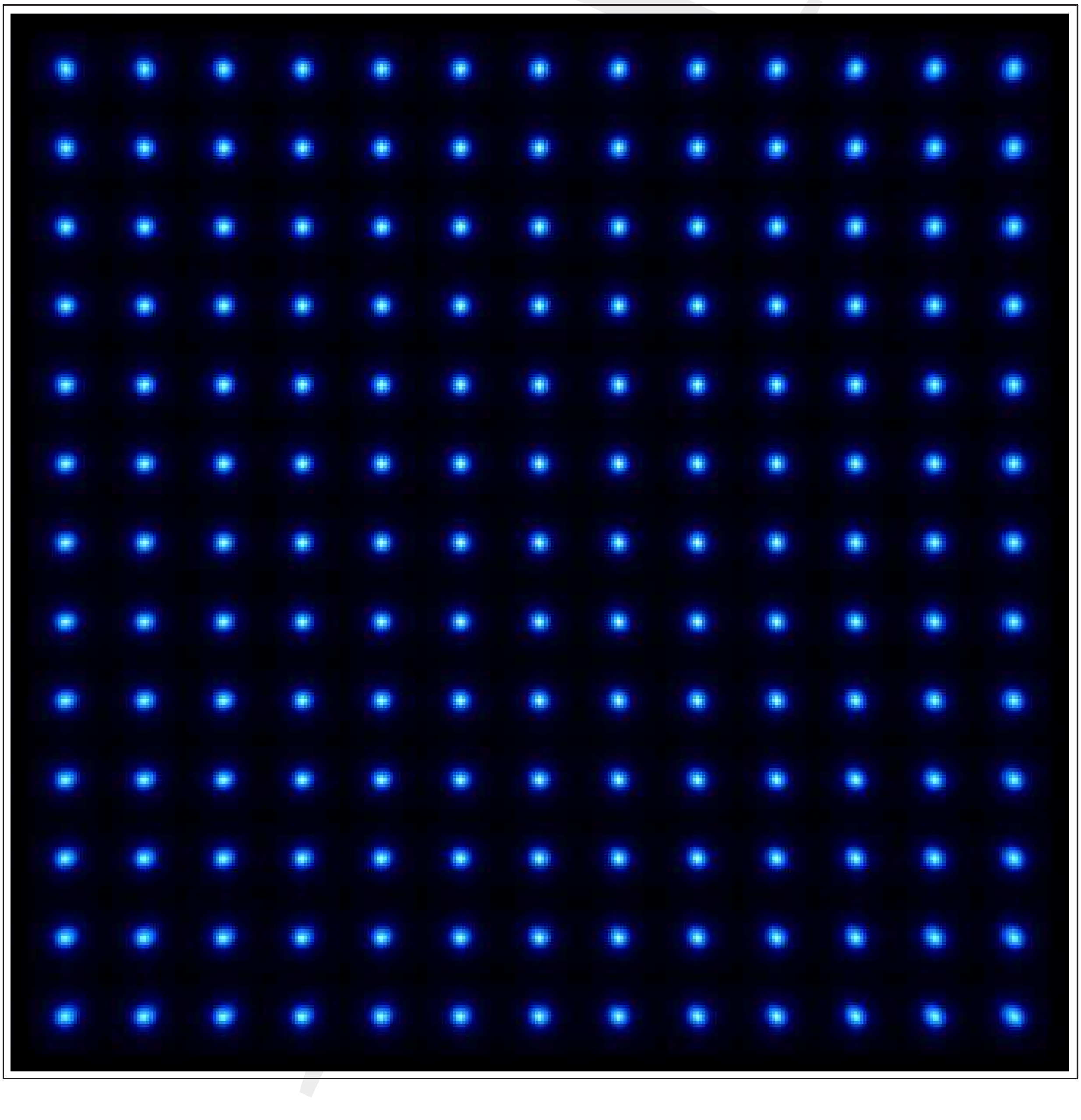
PSFEx
*v*3.9

User's guide

Emmanuel BERTIN
Institut d'Astrophysique de Paris

November 8, 2010



Contents

1 What is PSFEx?

PSFEx (PSF Extractor) is a computer program that extracts precise models of the Point Spread Functions (PSFs) from images processed by SExtractor[1] and measures the quality of images. The generated PSF models can be used for model-fitting photometry or morphological analyses. The main features of PSFEx are:

- Modelling of any arbitrary non-parametric or parametric, bandwidth-limited, PSF.
- Reconstruction of PSF from undersampled images using super-resolution on the pixel basis, the Gauss-Laguerre basis or a user-provided vector basis.
- Modelling of PSF variations as a polynomial function of position in image, any SExtractor measurement, or any numerical FITS parameter.
- Tracking of hidden PSF dependencies using Principal Component Analysis.
- Computation of PSF homogenisation kernels (to convert variable instrumental PSFs to constant round Moffat profiles).
- Automatic selection of point sources.
- Compatibility with SExtractor FITS or Multi-Extension-FITS catalogue format in input,
- XML VOTable-compliant output of meta-data.
- XSLT filter sheet provided for convenient access to metadata from a regular web browser.

2 Skeptical Sam's questions

Skeptical Sam doesn't have time to test software extensively but is always keen on asking aggressive questions to the author to find out if a program could fit his needs.

S. Sam: PSFEx represents PSFs as an array of tabulated values! Can it really deal with undersampled images? Isn't it too noisy?

Author: PSFEx was designed from the ground up to deal with undersampled images and arbitrary PSFs. Although the PSF "model" in PSFEx is actually a small image, it is sampled at a different step than the original pixels: more finely for undersampled observations, and more coarsely for oversampled observations, to avoid any loss and redundancy of information. Despite built-in regularisation, PSF models reconstructed on the pixel basis can indeed be noisy if the number of selected stars is small. This can be circumvented to some extent by using *ad hoc* basis to solve for the PSF model coefficients.

S. Sam: I heard that PSFEx has been developped almost 12 years ago, and has been used for production at TERAPIX for many years. Why have you waited until 2010 for releasing it to the general community?

Author: PSFEx was originally developped for doing PSF-fitting crowded-field photometry with SExtractor. However I was not very happy with the way it worked, as SExtractor's detection and deblending engine is not meant to deal with crowded star fields. The current

[1] Source Extractor, `http://www.astromatic.net/sextractor`

release of PSFEx is made in the framework of the EFIGI[2] and DES[3] projects, as a support tool for galaxy model-fitting.

S. Sam: I would like to use PSFEx to generate PSF models for weak-lensing analyses. Is it the right tool for that?

Author: Possibly. Simulations of 1h exposures with a 4m optical telescope and sub-arcsecond seeing show that ellipticities of galaxies with a Signal-to-Noise Ratio SNR>20 can be recovered with a level of systematics below 10^{-3} using PSFEx models, even in the presence of significant amounts of coma and astigmatism. This is for constant PSFs. Tests with variable PSFs are ongoing.

3 License

PSFEx is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version. PSFEx is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details. You should have received a copy of the GNU General Public License along with PSFEx. If not, see `http://www.gnu.org/licenses/`.

4 Installing the software

4.1 Obtaining PSFEx

The easiest way to obtain PSFEx is to download it from the official website[4]. At this address, the latest versions of the program (source code, configuration files, and documentation) are available as standard `.tar.gz` Unix source archives as well as RPM binary packages for various architectures.[5]

4.2 Software and hardware requirements

PSFEx has been developed on Unix machines (GNU/Linux), and should compile on any POSIX-compliant system (this should include Mac OS X and Cygwin under Windows, at the price of some difficulties with the configuration), provided that the following libraries/packages have been installed:

- ATLAS V3.6 and above[6] (`http://math-atlas.sourceforge.net/`)
- FFTw V3.0 and above[7] (`http://www.fftw.org/`)
- PLPlot V5.9 and above (`http://www.plplot.org/`)

[2] Extraction de Formes Idealisées de Galaxies en Imagerie, `http://www.efigi.org`

[3] Dark Energy Survey, `http://www.darkenergysurvey.org`

[4] `http://astromatic.net/software/psfex`

[5] Mac OS X dmg files should be available soon.

[6] Use the `--with-atlas` and/or `--with-atlas-incdir` options to specify the ATLAS library and include paths if the software is installed at unusual locations.

[7] Make sure that FFTW has been compiled with the `configure` options `--enable-threads --enable-float`).

PLPlot is only required for producing diagnostic plots. Note that ATLAS and FFTw are not necessary for the binary versions of PSFEx which come with these libraries statically linked.

The software is run in (ANSI) text-mode from a shell. A window system is necessary only when PLPlot is used in interactive mode.

The amount of memory required by PSFEx depends mostly on the number of point sources present in the input catalogues times the number of pixels in the small image that represents each of them. A typical figure is about 15 kbytes per point source; hence even on a modest computer with 256MB of memory, more than 10,000 point sources can easily be accommodated at once.

4.3 Installation from the source archive

To install from the source, you must first uncompress and "untar" the archive:

```
tar zxvf psfex-<version>.tar.gz
```

A new directory called `psfex-<version>` should now appear at the current location on your disk. You should then enter the directory and follow the instructions in the file called "`INSTALL`".

4.4 Installation from an RPM archive

PSFEx is also available as a binary RPM package for both Linux INTEL `x86` (32-bit) and `x86-64` (64-bit) architectures. To check which matches your system, use the shell command

```
uname -a
```

The RPM version of PSFEx requires the PLPlot package. Make sure it is installed before proceeding. To install PSFEx, type as a root user the following command in your shell (preceded with `su` if you don't have root access bu the system administrator trusts you well enough to make you part of the `wheel` group):

```
rpm -U psfex-<version>-1.<arch>.rpm
```

It is often necessary to force installation with

```
rpm -U --force --nodeps psfex-<version>-1.<arch>.rpm
```

You may now check that the software is properly installed by simply typing in your shell

```
psfex
```

(note that some shells require the `rehash` command to be run before making a freshly installed executable accessible in the execution path).

5 Using PSFEx

PSFEx is run from the shell with the following syntax:

```
% psfex Catalog1 [Catalog2 ...] -c configuration-file
        [ [-Parameter1 Value1] - Parameter2Value2 ...]
```

The parts enclosed within brackets are optional. The file names of input catalogues can be directly provided in the command-line, or in lists that are ASCII files with each catalogue name preceded by '@' (one per line). One should use lists instead of the catalogue file names if the number of input catalogues is too large to be handled directly by the shell. Any "-*Parameter Value*" statement in the command-line overrides the corresponding definition in the configuration-file or any default value (see below).

5.1 Input files

5.1.1 Catalogues

PSFEx does not work directly on images. Instead, it operates on SExtractor catalogues that have a small image ("vignette") recorded for each detection. This makes things much easier for PSFEx as it does not have to handle the detection and deblending processes. **The catalogue files read by PSFEx must be in SExtractor "`FITS_LDAC`" binary format** . This allows PSFEx to have access to the original image header content. The catalogues *must* contain all the following parameters in order to be processable by PSFEx:

- small image ("vignette") centered on the object `VIGNET(w,h)`, where `w` and `h` are respectively the width and the height of the image in pixels,
- centroid coordinates, e.g. `X_IMAGE` and `Y_IMAGE`,
- half-light radius `FLUX_RADIUS`,
- flux measured through a fixed aperture, e.g. `FLUX_APER(1)`,
- flux uncertainty, e.g. `FLUXERR_APER(1)`,
- object elongation `ELONGATION`,
- extraction flags `FLAGS`.

