Internal release of software tools

T-PHOT and gencat

INAF – OAR for the Astrodeep project

ASTRODEEP

"Unveiling the power of the deepest images of the Universe"

THEME [SPA.2012.2.1-01]

[Exploitation of space science and exploration data]

Grant agreement for: Collaborative project

Grant agreement no: 312725

ABSTRACT

In this document we present the software tools that were developed by the Astrodeep collaboration. More specifically, T-PHOT that is the evolution of CONVPHOT and implements also a set of improvements taken from TFIT, and gencat that was used for the creation of simulated images. Both software have been internally released and extensively used by the collaboration for obtaining simulated images and catalogs.
Deliverable Number D3.4 – Delivery date June 2015.

Prepared by:, E. Merlin, A. Fontana, D. Elbaz, C. Schreiber
Approved by: AEC
Date: 05/06/2015

T-PHOT: the most advanced tool for image de-confusion

T-PHOT is the first software tool delivered to the public from the AstroDeep project. Mainly developed at INAF-OAR by Emiliano Merlin, T-PHOT is a software package aimed at precision photometry of extragalactic multiwavelength datasets, using the PSF-matching technique to de-confuse the images and obtain reliable photometric flux measurements even for blended sources. T-PHOT stems from its direct predecessors TFIT (Laidler et al. 2007) and CONVPHOT (De Santis et al. 2007), and merges their features into a faster and more robust code. On top of that, T-PHOT improves on such codes, both in terms of performance and in the accuracy of the results. It also incorporates a number of new options and features to speed up the computation, remove potential sources of errors, and extend the range of cases in which it is possible to use the code. T-PHOT is therefore a completely new and versatile tool, suitable for performing detailed photometry on images taken in a very broad range of wavelengths, not only in the optical domain but also in the FIR and sub-mm regimes where its performance is comparable to, or better than, other existing codes (e.g. FASTPHOT by Bethermin et al., or DESPHOT by Roseboom et al.).

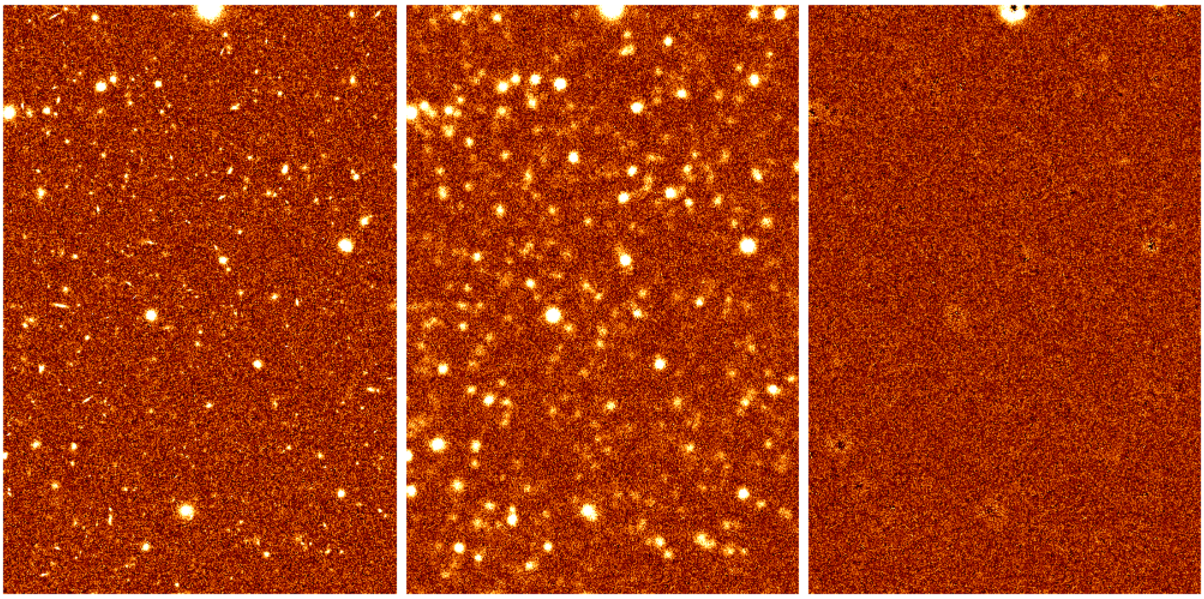


Figure 1: Example of the results of a standard T-PHOT run using real priors. Left to right: HRI (FWHM = 0.2"), LRI (FWHM = 1.66"), and residuals image for a simulated dataset. LRI and residual image are on the same intensity scale

