

FASTPHOT: a very simple and quick IDL PSF-fitting routine

Matthieu Béthermin, Morgan Cousin and Hervé Dole
<http://www.ias.u-psud.fr/irgalaxies/>

12 mai 2010

1 Introduction

The simplest method to measure the flux of a source is the aperture photometry. This method is very efficient for isolated point sources. But, at far-infrared and submillimeter wavelengths, the angular resolution of the instruments are usually limited, and the background galaxies are seen as blended point sources. In this case, aperture photometry is not efficient and biased. Another method, PSF fitting photometry, allows a simultaneously fit of a PSF profile on the sources. This method is really efficient to measure the flux of very close sources.

Many routines use PSF fitting photometry among which IRAF/allstar, Starfinder, Convphot. These routines are in general complex to use and slow. FASTPHOT is optimized for prior extraction (the position of the sources is known) and is very fast and simple. It is based on a simple formalism described in this document.

This program could be updated in the future. Your comments, suggestions or modified version are welcome.

2 Credits

If you use FASTPHOT for a published work, please cite Bethermin, Dole, Cousin and Bavouzet (2010, in press, A&A, astro-ph :1003.0833).

3 How to run the code ?

3.1 FASTPHOT

FASTPHOT is an IDL/GDL library and its path must be declared in the !path variable. It is thus very easy to install. To test the installation, you can run fastphot_test in your IDL/GDL terminal.

If you fit a little field with typically less than 2000 sources, you should use basic FASTPHOT routine.

```
fastphot, map, psf, noise_map, X, Y, flux, Fprior = Fprior, map_clean = map_clean
CALLING SEQUENCE: fastphot, map, psf, noise_map, X, Y, flux, $
                   Fprior = Fprior, map_clean = map_clean, $
                   eflux = eflux, cov = cov, mu = mu, emu = emu
```

INPUTS:

MAP: Array containing the map of the studied field (all the good pixel must be >-99998).

If bad pixel, put -99999. value

PSF: Array containing the PSF. It must be a $(2N+1)*(2N+1)$ image (in the unit of the map) for a source with flux = 1 (in output unit).

NOISE_MAP: Array containing the standard deviation of the noise. (same size as map)

X: Vector containing the x coordinates of the sources. (float if decimal coordinates or not)

Y: Vector containing the y coordinates of the sources. (float if decimal coordinates or not)

KEYWORD PARAMETERS:

FPRIOR: Estimation of the flux of the fitted sources
(accelerate the fit)

MAP_CLEAN: Map cleaned from fitted sources.
(useful to check the results of a fit)

EXACT: Invert exactly the matrix (do not use conjugate
gradient method)

EFLUX: Vector containing the uncertainties on the output fluxes (the unit is given by the PSF).

COV: Covariance matrix of the output fluxes.

Warning: EFLUX and COV suppose a purely instrumental non-correlated noise!

The last line and column are associated to mu (the constant background) if mu is not fixed.

MU: fitted background level in unit of the map.

EMU: Uncertainty on mu.

FIXED_MU: if you want to fix manually the background level.

OUTPUTS:

FLUX: flux of the fitted sources (the unit is given by the PSF).

3.2 FASTPHOT_AREA

If your field is larger, you can use FASTPHOT_AREA to perform the fit faster. This routine cut the field in several area . Each area is fitted separately. To avoid edge effect, additional sources near the selected area are also fitted, but the result are not saved. A good choice for the size of this additional edge is typically 5 to 10 PSF FWHM.

CALLING SEQUENCE: fastphot_area, map, psf, noise_map, subdimx, subdimy, edgex, edgey, X, Y, flux, \$
Fprior = Fprior, map_clean = map_clean, eflux = eflux, \$

INPUTS:

MAP: Array containing the map of the studied field (all
the good pixel must be >-99998).
If bad pixel, put -99999. value

PSF: Array containing the PSF. It must be the $(2N+1)*(2N+1)$
image (in the unit of the map) for a source with flux = 1
(in output unit).