The `VIGNET` dimensions `w` and `h` set the maximum size of the zone in which PSF models are fit to each candidate source; it is recommended to stick to `w` = `h`. The extraction parameters for making the input catalogues require little refinement; it is generally more convenient to set fairly high detection thresholds to keep catalogue sizes reasonable. However **it is important to make sure that the SExtractor configuration keywords `GAIN`, `SATUR_LEVEL` and `PHOT_APERTURES` (which will act as the reference aperture for profile-fitting photometry) are all set to the correct values**[8].

5.2 Output files

5.2.1 "`.psf`" PSF model files

The main purpose of PSFEx is to create a PSF model for each of the images from which the input catalogues were extracted. The PSF models are stored as FITS binary tables, under file names that are given the `.psf` extension by default (this may be changed with the `PSF_SUFFIX` configuration parameter). The `.psf` files can be read back in SExtractor to perform accurate model-fitting of the sources detected.

[8] Important: `GAIN` and `SATUR_LEVEL` are by default overridden by the values of the FITS image header keywords specified with the `GAIN_KEY` and `SATUR_KEY` configuration parameters, respectively.

5.2.2 “.homo” PSF homogenisation files

This is presently an experimental feature. In addition to computing PSF models, PSFEx has the possibility to derive “PSF homogenisation kernels” for all input catalogues. A PSF homogenisation kernel is a (variable) convolution kernel which, when applied to an image, gives the point sources it contains a constant, arbitrary shape. For practical purposes the target shape will preferably be a perfectly round analytical function, such as a Moffat (1969) profile:

$$I(r) = I_0 \left[1 + \left(\frac{r}{a}\right)^2\right]^{-\beta} \tag{1}$$

Homogenising the PSF of a set of images can allow for more consistent image combinations and measurements, once the consequences on noise have been properly taken into account.

PSFEx stores PSF homogenisation kernels as FITS data cubes. File names are given the `.homo` extension by default; this may be changed using the `HOMOKERNEL_SUFFIX` configuration parameter. `.homo` files can be read by the PSFNormalize software (developed by Tony Darnell from the Dark Energy Survey data-management team) to perform fast convolution of the original images. The SWarp software may also later include this possibility.

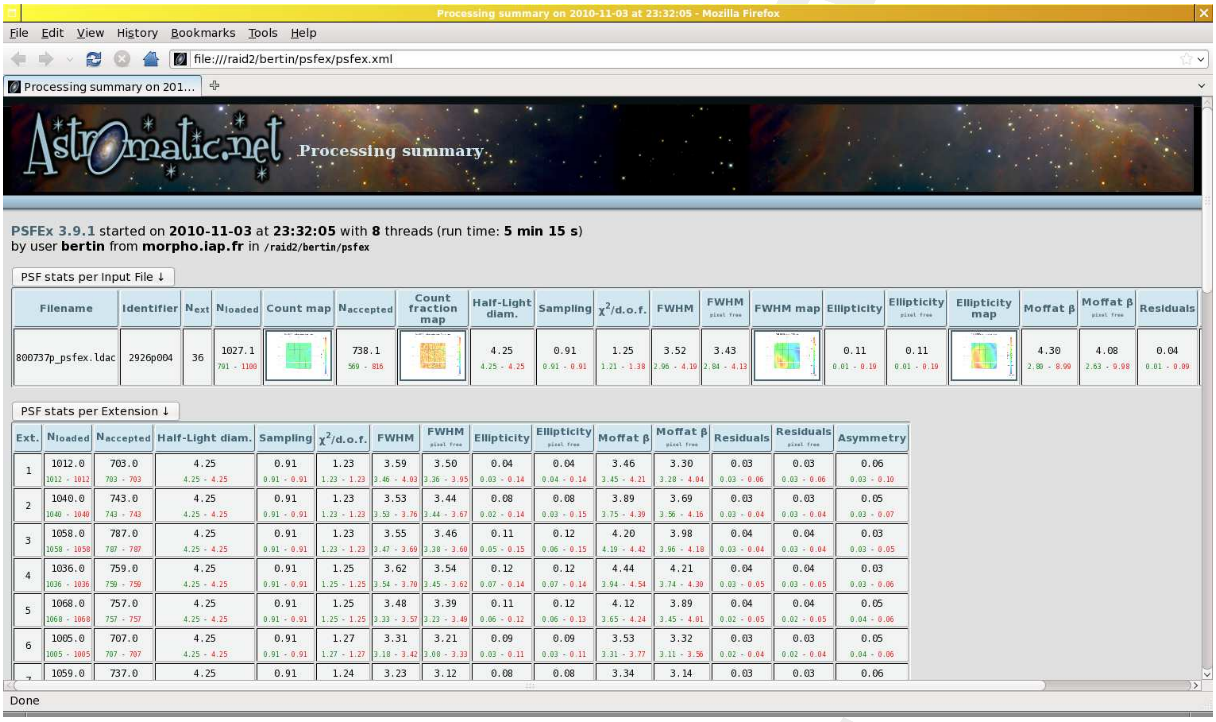
5.2.3 Diagnostic files

Three types of files can be generated by PSFEx, providing diagnostics about the derived PSF and the modelling process:

- “Check-images” are basic FITS files containing images of the PSF model, fit residuals, etc.. Configuration parameters `CHECKIMAGE_TYPE` and `CHECKIMAGE_NAME` allow the user to provide a list of check-image types and file names, respectively, to be produced by PSFEx. A complete list of available check-image types is given in §5.3.3. Many check-images are actually aggregates of several small images; they may be stored as grids (the default) or as datacubes if the `CHECKIMAGE_DATACUBE` parameter is set to `Y`.
- “Check-plots” are graphic charts generated by PSFEx, showing maps or trends of PSF measurements. The `CHECKPLOT_TYPE` and `CHECKPLOT_NAME` configuration parameters allow the user to provide a list of check-plot types and file names, respectively. A variety of raster and vector file formats, from JPEG to Postscript, can be set with `CHECKPLOT_DEV` (the default format is `PNG`). See the `CHECKPLOT` section of §5.3.3 for details.
- An XML file providing a processing summary and various statistics in VOTable format is written if the `WRITE_XML` switch is set to `Y` (the default). The `XML_NAME` parameter can be used to change the default file name `psfex.xml`. The XML file can be displayed with any recent web browser; the XSLT stylesheet installed together with PSFEx will automatically translate it into a dynamic, user-friendly web-page (Fig. 1). For more advanced usages (e.g. access from a remote web server), alternative XSLT translation URLs may be specified using the `XSL_URL` configuration parameter.

5.3 The Configuration file

Each time it is run, PSFEx looks for a configuration file. If no configuration file is specified in the command-line, it is assumed to be called “`default.psfex`” and to reside in the current directory. If no configuration file is found, PSFEx will use its own internal default configuration.


Processing summary on 2010-11-03 at 23:32:05 - Mozilla Firefox
File Edit View History Bookmarks Tools Help
file:///raid2/bertin/psfex/psfex.xml
Processing summary on 201...
Astromatic.net
Processing summary
Done

PSFEx 3.9.1 started on **2010-11-03** at **23:32:05** with **8** threads (run time: **5 min 15 s**)
by user **bertin** from **morpho.iap.fr** in `/raid2/bertin/psfex`

PSF stats per Input File ↓

Filename	Identifier	N_{ext}	N_{loaded}	Count map	$N_{accepted}$	Count fraction map	Half-Light diam.	Sampling	χ^2/d.o.f.	FWHM	FWHM pixel free	FWHM map	Ellipticity	Ellipticity pixel free	Ellipticity map	Moffat β	Moffat β pixel free	Residuals
800737p_psfex.ldac	2926p004	36	1027.1 791 - 1100		738.1 569 - 816		4.25 4.25 - 4.25	0.91 0.91 - 0.91	1.25 1.21 - 1.38	3.52 2.96 - 4.19	3.43 2.84 - 4.13		0.11 0.01 - 0.19	0.11 0.01 - 0.19		4.30 2.80 - 8.99	4.08 2.63 - 9.98	0.04 0.01 - 0.09

PSF stats per Extension ↓

Ext.	N_{loaded}	$N_{accepted}$	Half-Light diam.	Sampling	χ^2/d.o.f.	FWHM	FWHM pixel free	Ellipticity	Ellipticity pixel free	Moffat β	Moffat β pixel free	Residuals	Residuals pixel free	Asymmetry
1	1012.0 1012 - 1012	703.0 703 - 703	4.25 4.25 - 4.25	0.91 0.91 - 0.91	1.23 1.23 - 1.23	3.59 3.46 - 4.03	3.50 3.36 - 3.95	0.04 0.03 - 0.14	0.04 0.04 - 0.14	3.46 3.45 - 4.21	3.30 3.28 - 4.04	0.03 0.03 - 0.06	0.03 0.03 - 0.06	0.06 0.03 - 0.10
2	1040.0 1040 - 1040	743.0 743 - 743	4.25 4.25 - 4.25	0.91 0.91 - 0.91	1.23 1.23 - 1.23	3.53 3.53 - 3.76	3.44 3.44 - 3.67	0.08 0.02 - 0.14	0.08 0.03 - 0.15	3.89 3.75 - 4.39	3.69 3.56 - 4.16	0.03 0.03 - 0.04	0.03 0.03 - 0.04	0.05 0.03 - 0.07
3	1058.0 1058 - 1058	787.0 787 - 787	4.25 4.25 - 4.25	0.91 0.91 - 0.91	1.23 1.23 - 1.23	3.55 3.47 - 3.69	3.46 3.38 - 3.60	0.11 0.05 - 0.15	0.12 0.06 - 0.15	4.20 4.19 - 4.42	3.98 3.96 - 4.18	0.04 0.03 - 0.04	0.04 0.03 - 0.04	0.03 0.03 - 0.05
4	1036.0 1036 - 1036	759.0 759 - 759	4.25 4.25 - 4.25	0.91 0.91 - 0.91	1.25 1.25 - 1.25	3.62 3.54 - 3.70	3.54 3.45 - 3.62	0.12 0.07 - 0.14	0.12 0.07 - 0.14	4.44 3.94 - 4.54	4.21 3.74 - 4.30	0.04 0.03 - 0.05	0.04 0.03 - 0.05	0.03 0.03 - 0.06
5	1068.0 1068 - 1068	757.0 757 - 757	4.25 4.25 - 4.25	0.91 0.91 - 0.91	1.25 1.25 - 1.25	3.48 3.33 - 3.57	3.39 3.23 - 3.49	0.11 0.06 - 0.12	0.12 0.06 - 0.13	4.12 3.65 - 4.24	3.89 3.45 - 4.01	0.04 0.02 - 0.05	0.04 0.02 - 0.05	0.05 0.04 - 0.06
6	1005.0 1005 - 1005	707.0 707 - 707	4.25 4.25 - 4.25	0.91 0.91 - 0.91	1.27 1.27 - 1.27	3.31 3.18 - 3.42	3.21 3.08 - 3.33	0.09 0.03 - 0.11	0.09 0.03 - 0.11	3.53 3.31 - 3.77	3.32 3.11 - 3.56	0.03 0.02 - 0.04	0.03 0.02 - 0.04	0.05 0.04 - 0.06
7	1059.0	737.0	4.25	0.91	1.24	3.23	3.12	0.08	0.08	3.34	3.14	0.03	0.03	0.06

Figure 1: Rendition of a `psfex.xml` XML-VOTable file generated by PSFEx with the Firefox web-browser.