In its pipeline, T-PHOT goes through well-defined stages, in each of which a single task is performed. It uses high-resolution priors to determine the positions and, when possible, the morphological information of the sources, and then uses this information to measure the fluxes of those sources in a lower resolution image (LRI). An example of T-PHOT results is shown in Fig.1. T-PHOT accepts three different kinds of priors: a catalog of sources from a high resolution image (HRI), and/or analytical 2-d models obtained e.g. using Galfit (Peng et al. 2010), and/or a catalog of positions for unresolved point-sources (common practice or FIR and sub-mm band-passes). Note that the use of mixed priors is allowed, making it possible to e.g. remove foreground bright sources by modeling them as analytical 2-d profiles and simultaneously fitting them along with standard "real" cutouts (this is the procedure that has been followed to obtain K and IRAC catalogs on the Frontier Fields cluster images, see D4.3).

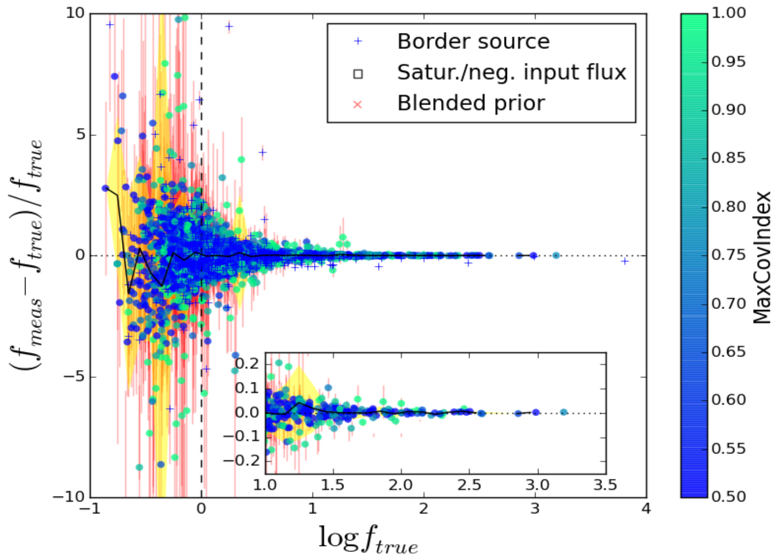


Figure 2: Accuracy of the flux determination in a simulation containing non-overlapping, PSF-shaped sources and with "perfect" detection. Relative measured flux difference (f_meas-f_true)/f_true is plotted versus logarithm of the input flux f_true, for a simulated image populated with PSF-shaped sources (FWHM=1.66"). Each dot corresponds to a single source, with different symbols and colors referring to various diagnostics as explained in the legend and in the colorbar. The black solid line is the average in bins, the yellow shade is the standard deviation. The vertical dashed line shows the limiting flux at 1σ, f=1. The inner panel shows a magnification of the brighter end of the distribution. The fit has been performed on the whole image at once.

A normalized low-resolution model (*template*) of each object is created degrading its HRI cutout, or its model profile, using a PSF-matching kernel - or just the LRI PSF if unresolved priors are used. Then, to overcome the problem of the blending of sources, a Chi-square minimization problem is solved, fitting all the sources at once in a chosen region. The fit can be performed on the whole LRI, giving the most reliable results, or constructing "cells" around each source including all its potentially contaminating neighbors in the fit. The standard TFIT approach, consisting in dividing the LRI in a regular, arbitrary grid of cells, is still allowed but it is strongly discouraged, since it has proven to introduce non negligible errors due to the potential contamination from the light coming from objects just outside the considered cell.
Nominal statistical uncertainties are assigned to each measurement from the covariance matrix of the problem. However, systematic errors may affect the measurements in some particular cases (e.g., saturated or blended priors and border sources): in such cases, a flag in the output catalogue highlights the problem. Also, strongly covariant objects can have badly measured fluxes: a "covariance index" offers a qualitative indication about this risk.

After the fitting stage, T-PHOT can perform a spatial cross-correlation between the LRI and the model image constructed from the templates, to obtain locally registered kernels which can be used for a second pass in order to minimize spatial inaccuracies. The main final products of the run are a catalogue including positions, measured fluxes, uncertainties and diagnostic flags, and residual image obtained subtracting the model image from the LRI, useful to check at a glance the overall goodness of the fit. Other sub-products and diagnostics are also produced. The accuracy of flux determination with T-PHOT in a simulation is shown in Fig.2 and the comparison to the TFIT residuals is shown in Fig.3.

