**Table 2.** The columns of the output table. Here  $n$  is the number of apertures.

Below is a sample output table.

```

\char comment = Output from APERTURE, version 2.00
\char Date-Time = Wed Oct 12 10:10:34 2005
\char comment = dimage library version 4.00
\char Mode = image
\int Use_Annulus = 1
\int N_Apertures = 1
\float Aperture_Radius_1 = 3.000000
\char Input_Image_Filename = IRAC/5598208/ch1/apex/Combine/mosaic.fits
\char Input_List_Filename = IRAC/5598208/ch1/apex/extract_raw.tbl
\char Input_List_Filename = IRAC/5598208/ch1/apex/extract_raw.tbl
\int Min_Number_Pixels = 10
\float Outer_Radius = 15.000000
\float Inner_Radius = 10.000000
\char Annulus_Compute_Type = mode
\char Output_Table_Filename = IRAC/5598208/ch1/apex/aperture.tbl
\float Conversion Factor = 33.778720
|          srcid|          x|          y|  aperture1|          bad_pix1 |          bckgrnd/pxl|

```

```
|i          |r          |r          |r          |r          |r          |
...          1          5.030e+01          5.963e+00          3.991e+00          0.000e+00          2.292e-01
```

### 3 aperture.pl

One can use the program *aperture* in order to compute the aperture photometry for a single point in a set of images. The functionality is encapsulated by the perl script *aperture.pl*. The input files are specified outside of the aperture namelist block either in the namelist or on the command line as shown in .

Input File	Default Name	Namelist name	Command line option	Required
Namelist *	aperture.nl	N/A	-n	y
List of input images	-	IMAGE_STACK_FILE_NAME	-I	y
Permanently damaged pixels mask image	-	PMASK_FILE_NAME	-M	n
List of DCE status mask images	-	DCE STATUS MASK LIST	-d	n
List of outlier Rmask images	-	RMASK_LIST	-R	n

**Table 3.** Input to the script *aperture.pl*.

Below is a sample namelist.

```
IMAGE_STACK_FILE_NAME = Long_Ch1ImageList.txt
OUTPUT_DIR = aperture_ch1_input_pl
```

```
&APERTURE
  Comment = 'Namelist file for APERTURE',
  SinglePoint_X = 281.832350,
  SinglePoint_Y = 68.279313,
  X_Column_Name = 'RA',
  Y_Column_Name = 'Dec',
  TransformationType = 'SkyToPlane',
  N_Apertures = 1,
  Aperture_Radius_1 = 5,
  Use_Refined_Pointing = 1,
  Use_Annulus = 1,
  Min_Number_Pixels = 10,
  Annulus_Compute_Type = 'median',
  Inner_Radius = 10,
  Outer_Radius = 15,
&END
```

The script goes through the list of input images. For those images that contain the input coordinates the aperture photometry is computed. The name and the flux are written in the output table. Two sets of values of the average aperture and the standard deviation are computed. The first set - `average_aperture0`, `sigma_aperture0` - is computed using all the images. The second set - `average_good_aperture0`, `sigma_good_aperture0` - is computed using the images which have no missing pixels within the aperture, i.e. `bad_pix0 = 0`.

The output table named `aperture.tbl` is written in the output directory, which is specified in the namelist (`OUTPUT_DIR`) or on the command line (`-O`) with the default being the running directory (`.`).

Below is a sample output table produced by `aperture.pl`.

```
\int Use_Annulus = 1
\int N_Apertures = 1
\float Aperture_Radius_1 = 5.000000
\int Min_Number_Pixels = 10
\float Outer_Radius = 15.000000
\float Inner_Radius = 10.000000
\char Annulus_Compute_Type = mode
\float RA = 2.8183234e+02
\float Dec = 6.8279312e+01
\char Comment = pi 2.8183234e+02 6.8279312e+01
\char mosaic/Interp/interp_bcd_fp_0.fits
\char mosaic/Interp/interp_bcd_fp_1.fits
...
\float average_aperture0 = 203146.529795918
\float sigma_aperture0 = 593.912795678604
\float average_good_aperture0 = 203146.529795918
\float sigma_good_aperture0 = 593.912795678604
|image_number |aperture0      |bad_pix0      |
|i            |f              |f              |
              0   2.02417e+05  0.000e+00
              1   1.94163e+05  0.000e+00
...
```

## 4 References

All of the documents referenced in this document can be obtained from the Spitzer Science Center website, <http://ssc.spitzer.caltech.edu/postbcd/documentation.html>.

1. Spitzer\_Apex