5.3.1 Creating a configuration file

PSFEx can generate an ASCII dump of its internal default configuration, using the "`-d`" option. By redirecting the standard output of PSFEx to a file, one creates a configuration file that can easily be modified afterwards:

```
% psfex -d > default.psfex
```

A more extensive dump with less commonly used parameters can be generated by using the "`-dd`" option.

5.3.2 Format of the configuration file

The format is ASCII. There must be only one parameter set per line, following the form:

Config-parameter *Value(s)*

Extra spaces or linefeeds are ignored. Comments must begin with a "#" and end with a linefeed. Values can be of different types: strings (can be enclosed between double quotes), floats, integers, keywords or Boolean (`Y/y` or `N/n`). Some parameters accept zero or several values, which must then be separated by commas. Integers can be given as decimals, in octal form (preceded by digit `0`), or in hexadecimal (preceded by `0x`). The hexadecimal format is particularly convenient for writing multiplexed bit values such as binary masks. Environment variables, written as `$HOME` or `${HOME}` are expanded.

5.3.3 Configuration parameter list

Here is a list of all the parameters known to PSFEx. Please refer to next section for a detailed description of their meaning. Some "advanced" parameters (indicated with an asterisk) are also listed. They must be used with caution, and may be rescoped or removed without notice in future versions.

`BADPIXEL_FILTER`* `N` *Boolean*

If true (`Y`), input objects with vignettes containing more than `BADPIXEL_NMAX` pixels flagged by SExtractor as bad or from deblended neighbours will be rejected.

`BADPIXEL_NMAX`* `0` *integer*

Maximum number of bad pixels tolerated in the vignette before an object is rejected (`BADPIXEL_FILTER` must be set to `Y`).

`BASIS_NAME`* `basis.fits` *string*

File name for the user-supplied FITS datacube of basis vector images (`BASIS_TYPE` must have been set to `FILE`).

`BASIS_NUMBER` `20` *integer*

Size of basis vector set: square-root of the number of pixels for `BASIS_TYPE PIXEL`, $n_{\max}$ for `BASIS_TYPE GAUSS-LAGUERRE`, or number of vectors for `BASIS_TYPE FILE`

`BASIS_SCALE`* `1.0` *float*

Scale size of `BASIS_TYPE GAUSS-LAGUERRE` vector images.

`BASIS_TYPE` `PIXEL_AUTO` *keyword*

Basis vector set:

`NONE`	No basis; the PSF is derived solely from the robust combination of resampled input vignettes.
`PIXEL`	Pixel basis (super-resolution).
`PIXEL_AUTO`	Equivalent to `NONE` for properly sampled images; switches automatically to `PIXEL` (super-resolution) for critically sampled and undersampled data.
`GAUSS_LAGUERRE`	Gauss-Laguerre basis (also known as *polar shapelets* in the weak-lensing community).
`FILE`	User-supplied vector basis, in the form of a FITS datacube (see `BASIS_NAME`).

`CENTER_KEYS` `X_IMAGE,Y_IMAGE` *strings*

Catalogue "Keys" (SExtractor measurement parameters) that define the initial guess for the source coordinates. Note that all input "vignettes" are automatically re-centred by PSFEx using an iterative Gaussian-weighted algorithm, hence the centring parameter is not critical.

`CHECKIMAGE_CUBE`* `N` *Boolean*

If true (`Y`), check-images will be saved as data-cubes.

`CHECKIMAGE_NAME` `chi.fits`, `proto.fits`, `samp.fits`, `resi.fits`, `snap.fits` *strings*

File name of the check-image (diagnostic FITS image) of each type (`.fits` extension is not required, as it is assumed by default).

`CHECKIMAGE_TYPE` `CHI`, `PROTOTYPES`, `SAMPLES`, `RESIDUALS`, `SNAPSHOTS` *keywords*

Types of check-images (diagnostic FITS images) to generate during PSFEx processing:

`NONE`	No check-image.
`CHI`	(square-root of) χ^2 maps for all input vignettes.
`PROTOTYPES`	Versions of input vignettes, recentred, rescaled and resampled to PSF resolution.
`SAMPLES`	Input vignettes in their original position, resolution and flux scaling.
`RESIDUALS`	Input vignettes with best-fitting local PSF models subtracted.
`SNAPSHOTS`	Grid of PSF model snapshots reconstructed at each position/context.
`MOFFAT`	Grid of Moffat models (eq. [15]) fitted to PSF model snapshots at each position/context.
`-MOFFAT`	Grid of PSF model snapshots reconstructed at each position/context with best-fitting Moffat models (eq. [15]) subtracted.
`-SYMMETRICAL`	Grid of PSF model snapshots reconstructed at each position/context with symmetrised image subtracted.
`BASIS`	Basis vector images used by PSFEx to model the PSF.

`CHECKPLOT_ANTIALIAS`* `Y` *Boolean*

If true (`Y`), PBM, PNG and JPEG check-plots are generated with anti-aliasing. ImageMagick's `convert` tool must be installed. See `http://www.imagemagick.org`.

`CHECKPLOT_DEV` `PNG` *keywords*

PLPlot devices to be used for check-plots (all devices may not be available, see PLPlot documentation for details):

`NULL`	No output
`XWIN`	X-Window
`TK`	Tk window (if available)
`XTERM`	XTerm window
`AQUATERM`	AquaTerm window (Mac OS X)
`PLMETA`	PLPlot `.plm` meta-file
`XFIG`	XFig `.fig` vector file
`LJIIP`	HP LaserJet IIP `.lj` bitmap file
`LJ_HPGL`	HP LaserJet `.hpg` HPGL vector file
`IMP`	Impress `.imp` file
`PBM`	Portable BitMap `.pbm` image
`PNG`	Portable Network Graphics `.png` image
`JPEG`	JPEG `.jpg` image
`PDF`	Portable Document Format `.pdf` file
`PS`	Black-and-white `.ps` Postscript file
`PSC`	Colour `.ps` Postscript file
`PSTEX`	PSTeX (a variant of Postscript) `.ps` file

`CHECKPLOT_NAME` `fwhm`, `ellipticity`, `counts`, `countfrac`,`chi2`, `resi` *strings*

File names for each series of check-plots. PSFEx will automatically insert the associated catalogue names, and append/replace file name extensions with the appropriate ones, depending on the chosen CHECKPLOT_DEV(s) (.png for PNG files, .jpg for JPEG, etc.).

CHECKPLOT_RES* 0 *integers* ($n \leq 2$)

Check-plot x,y resolution for bitmap devices (0 is equivalent to $800, 600$).

CHECKPLOT_TYPE FWHM, ELLIPTICITY, COUNTS,COUNT_FRACTION, CHI2, RESIDUALS *keywords*

Diagnostic check-plots to be generated during PSFEx processing (PSFEx must have been configured without the --without-plplot option):

NONE	No plot.
FWHM	Map of the model PSF Full-Width at Half-Maximum over the field of view (one for each input catalogue).
ELLIPTICITY	Map of the model PSF ellipticity over the field of view (one for each input catalogue).
COUNTS	Map of the spatial density of point sources (initially) selected over the field of view (one for each catalogue).
COUNT_FRACTION	Map of the fraction of point sources accepted over the field of view (one for each catalogue).
CHI2	Map of the average χ^2/d.o.f. over the field of view (one for each catalogue).
MOFFAT_RESIDUALS	Map of Moffat (eq. [15]) residual indices over the field of view (one for each catalogue).
ASYMMETRY	Map of asymmetry indices over the field of view (one for each catalogue).

HOMOBASIS_NUMBER* 10 *integer*

Size of the homogenisation kernel basis vector set: $n_{\max}$ for HOMOBASIS_TYPE GAUSS-LAGUERRE

HOMOBASIS_SCALE* 1.0 *float*

Scale size of HOMOBASIS_TYPE GAUSS-LAGUERRE homogenisation kernel vector images.

HOMOBASIS_TYPE* NONE *keyword*

Basis vector set for the homogenisation kernel:

NONE	No basis; no homogenisation kernel is computed.
GAUSS_LAGUERRE	Gauss-Laguerre basis (also known as *polar shapelets* in the weak-lensing community).

HOMOKERNEL_SUFFIX* .homo.fits *string*

Filename suffix of the homogenisation kernels computed by PSFEx.

HOMOPSF_PARAMS* 2.0, 3.0 *floats* ($n \leq 2$)

Moffat Full-Width at Half-Maximum and β parameters (eq. [15]) of the idealised target PSF chosen for homogenisation.

MEF_TYPE* INDEPENDENT *keyword*

How PSFEx should deal with multi-extension catalogues (extracted from mosaic camera images):

`INDEPENDENT`	Derive the PSF model for each extension independently.
`COMMON`	Derive a common PSF model for all extensions.

`NEWBASIS_NUMBER*` `8` *integer*

Size of the image vector set (number of basis vectors) derived by PSFEx from the input vignettes.

`NEWBASIS_TYPE*` `NONE` *keyword*

Type of image vector bases derived from input vignettes by PSFEx:

`NONE`	No basis is computed.
`PCA_MULTI`	Karhunen-Loève basis from Principal Component Analysis on all FITS extensions.
`PCA_SINGLE`	Karhunen-Loève bases from Principal Component Analysis on individual FITS extensions.

`NTHREADS` `0` *integer*

Number of threads (processes) to be used for parallel computation. PSFEx must have been configured with the `--disable-threads` option at compile time for this parameter to take effect. Note that multi-threading is disabled in the current version of PSFEx

`PHOTFLUX_KEY` `FLUX_APER(1)` *string*

Catalogue "Key" (SExtractor measurement parameter) that defines the flux of sources, and therefore the normalisation of the PSF amplitude. It is recommended to use a fixed aperture magnitude; the aperture diameter set in SExtractor should be large enough so that the fraction of flux enclosed stays constant from point source to point source, and small enough to preserve the signal-to-noise ratio.

`PHOTFLUXERR_KEY` `FLUXERR_APER(1)` *string*

Catalogue "Key" (SExtractor measurement parameter) that defines the flux measurement uncertainty on each source. It is used for computing the source signal-to-noise ratio.

`PSF_ACCURACY` `0.01` *float*

Expected accuracy of vignette pixel values (standard deviation of the flux fraction).

`PSF_PIXELSIZE` `1.0` *float*

Effective pixel size (width of the top-hat intra-pixel response function) in pixel step units.

`PSF_RECENTER` `Y` *Boolean*

If true (`Y`), input vignettes are recentred at each iteration of the PSF modelling process.