NOISE_MAP: Array containing the standard deviation of the
noise. (same size as map)

SUBDIMX: Dimension on x direction of the subextracted map

SUBDIMY: Dimension on y direction of the subextracted map

EDGEX: Additional edge in x fitted with the submap (the flux of
the sources on the edge are not saved).
Must be about 10 PSF FWHM.

EDGEY: same thing in y

X: Vector containing the x coordinates of the sources (float if decimal coordinates or not).

Y: Vector containing the y coordinates of the sources (float if decimal coordinates or not).

KEYWORD PARAMETERS:

FPRIOR: Estimation of the flux of the fitted sources
(accelerate the fit)

MAP_CLEAN: Map cleaned from fitted sources. (useful to check the results of a fit)

EXACT: Invert exactly the matrix (do not use conjugate
gradient method)

EFLUX: Vector containing the uncertainties on the output fluxes (the unit is given by the PSF).

OUTPUTS:

FLUX: flux of the fitted sources (the unit is given by the PSF).

4 Formalism of FASTPHOT

FASTPHOT formalism assumes that an extragalactic map is composed of :

- Point sources with the same PSF at known positions ;
- A random gaussian noise.

The map can be written as an array :

$$m = \sum_{k=1}^{N_s} S_k PSF_{x_k, y_k} + \sigma + \mu, \quad (1)$$

where m is the map array, N_s the number of sources to fit, S_k the flux of the k -th source, PSF_{x_k, y_k} is an array of the same size as the map which contains the response of a unit-flux source centered on the coordinates (x_k, y_k) , σ a random gaussian noise, and μ a constant background.

We define S as the vector of the S_k . The likelihood of a given map is :

$$L(m|S) = \prod_{pixels} C(n) \times \exp\left[-\frac{\left(m - \sum_{i=1}^{N_{sources}} PSF_{x_i, y_i} \times S_i - \mu\right)^2}{2n^2}\right], \quad (2)$$

where $C(n)$ is a normalization constant which depends only on n , the standard deviation of the noise. The logarithm of the likelihood is then :

$$-\log(L) = \sum_{pixels} \frac{\left(m - \sum_{i=1}^{N_{sources}} PSF_{x_i, y_i} \times S_i - \mu\right)^2}{2n^2} + \sum_{pixel} \log(C(n)) \quad (3)$$

The values of S and μ which maximize the likelihood verifies :

$$\forall S_k, \frac{\partial(-\log(L))}{\partial S_k} = 0 \quad \& \quad \frac{\partial(-\log(L))}{\partial \mu} = 0 \quad (4)$$

and consequently,

$$\forall S_k, \sum_{pixels} \frac{-PSF_{x_k, y_k} \left(m - \sum_{i=1}^{N_{sources}} PSF_{x_i, y_i} \times S_i - \mu\right)}{n^2} = 0 \quad \& \quad \sum_{pixels} \frac{m - \sum_{i=1}^{N_{sources}} PSF_{x_i, y_i} \times S_i - \mu}{n^2} = 0 \quad (5)$$

We can rewrite it as :

$$\forall S_k, \sum_{pixels} \frac{\mu \times PSF_{x_k, y_k}}{n^2} + \sum_{i=1}^{N_{sources}} S_i \times \sum_{pixels} \frac{PSF_{x_k, y_k} \times PSF_{x_i, y_i}}{n^2} = \sum_{pixels} \frac{PSF_{x_k, y_k} \times m}{n^2} \quad (6)$$

and

$$\sum_{pixels} \frac{\mu}{n^2} + \sum_{i=1}^{N_{sources}} S_i \times \sum_{pixels} \frac{PSF_{x_k, y_k}}{n^2} = \sum_{pixels} \frac{m}{n^2}. \quad (7)$$