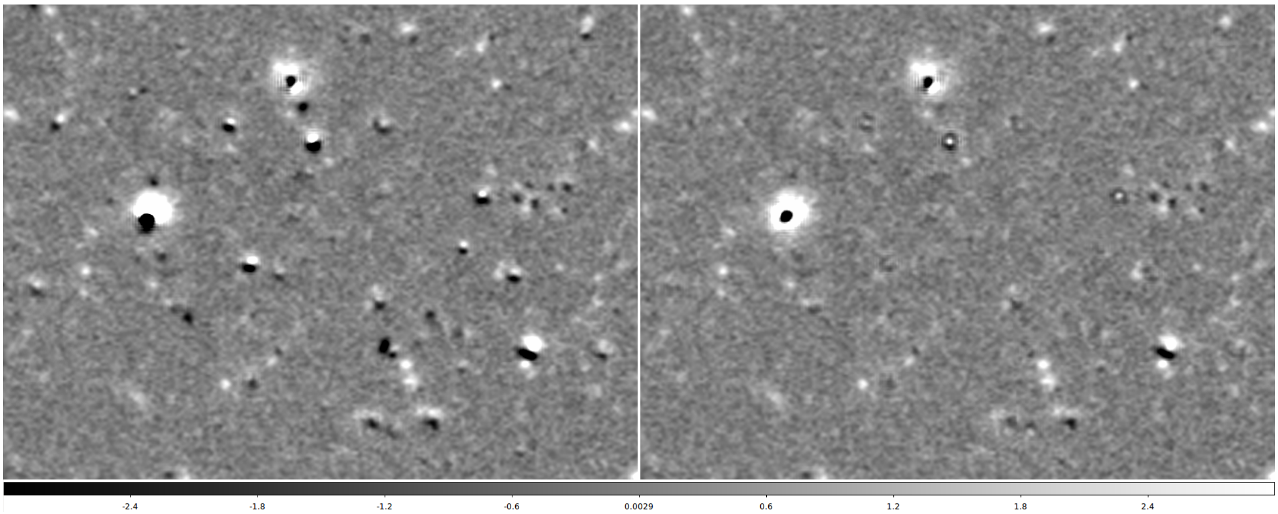


Figure 3: UDS I band TFIT vs. T-PHOT comparison. The panel on the left shows a small patch of the "official" CANDELS residual image obtained using TFIT. The residual image of the same regions is showed in the right panel, this time obtained using T-PHOT with "cells-on-objects" method and improved local kernel registration. Note the disappearance of many spurious black spots

We checked the performance of T-PHOT on a wide set of simulated data and on real datasets. T-PHOT proves to be much faster than its predecessors, up to a factor of hundreds in the most favorable situation; it can deal with large datasets with, and to give more accurate results with an appropriate choice of the input parameters.

For all these reasons, T-PHOT is already being used by many researchers both within the ASTRODEEP consortium and not. In particular, it is currently being used to obtain photometric catalogues of K and IRAC Frontier Fields images (see D.43) and SCUBA CANDELS images (see D3.3), and will be used in the near future to re-analyze IRAC CANDELS GOODS-S data.

T-PHOT has been presented and advertised by E. Merlin at the ADASS XXIV conference held in Calgary (CA) in October 2014, and at the conference "The spectral energy distribution of high redshift galaxies" held in Sexten (IT) in January 2015.

The paper "T-PHOT: a new code for PSF-matched, prior-based, multiwavelength extragalactic de-confusion photometry" (E. Merlin et al.) has been submitted to A&A, and is attached to this document.

T-PHOT comes as a tarball including documentation, installation scripts, and the source code. It consists of Python envelopes calling fast C and C++ codes, only needing a few external dependencies (some standard Python modules, the CFITSIO and FFTW3 libraries). It is easy to install and to use on UNIX and MAC-OS, with a user-friendly parameter file and a straightforward command line from a terminal.

T-PHOT is a public software and can be downloaded from the ASTRODEEP website by subscribing to a mailing list. New releases are planned in the near future, with the inclusion of additional options to make it even further versatile and appealing to the scientific community.

Generating mock catalogs with gencat

1 Introduction

Up until now we have been using the *SkyMaker*[1] program (E. Bertin) to produce realistic high resolution images, which we regularly use to test the various source extraction methods and algorithms we are developing within Astrodeep. In input, this program requires a simulated galaxy catalog, which is produced by the *Stuff*[2] program (also created by E. Bertin).