`PSF_SAMPLING` `0.0` *float*

Sampling step of the PSF models, in pixels. Use `0` for automatic sampling.

`PSF_SIZE` `25, 25` *integers ($n \leq 2$)*

Dimensions of the tabulated PSF models, in PSF "pixels".

`PSF_SUFFIX*` `.psf` *string*

Filename suffix for PSF models computed by PSFEx.

`PSFVAR_DEGREES` `2` *integers ($n = n_{\rm groups}$)*

Degree of polynomial of each context group. `0` indicates a constant PSF.

`PSFVAR_GROUPS` `1, 1` *integers ($n = n_{\tt PSFVAR\_KEYS}$)*

Polynomial group which each context key belongs to.

`PSFVAR_KEYS` `X_IMAGE, Y_IMAGE` *strings* ($n \leq 2$)

List of "keys" (SExtractor measurement parameters) on which the PSF is supposed to depend (e.g. `X_IMAGE,Y_IMAGE` for a spatial mapping of the PSF). Keywords preceded with a colon are interpreted as FITS image keywords instead of SExtractor parameters.

`PSFVAR_NSNAP`* `9` *integer*

Number of PSF snapshots computed on each axis. This also defines the resolution of the grid on which diagnostics and check-plot maps are computed.

`SAMPLE_AUTOSELECT` `Y` *Boolean*

If true (`Y`), input vignettes are automatically selected based on the source FWHMs, inside the range specified by `SAMPLE_FWHMRANGE`, with fractional FWHM variability `SAMPLE_VARIABILITY`.

`SAMPLE_FLAGMASK`* `0x00fe` *integer*

Bit mask applied to SExtractor flags for rejecting input vignettes.

`SAMPLE_FWHMRANGE`* `2.0, 10.0` *floats ($n = 2$)*

Range (in pixels) of source FWHMs (Full-Width at Half-Maximum) allowed for input vignettes.

`SAMPLE_MAXELLIP` `0.3` *float*

Maximum source ellipticity allowed for input vignettes (i.e. `B_IMAGE/A_IMAGE` > 0.7 by default).

`SAMPLE_MINSN` `20.0` *float*

Minimum source Signal-to-Noise ratio allowed for input vignettes.

`SAMPLE_VARIABILITY` `0.2` *float*

Maximum fractional FWHM variability (1.0 = 100%) allowed for input vignettes.

`SAMPLEVAR_TYPE`* `SEEING` *keyword*

Catalogue-to-catalogue variability criteria for vignette selection:

`NONE`	No differences between catalogues.
`SEEING`	Seeing (hence FWHM) is expected to vary.

`STABILITY_TYPE`* `EXPOSURE` *keyword*

The SMP version of PSFEx:

`EXPOSURE`	???
`SEQUENCE`	???

`VERBOSE_TYPE` `NORMAL` *keyword*

Degree of verbosity of the software on screen:

`QUIET`	No Output besides warnings and error messages
`NORMAL`	"Normal" display with messages updated in real time using ASCII escapes-sequences

`LOG`	Like `NORMAL`, but without real-time messages and ASCII escape-sequences
`FULL`	Everything

`WRITE_XML` `Y` *Boolean*

If true (`Y`), an XML summary file will be written after completing the processing.

`XML_NAME` `psfex.xml` *string*

File name for the XML output of PSFEx.

`XSL_URL*` `.` *string*

URL of an XSL style-sheet for the XML output of PSFEx. This URL will appear in the `href` attribute of the `style-sheet` tag.

6 How PSFEx works

6.1 Overview of the software

The global layout of PSFEx is presented in Fig. 2. There are many ways to operate the software. Let us now describe the important steps in the most common usage modes.

1. PSFEx starts by examining the catalogues given in the command line. In the default operating mode, for mosaic cameras, Multi-Extension FITS (MEF) files are processed extension by extension. PSFEx pre-selects detections which are likely to be point sources, based on source characteristics such as half-light radius and ellipticity, while rejecting contaminated or saturated objects.
2. For each pre-selected detection, the "vignette" (produced by SExtractor) and a "context vector" are loaded in memory. The context vector represents the set of parameters (like position) on which the PSF model will depend explicitly.
3. The PSF modelling process is iterated 4 times. Each iteration consists of computing the PSF model, comparing the vignettes to the model reconstructed in their "local contexts", and excluding detections that show too much departure between the data and the model.
4. Depending on the configuration, two types of Principal Component Analyses (PCAs) may be included at this stage, either to build an optimised image vector basis to represent the PSF, or to track hidden dependencies of the model. In both cases, they result in a second round of PSF modelling.
5. The PSF models are saved to disk. If requested, PSF homogenisation kernels may also be computed and written to disk at this stage. Finally, diagnostic files are generated.

6.2 Point source selection

PSFEx requires the presence of unresolved sources (stars or quasars) in the input catalogue(s) to extract a valid PSF model. In some astronomical observations, the fraction of suitable point sources that may be used as good approximations to the local PSF may be rather low. This is especially true for deep imaging in the vicinity of galaxy clusters at high galactic latitudes, where unsaturated stars may comprise only a small percentage of all detectable sources.

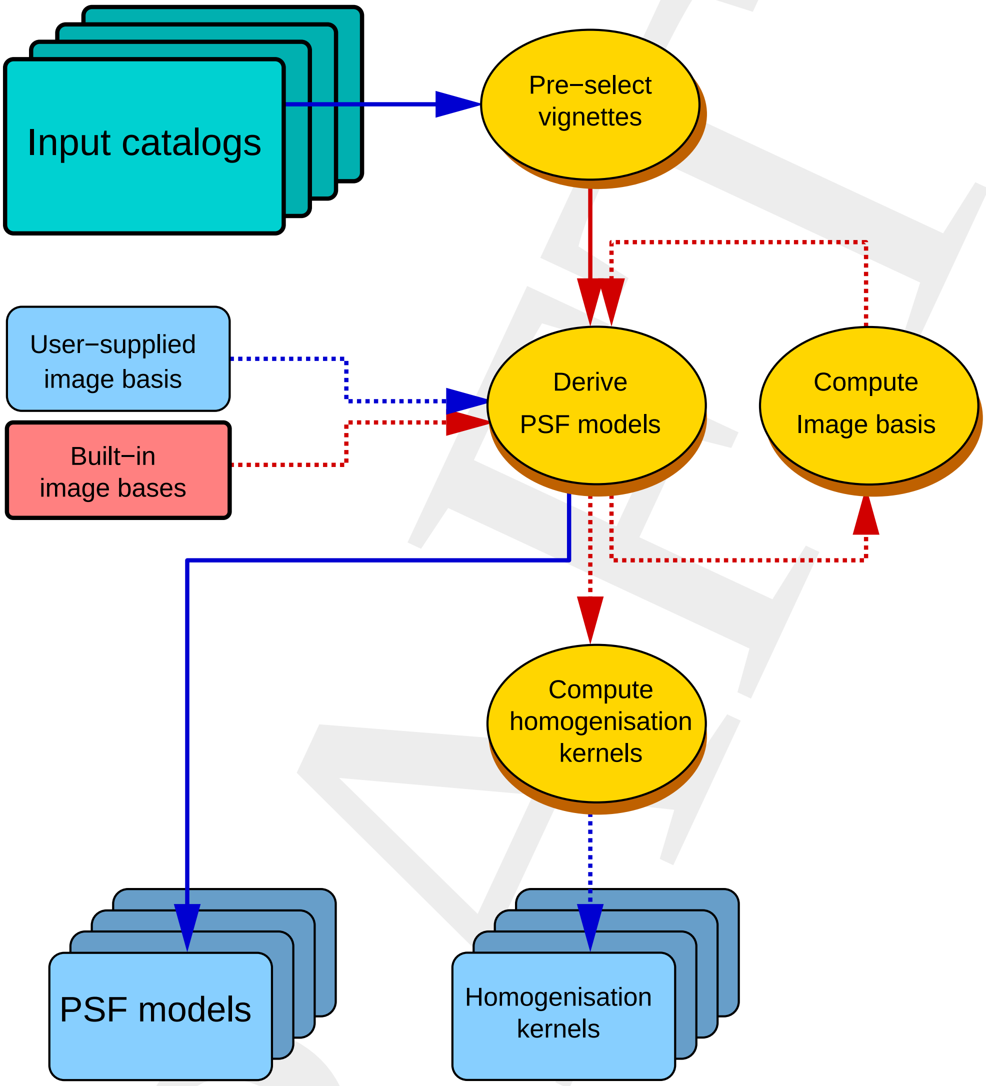


Figure 2: Global Layout of PSFEx.

6.2.1 Selection criteria

To minimise as much as possible the assumptions on the shape of the PSF, PSFEx adopts the following selection criteria:

- the shape of suitable unresolved (unsaturated) sources does not depend on the flux.
- amongst image profiles of all real sources, those from unresolved sources have the smallest Full-Width at Half Maximum (FWHM).

These considerations as well as much experimentation led to adopting a first-order selection similar to the rectangular cut in the half-light-radius ($r_{\rm h}$) vs. magnitude plane, popular amongst members of the weak lensing community (Kaiser et al. 1995). SExtractor's `FLUX_RADIUS` parameter with input parameter `PHOT_FLUXFRAC`=`0.5` provides a good estimate for $r_{\rm h}$. In PSFEx, the "vertical" locus produced by point sources (whose shape does not depend on magnitude) is automatically framed between a minimum signal-to-noise threshold and the saturation limit on the magnitude axis, and within some margin around the local mode on the $r_{\rm h}$ axis (Fig. 3). The

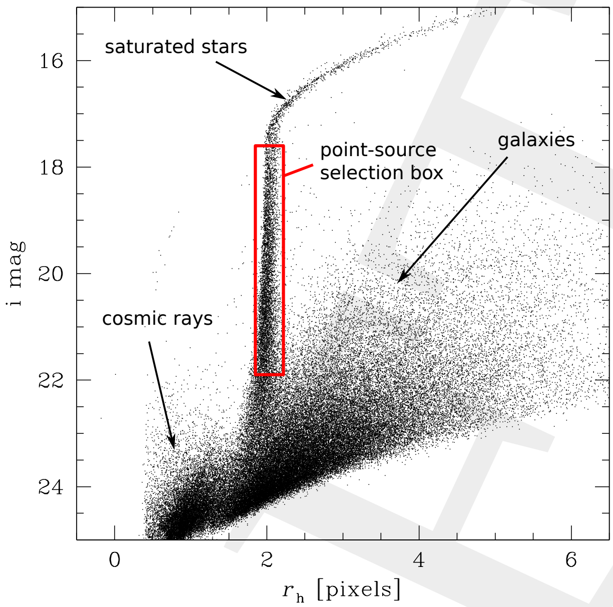


Figure 3: Half-light-radius ($r_{\rm h}$, estimated by SExtractor's FLUX_RADIUS) vs magnitude (MAG_AUTO) for a 520 s CFHTLS exposure at high galactic latitude taken with the Megaprime instrument in the i band. The rectangle enclosing part of the stellar locus represents the approximate boundaries set automatically by PSFEx to select point sources.

relative width of the selection box is set by the SAMPLE_VARIABILITY configuration parameter (0.2 by default), within boundaries defined by half the SAMPLE_FWHMRANGE parameter (between 2 and 10 pixels by default). Additionally, to provide a better rejection of image artifacts and multiple objects, PSFEx excludes detections

- with a Signal-to-Noise Ratio (SNR) below the value set with the SAMPLE_MINSN configuration parameter (20 by default). The SNR is defined here as the ratio between the source flux and the source flux uncertainty.
- with SExtractor extraction FLAGS that match the mask set by the SAMPLE_FLAGMASK configuration keyword. The default mask (00fe in hexadecimal) excludes all flagged objects, except those with FLAGS=1 (indicating a crowded environment).
- with an ellipticity exceeding the value set with the SAMPLE_MAXELLIP configuration parameter (0.3 by default). The ellipticity is defined here as $(1-a)/(1+a)$, where a is the source aspect ratio; an ellipticity of 0.3 corresponds roughly to an aspect ratio of 1:2.
- that include pixels that were given a weight of 0 (for weighted source extractions).