It is equivalent to the $(N_{sources} + 1) \times (N_{sources} + 1)$ matrix equation :

$$A \cdot X = B \quad (8)$$

where X is a vector containing $[S_1, S_1, \dots, S_{N_{sources}}, \mu]$ and where A and B are defined by :

$$\forall i \in [1, N_{sources}], \forall j \in [1, N_{sources}], A = (a_{ij}) = \sum_{pixels} \frac{PSF_{x_i, y_i} \times PSF_{x_j, y_j}}{n^2} \quad (9)$$

$$\forall i \in [1, N_{sources}], (a_{i(N_{sources}+1)}) = (a_{(N_{sources}+1)i}) = \sum_{pixels} \frac{PSF_{x_i, y_i}}{n^2} \quad (10)$$

$$(a_{(N_{sources}+1)(N_{sources}+1)}) = \sum_{pixels} \frac{1}{n^2} \quad (11)$$

$$\forall i \in [1, N_{sources}], B = (b_i) = \sum_{pixels} \frac{PSF_{x_i, y_i} \times m}{n^2} \quad (12)$$

$$B = (b_{N_{sources}+1}) = \sum_{pixels} \frac{m}{n^2} \quad (13)$$

The matrix A is symmetrical and thus can be inverted. The maximum likelihood estimator for S is then :

$$S = A^{-1}B \quad (14)$$

The simultaneous PSF-fitting of a catalog of sources at known position can be reduce to an inversion of a $(N_{sources} + 1) \times (N_{sources} + 1)$ matrix.

5 Uncertainties

The Fisher matrix F at the best-fit S of the problem is :

$$F(S) = (f_{mn})(X) = -E\left(\frac{\partial^2 \log(L)}{\partial X_m \partial X_n} | X\right) \quad (15)$$

where $E(X|S)$ is the expected value of a random variable X knowing S. We use the results of the equation 7 :

$$\frac{\partial(\log(L))}{\partial S_n} = - \sum_{i=1}^{N_{sources}} S_i \times \sum_{pixels} \frac{PSF_{x_n, y_n} \times PSF_{x_i, y_i}}{n^2} + \sum_{pixels} \frac{PSF_{x_n, y_n} \times m}{n^2} \quad (16)$$

And then :

$$\frac{\partial^2(\log(L))}{\partial S_m \partial S_n} = - \sum_{pixels} \frac{PSF_{x_n, y_n} \times PSF_{x_m, y_m}}{n^2} = (a_{nm}). \quad (17)$$

We can also show

$$\frac{\partial^2(\log(L))}{\partial S_n \partial \mu} = (a_{n(N_{sources}+1)}) \quad \& \quad \frac{\partial^2(\log(L))}{\partial \mu \partial \mu} = (a_{(N_{sources}+1)(N_{sources}+1)}) \quad (18)$$

A is thus the Fisher matrix of the problem. The variances on the estimated flux are the diagonal terms of the inverse of the A matrix, because the covariance matrix is the inverse of Fisher matrix ($Cov = F^{-1}$) :

$$Var(S_k) = (A^{-1})_{kk}. \quad (19)$$

6 FASTPHOT programmation

FASTPHOT was developed to run quickly. We consequently optimized the speed of computation of A and B matrix, and the inversion of the A matrix. In general, the PSF is known only for scales smaller than the maps. The (a_{ij}) are computed only if the i-th and j-th sources are separated by less than the size of the PSF image. We use a linear interpolation of the PSF of the j-th sources to take into account the fraction of pixels. If the uncertainty are not required, we use a conjugate gradient algorithm to solve the equation 8 quickly.

7 Reliability of the uncertainties given by FASTPHOT

The FASTPHOT formalism supposes that the positions of all the sources are listed in the prior catalog. In the reality, the faint sources are not in this catalog and behave like a correlated confusion noise. In some case, this noise is larger than the instrumental noise. FASTPHOT supposes that the instrumental noise is independent between two pixels. The uncertainties of FASTPHOT could be wrong if the noise is correlated. BEWARE : the estimated noise will be underestimated if the confusion dominates the noise budget.

A simple way to test the photometric accuracy is to add few sources of known fluxes at random positions on the original map and in the input catalog and perform a new extraction. It must be done several times to obtain significant statistics. You can then compare the input and output values.