The quality of these simulated catalogs is not optimal. In particular, the distribution of the simulated fluxes in some bands differ substantially from those that are observed, leading to simulated images that are not representative of the real products we are working on. Unfortunately, both *SkyMaker* and *Stuff* are poorly documented, and we cannot easily remedy to this problem. For this reason, we have developed a new tool to generate simulated galaxy catalog, called `gencat`[3]. The main ideas behind the procedure are summarized in Section 2.

This new tool can generate catalogs in the format required by *Sky Maker*, and therefore can be used as a "drop-in" replacement for *Stuff*. Using this tool we are able not only to generate fluxes in all the photometric bands from 3000 Å to 8 μm, like *Stuff*, but we also merge in our technique to simulate far-IR fluxes from 8 μm to 3 mm, essentially covering, in a single tool, the whole wavelength range where stellar and dust emission dominate.

Finally, the quality of the generated catalogs has greatly improved compared to original catalogs built with *Stuff*. As can be seen in Section 3, we are able to produce flux distributions in all the bands which are indistinguishable from the real, observed flux distributions. The simulated images, both at *Hubble* and *Herschel*-like resolution, have very good statistical properties. This will allow us to perform more accurate tests of our methods, and also to deliver high quality simulations to the community.

2 Creating the mock catalog

The main idea behind the generation process of this mock catalog is that everything can be statistically inferred from the redshift, the stellar mass and the "star-forming" flag of each galaxy. The procedure is therefore composed of two main steps: first, generate a realistic distribution of masses at different redshifts both for active and passive galaxies using observed mass-functions; second, estimate all the other physical properties using statistical recipes calibrated on the observed galaxies: morphology, SFR, attenuation, optical colors, and sky-projected position.

2.1 Generating redshifts and masses

The purpose of the mock catalog is the simulate a field similar to the GOODS–South CANDELS field. Therefore, in order to most closely mimic the properties of this field, we use the conditional mass functions at different redshifts which are described in Schreiber et al. (2015). Briefly, the whole GOODS–South catalog is cut at $H < 26$ to ensure high completeness, split in two population of "active" and "passive" galaxies according to the UVJ color-color selection, and further split in multiple redshift bins from $z = 0.3$ to $z = 4.5$. These redshifts and stellar masses have been computed by Maurilio Pannella with EAZY and FAST, respectively, on the official CANDELS photometry. We then computed the mass distribution of each of these sub-samples, performing first order completeness corrections, and fit a double Schechter law. Using these fits, we can generate mass functions down to arbitrarily low stellar masses. To reach higher redshifts, we have used the mass functions calculated by Grazian et al. (2015)

[1]http://www.astromatic.net/software/skymaker
[2]http://www.astromatic.net/software/stuff
[3]https://github.com/cschreib/gencat

for $z < 7.5$. The $z = 0$ mass function is adapted from Baldry et al. (2012), but it should not matter much for now since we are aiming for pencil-beam surveys which contain very few local galaxies.

Once this is done, we define a fine grid of redshifts, e.g. from $z = 0.01$ to $z = 6$, and choose the sky area of the mock catalog. For now we work with an area similar to the first catalog produced with Stuff, i.e. 17×17 arcmin. Then for each element of the redshift grid, we use the mass functions to generate a sample of stellar masses. The minimum stellar mass $M_{\min}$ can be chosen either to be constant (e.g. $10^7\,\mathrm{M}_\odot$) or to vary with redshift so as to reach a given magnitude limit in the selection band, for example $H < 27$. This requires using the optical SED library described below to obtain a rough estimate of the mass completeness.

At this stage, the mock catalog has exactly the same mass and redshift distribution as the CANDELS catalog in GOODS–South. This is a good thing to ensure a high fidelity of the simulated catalog, but one has to keep in mind that, by construction, this also means that we have imposed the same cosmic variance than in the real GOODS–South field.

2.2 Generating morphology

The Stuff program was not only generating photometry, but also detailed morphology in each band. In particular, each galaxy is assumed to be composed of two component: a bulge (de Vaucouleur profile, Sérsic $n = 4$) and a disk (exponential profile, Sérsic $n = 1$). In order to be able to plug this new mock catalog in *SkyMaker* directly, we also need to generate these informations.