6.2.2 Iterative filtering

Despite the filtering process, a small fraction of the remaining point source candidates (typically 5-10% on ground-based optical images at high galactic latitude) is still unsuitable to serve as a realisation of the local PSF, because of contamination by neighbour objects. Iterative procedures to subtract the contribution from neighbour stars have been successfully applied in crowded fields (Stetson 1987, Magain et al. 2007). However these techniques do not solve the problem of pollution by non-stellar objects like image artifacts, a common curse of wide field imaging, and contaminated point sources still have to be filtered out.

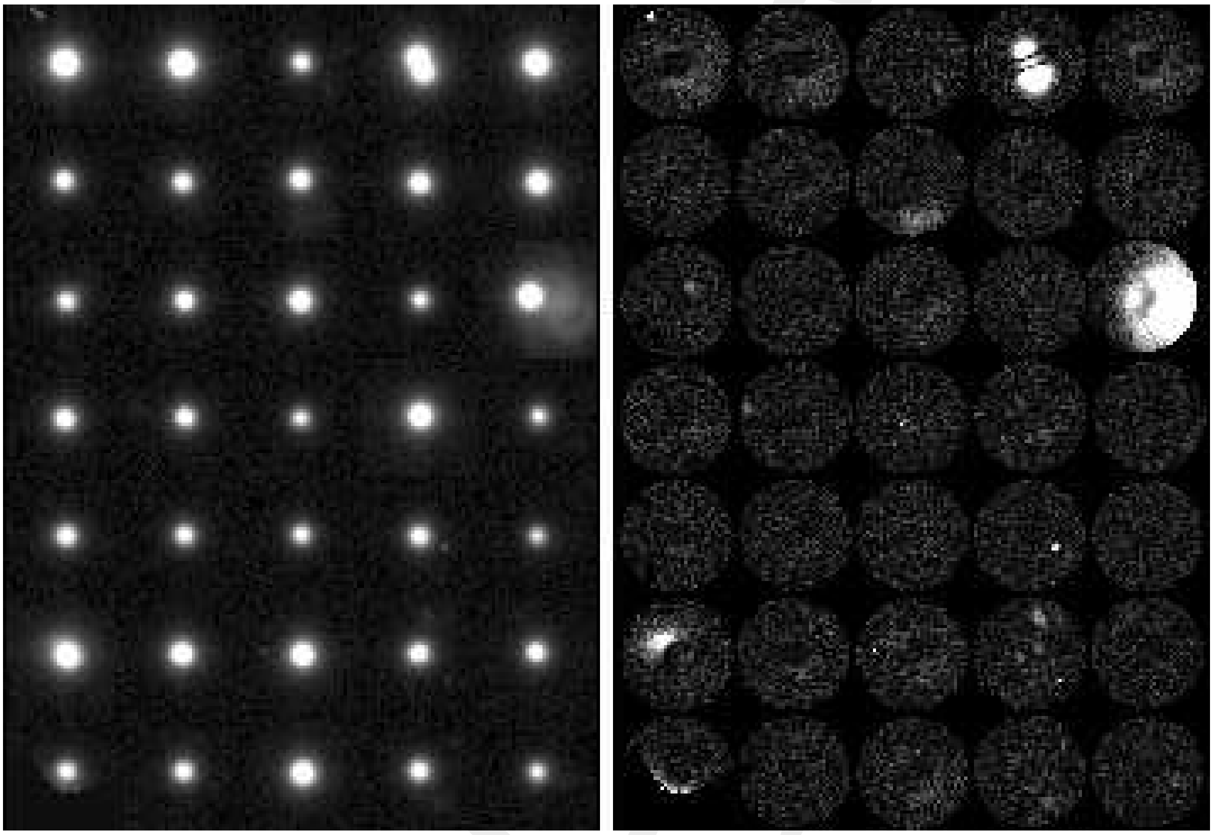


Figure 4: *Left*: some source images selected for deriving a PSF model of a MEGACAM image (the basic rejection tests based on SExtractor flags and measurements were voluntarily bypassed to increase the fraction of contaminants in this illustration). *Right*: map of residuals computed as explained in the text; bright pixels betray interlopers like cosmic ray hits and close neighbour sources.

The iterative rejection process in PSFEx works by deriving a 1st-order estimate of the PSF model, and computing a map of the residuals of the fit of this model to each point source (Fig. 4): each pixel of the map is the square of the difference square of the model with the data, divided by the σ_i^2 estimate from equation (4). The PSF model may be "rough" at the first iteration, hence to avoid penalising poorly fitted bright source pixels, the factor α is initially set to a fairly large value, 0.1-0.3. Assuming that the fitting errors are normally distributed, and given the large number of degrees of freedom (the tabulated values of the model), the distribution of $\sqrt{\chi^2}$ derived from the residual maps of point sources is expected to be Gaussian to a good approximation. Contaminated profiles are identified using $\kappa - \sigma$ clipping to the distribution of $\sqrt{\chi^2}$. Our experiments indicate that the value $\kappa = 4$ provides a consistent compromise between being too restrictive and being too permissive. PSFEx repeats the PSF modelling / source rejection process 3 more times, with decreasing α, before delivering the "clean" PSF model.

6.3 Modelling the PSF

In PSFEx, the PSF is modelled as a linear combination of basis vectors. Since the PSF of an optical instrument is the Fourier Transform of the auto-correlation of its pupil, the PSF of any instrument with a finite aperture is bandwidth-limited. According to the Shannon sampling theorem, the PSF can therefore be perfectly reconstructed (interpolated) from an infinite table of regularly-spaced samples. For a finite table, the reconstruction will not be perfect: extended features, such as profile wings and diffraction spikes caused by the high frequency component of the pupil function, will obviously be cropped. With this limitation in mind, one may nevertheless reconstruct with good accuracy a tabulated PSF thanks to sinc interpolation (e.g. Lupton & Gunn 1986). Undersampled PSFs can also be represented in the form of tabulated data provided that a finer grid satisfying Nyquist's criterion is used (Anderson & King 2000, Mighell 2005).

For reasons of flexibility and interoperability with other software, we chose to represent PSFs in PSFEx as small images with adjustable resolution. These PSF "images" can be either derived directly, treating each pixel as a free parameter ("pixel" vector basis), or more generally as a combination of basis vector images.

6.3.1 Pixel basis

The pixel basis is selected by setting `BASIS_TYPE` to `PIXEL`.

Recovering aliased PSFs If the data are undersampled, unaliased Fourier components can in principle be recovered from the images of several point sources randomly located with respect to the pixel grid, using the principle of super-resolution (Huang & Tsai 1984). Working in the Fourier domain, Lauer (1999) shows how PSFs from the Hubble Space Telescope Planetary Camera and Wide-Field Planetary Camera can be reconstructed at 3 times the original instrumental sampling from a large number of undersampled star images. However, solving in the Fourier domain gives far from satisfactory results with real data. Images have boundaries; the wings of point source profiles may be contaminated with artifacts or background sources; the noise process is far from stationary behind point sources with high S/N, because of the local photon-noise contribution from the sources themselves. All these features generate spurious Fourier modes in the solution, which appear as parasitic ripples in the final, super-resolved PSF.

A more robust solution is to work directly in pixel space, using an interpolation function; we may use the same interpolation function later on to *fit* the tabulated PSF model for point source photometry. Let $\boldsymbol{\phi}$ be the vector representing the tabulated PSF, $h_s(\boldsymbol{x})$ an interpolation function, η the ratio of the PSF sampling step to the original image sampling step (oversampling factor). The interpolated value at image pixel i of $\boldsymbol{\phi}$ centered on coordinates $\boldsymbol{x}_s$ is

$$\phi'_i(\boldsymbol{x}_s) = \sum_j h_s\left(\boldsymbol{x}_j - \eta\left(\boldsymbol{x}_i - \boldsymbol{x}_s\right)\right)\phi_j \tag{2}$$

Note that η can be less than 1 in the case where the PSF is oversampled. Using multiple point sources s sharing the same PSF, but centred on various coordinates $\boldsymbol{x}_s$, and neglecting the correlation of noise between pixels, we can derive the components of ϕ that provide the best fit (in the least-square sense) to the point source images by minimising the cost function:

$$E(\boldsymbol{\phi}) = \chi^2(\boldsymbol{\phi}) = \sum_s \sum_{i\in\mathcal{D}_s} \frac{\left(p_i - f_s\phi'_i(\boldsymbol{x}_s)\right)^2}{\sigma_i^2}, \tag{3}$$

where f_s is the integrated flux of point source s, p_i the pixel intensity (number of counts in ADUs) recorded above the background at image pixel i, and $\mathcal{D}_s$ the set of pixels around s. In the variance estimate of pixel i, σ_i^2, we identify three contributions:

$$\sigma_i^2 = \sigma_\mathrm{b}^2 + \frac{p_i}{g} + (\alpha\, p_i)^2\ , \tag{4}$$

where σ_b^2 is the pixel variance of the local background, p_i/g, where g is the detector gain in e^-/ADU (which must have been set appropriately before running SExtractor, is the variance contributed by photons from the source itself. The third term in equation (4) will generally be negligible except for high p_i values; the α factor accounts for pixel-to-pixel uncertainties in the flat-fielding, variation of the intra-pixel response function, and apparent fluctuations of the PSF due to interleaved "micro-dithered" observations[9] or lossy image resampling. The value of α is set by user with the `PSF_ACCURACY` configuration parameter. Depending on image quality, suitable values for `PSF_ACCURACY` range from less than one thousandth to 0.1 or even more. The default value, 0.01, should be appropriate for typical CCD images.

The flux f_s is measured by integrating over a defined aperture, which defines the normalisation of the PSF. Its diameter must be sufficiently large to prevent the measurement from being too sensitive to centering or pixelisation effects, but not excessively large to avoid too strong S/N degradation and contamination by neighbours. In practice, a $\approx 5''$ diameter provides a fair compromise with good seeing images (PSF FWHM $< 1.2''$), but smaller in very crowded fields.

Interpolating the PSF model As we saw, one of the main interests of interpolating the PSF model in direct space is that it involves only a limited number of PSF "pixels". However, as in any image resampling task, a compromise must be found between the perfect Shannon interpolant (unbounded sinc function), and simple schemes with excessive smoothing and/or aliasing properties like bi-linear interpolation ("tent" function) (see Wolberg 1992). Experimenting with the SWarp[10] image resampling prototype (Bertin et al. 2002), we found that the Lanczos4 interpolant

$$h(x) = \begin{cases} 1 & x = 0 \\ \mathrm{sinc}(x)\,\mathrm{sinc}\,(x/4) & 0 < |x| \leq 4 \\ 0 & |x| > 4 \end{cases}, \qquad (5)$$

where $\mathrm{sinc}(x) = \sin(\pi x)/(\pi x)$[11], provides reasonable compromise: the kernel footprint is 8 PSF pixels in each dimension, and the modulation transfer function is close to flat up to $\approx 60\%$ of the Nyquist frequency (Fig. 5. A typical minimum of 2 to 2.5 pixels per PSF Full-Width at Half-Maximum (FWHM) is required to sample an astronomical image without generating significant aliasing (see Bernstein 2002). Consequently, an appropriate sampling step for the PSF model would be 1/4th of the PSF FWHM. This is automatically done in PSFEx, when the `PSF_SAMPLING` configuration parameter is set to 0 (the default). The PSF sampling step may be manually adjusted (in units of image pixels) by simply setting `PSF_SAMPLING` to a non-zero value.

Regularisation For $\eta \gg 1$, the system of equations obtained by minimising equation (3) becomes ill-conditioned and requires regularisation (Pinheiro da Silva et al. 2006). Our experience with PSFEx shows that the solutions obtained over the domain of interest for astronomical imaging ($\eta \leq 3$) are robust in practice, and that regularisation is generally not needed. However, it may happen, especially with infrared detectors, that samples of undersampled point sources are contaminated by image artifacts; and solutions computed with equation (3) become

[9] Micro-dithering consists of observing n^2 times the same field with repeated $1/n$ pixel shifts in each direction to provide properly sampled images despite using large pixels. Although the observed frames can in principle be recombined with an "interleaving" reconstruction procedure, changes in image quality from exposure to exposure may often lead to jaggies (artifacts) along gradients of source profiles, as can sometimes be noticed in DeNIS or WFCAM images.

[10] Coaddition of images, `http://www.astromatic.net/swarp`.

[11] This is the definition of the *normalised* sinc function, which the reader should not confuse with the unnormalised definition of $\mathrm{sinc}(x) = \sin(x)/x$.

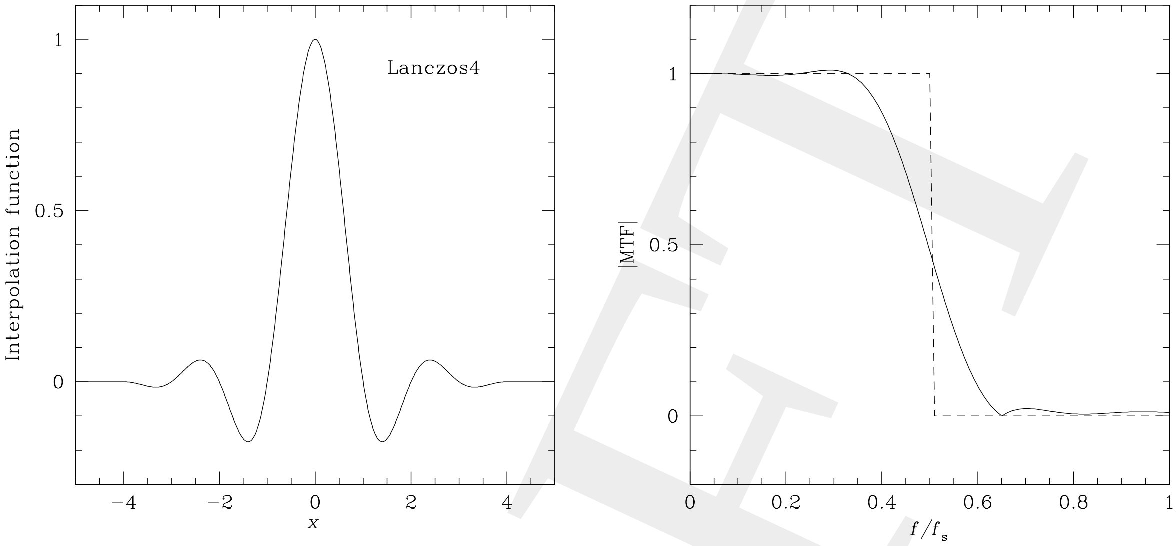


Figure 5: The Lanczos4 interpolant in one dimension (*left*), and its modulation transfer function (*right*).

unstable. We therefore added a simple Tikhonov regularisation scheme to the cost function:

$$E(\boldsymbol{\phi}) = \chi^2(\boldsymbol{\phi}) + \|\mathrm{T}\boldsymbol{\phi}\|^2. \tag{6}$$

In image processing problems the (linear) Tikhonov operator T is usually chosen to be a high-pass filter to favour "smooth" solutions. PSFEx adopts a slightly different approach by reducing T to a scalar weight $1/\sigma_\phi^2$ and performing a procedure in two steps.

1. PSFEx makes a first rough estimate of the PSF by simply shifting point sources to a common grid and computing a median image $\boldsymbol{\phi}^{(0)}$. With undersampled data this image represents a smooth version of the real PSF.

2. Instead of fitting directly the model to pixel values, PSFEx fits the difference $\Delta\phi$ between the model and $\boldsymbol{\phi}^{(0)}$. $E(\boldsymbol{\phi})$ becomes

$$E(\boldsymbol{\phi}) = \sum_s \sum_{i\in\mathcal{D}_s} \frac{\left[p_i - f_s\left(\phi_i'^{(0)}(\boldsymbol{x}_s) + \Delta\phi_i'(\boldsymbol{x}_s)\right)\right]^2}{\sigma_i^2} + \sum_j \frac{\Delta\phi_j^2}{\sigma_\phi^2}. \tag{7}$$

Minimising equation (7) with respect to the $\Delta\phi_j$'s comes down to solving the system of equations

$$\begin{aligned} 0 &= \frac{\partial E}{\partial \Delta\phi_k} \\ &= 2\, f_s \sum_s \sum_{i\in\mathcal{D}_s} \frac{1}{\sigma_i^2} h_s\left(\boldsymbol{x}_k - \eta\left[\boldsymbol{x}_i - \boldsymbol{x}_s\right]\right) \\ &\quad \times \left(f_s \sum_j h_s\left(\boldsymbol{x}_j - \eta\left[\boldsymbol{x}_i - \boldsymbol{x}_s\right]\right)(\phi_j^{(0)} + \Delta\phi_j) - p_i\right) \\ &\quad + \frac{2}{\sigma_\phi^2}\,\Delta\phi_k\ . \end{aligned} \tag{8}$$

In practice the solution appears to be fairly insensitive to the exact value of σ_ϕ except with low signal-to-noise conditions or contamination by artifacts. $\sigma_\phi \approx 10^{-2}$ seems to provide a good compromise by bringing efficient control of noisy cases but no detectable smoothing of PSFs with good data and high signal-to-noise.

The system in equation (8) is solved by PSFEx in a single pass. Much of the processing time is actually spent in filling the normal equation matrix, which would be prohibitive for large PSFs if the sparsity of the design matrix were not put to contribution to speed up computations.

6.3.2 Other bases

The pixel basis is quite a "natural" basis for describing in tabulated form bandwidth-limited PSFs with arbitrary shapes. But in a majority of cases, more restrictive assumptions can be made about the PSF that allow the model to be represented with a smaller number of components, e.g. a bell-shaped profile, a narrow scale range... Less basis vectors make for more robust models. For close-to-Gaussian PSFs, the Gauss-Laguerre basis (see §6.3.2) is a sensible choice.

With the `BASIS_TYPE FILE` option, PSFEx offers the possibility to use an external image vector basis. The basis should be provided as a FITS datacube (the 3rd dimension being the vector index), and the file name given to PSFEx with the `BASIS_NAME` parameter. External bases do not need to be normalised.

Gauss-Laguerre basis The Gauss-Laguerre basis is selected by setting `BASIS_TYPE` to be `GAUSS-LAGUERRE`. The Gauss-Laguerre functions, also known as *polar shapelets* in the weak-lensing community (Massey & Refregier 2005) provide a "natural" orthonormal basis for broadly Gaussian profiles:

$$\psi_{n,m}(r,\theta) = \frac{(-1)^{(n-|m|)/2}}{\sqrt{\pi}\,\sigma}\sqrt{\frac{[(n-|m|)/2]!}{[(n+|m|)/2]!}}\left(\frac{r}{\sigma}\right)^{|m|}\exp\left[-\frac{1}{2}\left(\frac{r}{\sigma}\right)^2 - im\theta\right] L^{|m|}_{(n-|m|)/2}\left(\frac{r^2}{\sigma^2}\right), \quad (9)$$

where σ is a typical scale for r, $(n-|m|)/2 \in \mathbb{N}$ and $L_n^k(x)$ is the associated Laguerre polynomial

$$\begin{aligned} L_n^k(x) &= \frac{1}{n!}\,x^{-k}\,\mathrm{e}^x\,\frac{\mathrm{d}^n}{\mathrm{d}x^n}\left(x^{n+k}e^{-x}\right) & (10)\\ &= \sum_{j=0}^{n}\frac{(k+n)!}{j!\,(n-j)!\,(j+k)!}\,(-x)^j. & (11)\end{aligned}$$

The number of shapelet vectors with $n \le n_{\max}$ is

$$N_{\max} = \frac{(n_{\max}+1)\,(n_{\max}+2)}{2}\ . \quad (12)$$

Shapelet decompositions with finite $n \le n_{\max}$ are only able to probe a restricted range of scales. Refregier (2003) quotes $r_{\min} = \sigma/\sqrt{n_{\max}+1}$ and $r_{\max} = \sigma\sqrt{n_{\max}+1}$ as the standard deviation of the central lobe and the whole shapelet profile, respectively (so that σ is the geometric mean of $r_{\min}$ and $r_{\max}$). In practice, the diameter of the circle enclosing the region where images can be fitted with shapelets is only about $\approx 2.5\,r_{\max}$. Hence modelling accurately both the wings and the core of PSFs with a unique set of shapelets requires a very large number of shapelet vectors, typically several hundreds.

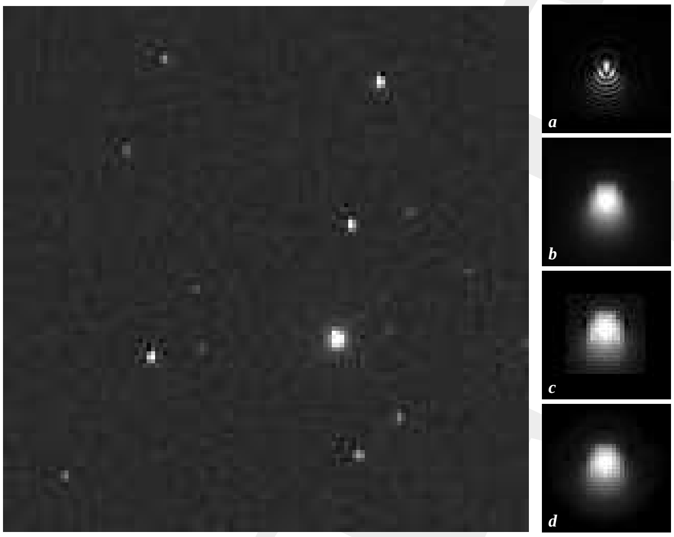


Figure 6: Example of an extreme case of PSF recovery. *Left*: part of a simulated star field image with strong undersampling. *Right, from top to bottom*: (a) simulated optical PSF, (b) simulated PSF convolved by the pixel response, (c) PSF recovered by PSFEx at 4.5 times the image resolution from a random sample of 212 stars extracted in the simulated field above, using the "pixel" vector basis (§6.3.1), and (d) PSF recovered using the "shapelet" basis (§6.3.2) with $n_{\max} = 16$.

7 Managing PSF variations

7.1 Basic formalism

Few imaging systems have a perfectly stable PSF, be it in time or position: for most instruments the approximation of a constant PSF is valid only on a small portion of an image at a time. Position-dependent variations of the PSF on the focal plane are generally caused by optics, and exhibit a smooth behaviour which can be modelled with a low-order polynomial.

The most intuitive way to generate variations of the PSF model is to apply some warping to it (enlargement, elongation, skewness, ...). But this description is not appropriate with PSFEx because of the non-linear dependency of PSF vector components towards warping parameters. Instead, one can extend the formalism of equation (7) by describing the PSF as a variable, linear combination of PSF vectors $\boldsymbol{\phi}_c$; each of them associated to a basis function X_c of some parameter vector $\boldsymbol{p}$ like image coordinates:

$$E(\boldsymbol{\phi}) = \sum_s \sum_{i \in \mathcal{D}_s} \frac{\left(p_i - f_s \sum_c X_c(\boldsymbol{p}) \left(\phi'^{(0)}_{c\,i}(\boldsymbol{x}_s) + \Delta\phi'_{c\,i}(\boldsymbol{x}_s)\right)\right)^2}{\sigma_i^2} + \sum_j \sum_c \frac{\Delta\phi^2_{c\,j}}{\sigma^2_\phi}\ . \qquad (13)$$

The basis functions X_c in the current version of PSFEx are limited to simple polynomials of the components of $\boldsymbol{p}$. Each of these components p_l belongs to a "PSF variability group" $g = 0, 1, ..., N_g$, such that

$$X_c(\boldsymbol{p}) = \prod_{g \leq N_g} \left(\prod_{(\sum_{l \in \Lambda_g} d_l) \leq D_g} p_l^{d_l} \right), \qquad (14)$$

where Λ_g is the set of l's that belongs to the distortion group g, and $D_g \in \mathbb{N}$ is the polynomial degree of group g. The polynomial engine of PSFEx is the same as the one implemented in the SCAMP software (Bertin 2006) and can use any set of SExtractor and/or FITS header parameters as components of p. Although PSF variations are more likely to depend essentially on source position on the focal plane, it is thus possible to include explicit dependency on parameters such as telescope position, time, source flux (Fig. 8) or instrument temperature.

The p_l components are selected using the `PSFVAR_KEYS` configuration parameter. The arguments can be names of SExtractor measurements, or keywords from the image FITS header representing numerical values. FITS header keywords must be preceded with a colon (:), like in `:AIRMASS`. The default `PSFVAR_KEYS` are `X_IMAGE,Y_IMAGE`.

The `PSFVAR_GROUPS` configuration parameters must be filled in in combination with the `PSFVAR_KEYS` to indicate to which PSF variability group each component of $\boldsymbol{p}$ belongs. The default for `PSFVAR_GROUPS` is `1,1`, meaning that both `PSFVAR_KEYS` belong to the same unique PSF variability group. The polynomial degrees D_g are set with `PSFVAR_DEGREES`. The default `PSFVAR_DEGREES` is `2`. In practice, a third-degree polynomial on pixel coordinates (represented by 20 PSF vectors) should be able to map PSF variations with good accuracy on most exposures (Fig. 7).

8 Quality assessment

Maintaining a certain level of image quality, and especially PSF quality, by identifying and rejecting "bad" exposures, is a critical issue in large imaging surveys. Image control must be automated, not only because of the sheer quantity of data in modern digital surveys, but also to ensure an adequate level of consistency. Automated PSF quality assessment is traditionally based upon point source FWHM and ellipticity measurements. Although this is certainly efficient for finding fuzzy or elongated images, it cannot make the distinction between e.g. a defocused image and a moderately bad seeing.

PSFEx can trace out the apparition of specific patterns using customized basis functions. Moreover, PSFEx implements a series of generic quality measurements performed on the PSF model as it varies across the field of view. The main set of measurements is done in PSF pixel space (with oversampling factor η) by comparing the actual PSF model vector $\boldsymbol{\phi}$ with a reference PSF model $\rho(\boldsymbol{x}')$. We adopt as a reference model the elliptical Moffat (1969) function that fits best (in the chi-square sense) the model:

$$\rho(\boldsymbol{x}') = I_0 \left(1 + \left|\left|\mathrm{A}(\boldsymbol{x}' - \boldsymbol{x}'_c)\right|\right|^2\right)^{-\beta}, \tag{15}$$

with

$$\mathrm{A} = \frac{4}{\eta}\,(2^{-\frac{1}{\beta}} - 1) \begin{pmatrix} \cos\theta/W_{\max} & \sin\theta/W_{\max} \\ -\sin\theta/W_{\min} & \cos\theta/W_{\min} \end{pmatrix}, \tag{16}$$

where I_0 is the central intensity of the PSF, $\boldsymbol{x}'_c$ the central coordinates (in PSF pixels), $W_{\max}$, the PSF FWHM along the major axis, $W_{\min}$ the FWHM along the minor axis, and θ the position angle (6 free parameters). As a matter of fact, the Moffat function provides a good fit to seeing-limited images of point-sources, and to a lesser degree, to the core of diffraction-limited images for instruments with circular apertures (Trujillo et al. 2001): in most imaging surveys, the "correct" instrumental PSF will be very similar to a Moffat function with low ellipticity.

Since PSFEx is meant to deal with significantly undersampled PSFs, another fit — which we call "pixel-free" — is also performed, where the Moffat model is convolved with a square top-hat function the width of a physical pixel, as an approximation to the real intra-pixel response function. The width of the pixel is set to 1 in image sampling step units by default, which corresponds to a 100% fill-factor. It can be changed using the `PSF_PIXELSIZE` configuration

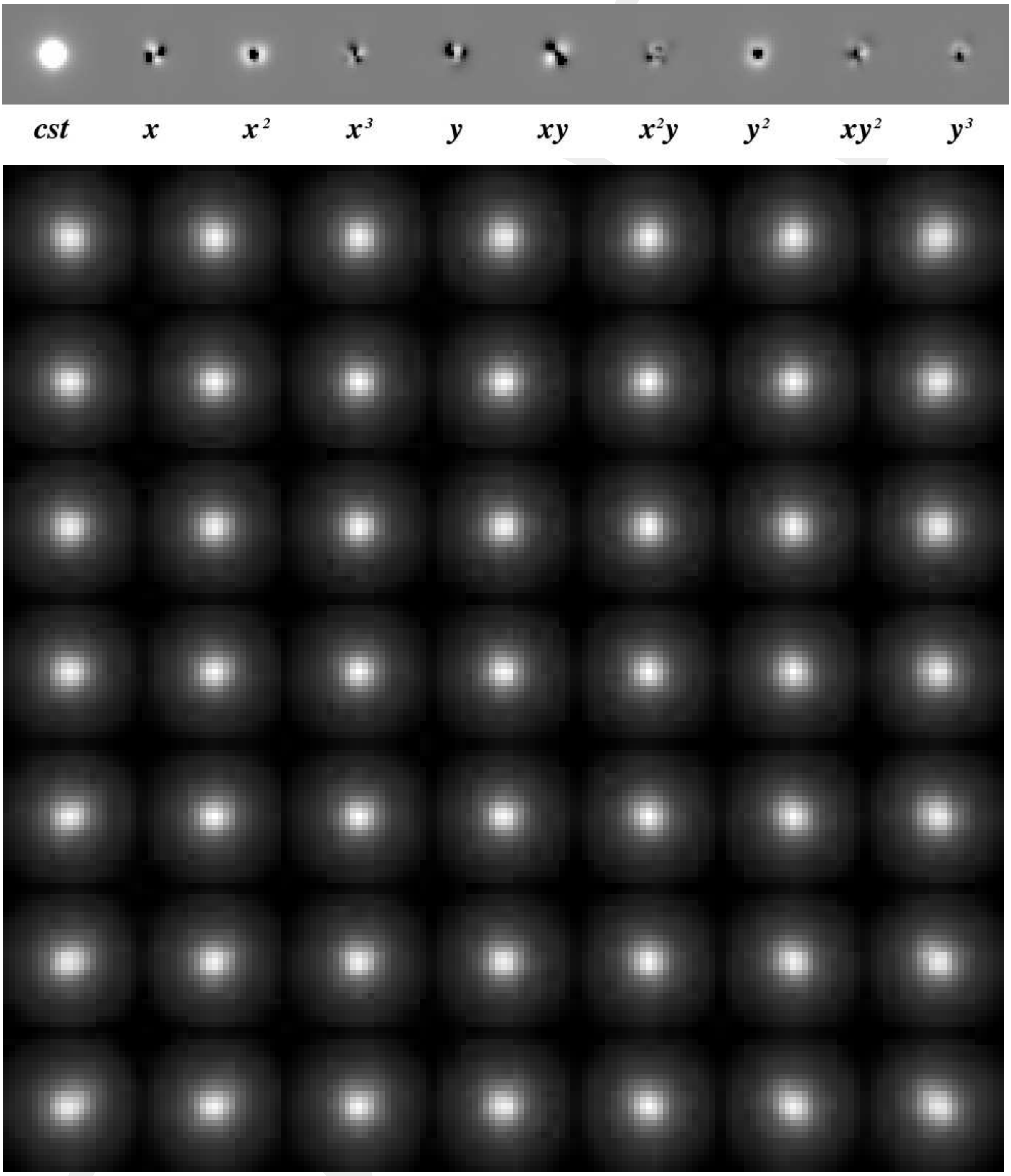


Figure 7: Example of PSF mapping as a function of pixel coordinates in PSFEx. *Top*: PSF component vectors for each polynomial term derived from the CFHTLS-deep "D4" r-band stack observed with the MEGACAM camera. A third-degree polynomial was chosen for this example. Note the prominent variation of PSF width with the square of the distance to the field centre. *Bottom*: reconstruction of the PSF over the 1° field of view (the grey scale has been slightly compressed to improve clarity).

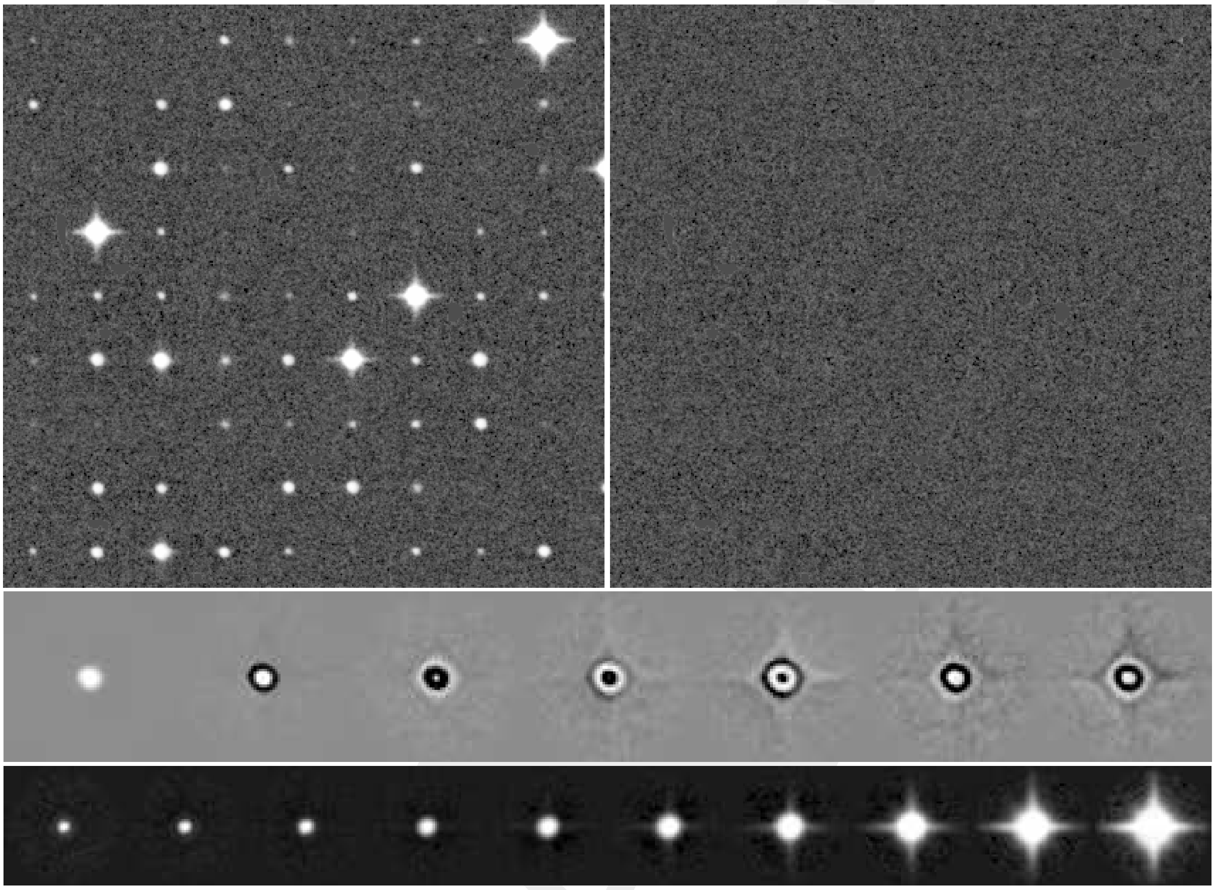


Figure 8: Example of PSF mapping on images from a non-linear imaging device. 1670 point sources from the central 4096×4096 pixels of a photographic scan (SERC J #418 survey plate, courtesy of J. Guibert, CAI, Paris observatory) were extracted using SExtractor, and their images run through PSFEx. A sample is shown at the *top-left*. The PSF model was given a 6$^{\rm th}$ degree polynomial dependency on the instrumental magnitude measured by SExtractor (`MAG_AUTO`). *Middle*: PSF components derived by PSFEx. *Bottom*: reconstructed PSF images as a function of decreasing magnitude. *Top-right*: sample residuals after subtraction of the PSF-model.

parameter. Future versions of PSFEx might propose more sophisticated models of the intra-pixel response function.

The (non-linear) fits are performed using the LevMar implementation of the Levenberg-Marquardt algorithm (Lourakis 2004+). They are repeated at regular intervals on a grid of PSF parameter vectors $\boldsymbol{p}$, generally composed of the image coordinates $\boldsymbol{x}$), but with possible additional parameters such as time, observing conditions, etc. The density of the grid may be adjusted using the `PSFVAR_NSNAP` configuration parameter. The default value for `PSFVAR_NSNAP` is 9 (snapshots per component of $\boldsymbol{p}$). Larger numbers can be useful to track PSF variations on large images with greater accuracy; but beware of the computing time, which increases as the total number of PSF snapshots (grid points).

The average FWHM $(W_{\rm max} + W_{\rm min})/2$, ellipticity $(W_{\rm max} - W_{\rm min})/(W_{\rm max} + W_{\rm min})$ and β parameters derived from the fits provide a first set of local IQ estimators (Fig. 9). The second set is composed of the so-called *residuals* index

$$r = 2\frac{\sum_i \left(\phi_i + \rho'(\boldsymbol{x}'_i)\right) |\phi_i - \rho'(\boldsymbol{x}'_i)|}{\sum_i \left(\phi_i + \rho'(\boldsymbol{x}'_i)\right)^2} \tag{17}$$

and the *asymmetry* index

$$\alpha = 2\frac{\sum_i \left(\phi_i + \phi_{N-i}\right) |\phi_i - \phi_{N-i}|}{\sum_i \left(\phi_i + \phi_{N-i}\right)^2}, \tag{18}$$

where the ϕ_{N-i}'s are the point-symmetric counterparts to the ϕ_i components.

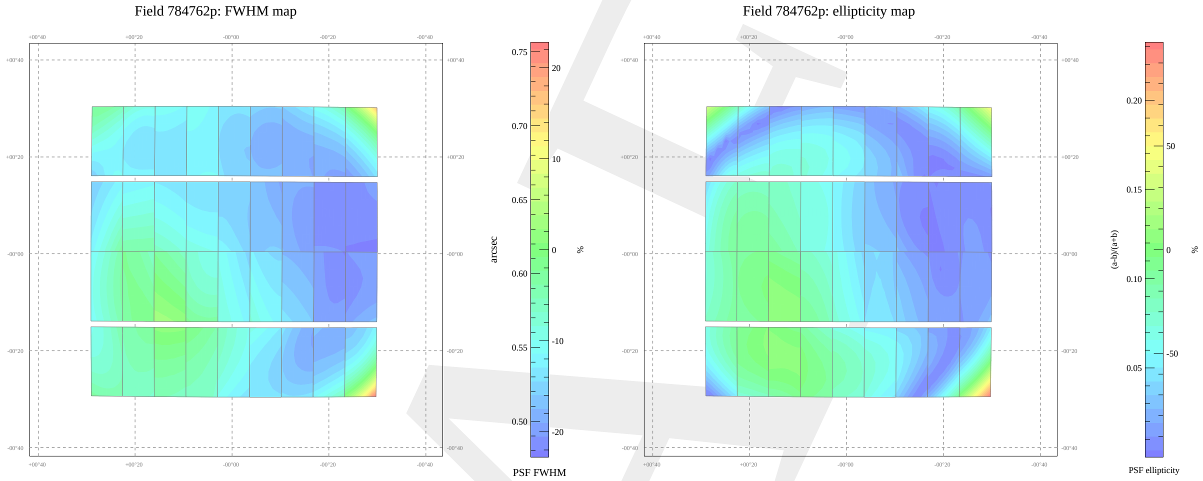


Figure 9: Colour-coded maps of the PSF FWHM (*left*) and ellipticity (*right*) generated by PSFEx from a CFHTLS-Wide exposure The map and the individual Megaprime CCD footprints on the sky are presented in gnomonic projection (north is on top, east on the left). PSF variations are modelled independently on each CCD using a 3$^{\text{rd}}$ degree polynomial (see text).

9 Examples

In the following, examples of use of PSFEx are given, together with commented command lines.

9.1 Example 1: wide field mosaic

TBW

9.2 Example 2: very wide photographic plate

TBW

9.3 Example 3: unfocused instrument

TBW

10 FAQs

TBW

11 Troubleshooting

TBW

12 Acknowledging PSFEx

Please use the following reference: **Bertin E. et al.**, 2010, in preparation.

13 Acknowledgements

The authors would like to thank Mireille Dantel, Frédéric Magnard, Chiara Marmo, Gregory Sémah, and the TERAPIX team at IAP for testing and support on image quality indices, Shantanu Desai, Tony Darnell, Greg Daues, Joe Mohr and the Dark Energy Survey Management team at University of Illinois and NCSA, for testing and support on PSF homogenisation, Philippe Delorme for his contributions to PSF-fitting in SExtractor, Valérie de Lapparent , Pascal Fouqué, and Jason Kalirai for extensive testing and suggestions, Mark Calabretta for his great astrometric library, Manolis Lourakis for making his LevMar library public, Akim Demaille for his help with the autotools, and Gary Mamon for his careful reading and corrections to the manuscript.

References

[1] Anderson J., King I.R., 2000, PASP, 112,1360

[2] Bernstein G., 2002, PASP, 114, 98

[3] Bertin E., 1999, *SExtractor 2.1, User's manual*, IAP

[4] Bertin E., Mellier Y., Radovich M., Missonnier.G., Didelon P., Morin B., 2002, Astronomical Data Analysis Software and Systems XI, A.S.P. Conference Series, 281, 228

[5] Huang T.S., Tsai R.Y., 1984, Advances in Computer Vision and Image Processing 1, 317

[6] Jee M.J., Blakeslee J.P., Sirianni M., Martel A.R., White R.L., Ford H.C., 2007, PASP, 119,1403

[7] Kaiser N., Squires G., Broadhurst T., 1995, ApJ, 449, 460

[8] Lauer T.R., 1999, PASP, 111, 227

[9] Lourakis M.I.A., 2004+, `http://www.ics.forth.gr/~lourakis/levmar`

[10] Lupton R.H., Gunn J.E., 1986, AJ, 91, 1106

[11] Lupton R., Gunn J.E., Ivezić Z., Knapp G.R., Kent S., 2001, in Astronomical Data Analysis Software and Systems X, A.S.P. Conference Series, 238, 269

[12] Magain P., Courbin F., Gillon M., Sohy S., Letawe G., Chantry V., Letawe Y., 2007, A&A, 461, 373

[13] Massey R., Refregier A., 2005, MNRAS, 363, 197

[14] Mighell K.J., 2005, MNRAS, 361, 861

[15] Moffat A.FJ., 1969, A&A 3, 455

[16] Pinheiro da Silva L., Auvergne M., Toublanc D., Rowe J., Kuschnig R., Matthews J., 2006, A&A, 452, 363

[17] Refregier A., 2003, MNRAS, 338,35

[18] Stetson P.B., 1987, PASP, 99, 191

[19] Theodoridis S., Koutroumbas K., 2003,

[20] Trujillo I., Aguerri J.A.L.,Cepa J., Gutiérrez C.M., 2001, MNRAS, 328, 977

[21] Wolberg G., 1992, Digital Image Warping, IEEE Computer Society Press