The first important quantity is the bulge-to-total ratio B/T, which tells what fraction of the total mass of the galaxy goes into the bulge, as opposed to the disk. We generate this quantity using the relations between B/T and M_* published by Lang et al. (2014). These relations are conveniently provided both for active and passive galaxies, at different redshifts. They report no strong redshift evolution between $z = 1$ and $z = 2$, so we chose to make the B/T simply depend on mass following

$$(B/T)_{\mathrm{active}} = 0.2 \times \left(\frac{M_*}{10^{10}}\right)^{0.27} \times 10^{G(0.2)} \text{ and} \tag{1}$$

$$(B/T)_{\mathrm{passive}} = 0.5 \times \left(\frac{M_*}{10^{10}}\right)^{0.1} \times 10^{G(0.2)}\,, \tag{2}$$

where $G(\sigma)$ is a zero-mean Gaussian noise of amplitude σ. The B/T is then clamped to $0 \leq B/T \leq 1$. This quantity will also be used later to define the colors of the galaxies.

The other set of morphological properties we need to generate are the axis ratio, position angle and size of both the disk and the bulge component of each galaxy. We chose to give the same position angle to both components (which is the average trend observed in the morphological catalogs of Simard et al. 2011 for galaxies in the SDSS), and chose it randomly with uniform probability between -90 deg and $+90$ deg.

The axis ratio is generated following the distribution observed in the real catalogs: for the disk (resp. bulge), we built a sample of galaxies with Sérsic index $n < 1.5$ (resp. $n > 2.5$) and computed their axis ratio distribution (Sérsic indices were computed by van der Wel et al. 2014). The result is shown in Fig. 1. As expected, disks-dominated galaxies (blue) are found to be more elongated than bulge-dominated galaxies (red).

To estimate the sizes, we used the same sub-samples as above, and looked at the relation between the observed H-band size, mass, and redshift. We could parametrize the observed relations and their scatter with the following formula

$$R_{\mathrm{disk}} = \begin{cases} (1+z)^{-1.25} \times \left(\frac{M_*}{10^{10}}\right)^{0.17} \times 10^{G(0.2)} & \text{for } z < 1.5, \\ 0.4 \times (1+z)^{-0.25} \times \left(\frac{M_*}{10^{10}}\right)^{0.17} \times 10^{G(0.2)} & \text{for } z > 1.5, \text{ and} \end{cases} \tag{3}$$

$$R_{\mathrm{bulge}} = (1+z)^{-2.5} \times \left(\frac{M_*}{10^{10}}\right)^{0.7} \times 10^{G(0.2)}\,, \tag{4}$$

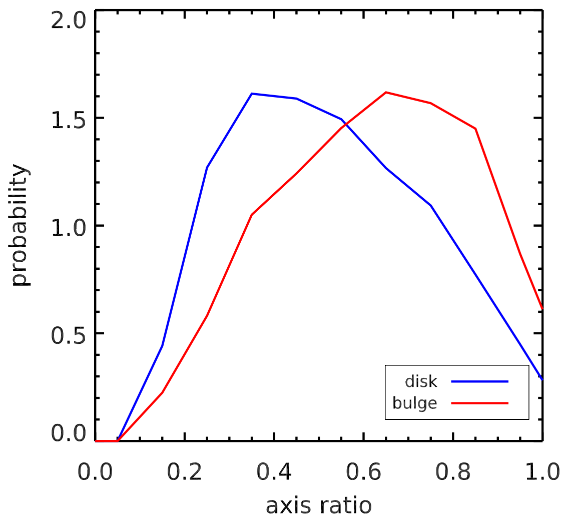


Figure 1: Observed axis ratio distribution of disk-dominated ($n < 1.5$) and bulge-dominated ($n > 2.5$) galaxies. Sérsic fits were taken from the CANDELS wiki, and were produced by Arjen van der Wel. Note that we also added a cut in stellar mass, in order not to be polluted by low mass faint galaxies ($M_* > 10^9\,\mathrm{M}_\odot$ for disks, $M_* > 3 \times 10^{10}\,\mathrm{M}_\odot$ for bulges).

2.3 Generating star formation rate

To generate star formation rates (SFRs), we used the Main Sequence approach, which attributes a "main sequence" SFR to every galaxy, knowing its redshift and its stellar mass. We used the calibration published in Schreiber et al. (2015), Eq. 9. On top of this, a random lognormal scatter of 0.3 dex is added, and a small fraction (3.3%) of the sample is randomly put in the "starburst" mode, following the 2SFM model (Sargent et al. 2012), and using the best-fit parameters obtained in Schreiber et al. (2015). In the end:

$$R_{\mathrm{SB}} = \begin{cases} 10^{G(0.3)} & \text{for Main Sequence galaxies} \\ 5.2 \times 10^{G(0.3)} & \text{for Staburst galaxies} \end{cases} \tag{5}$$

$$\mathrm{SFR} = \mathrm{SFR}_{\mathrm{MS}} \times R_{\mathrm{SB}}. \tag{6}$$

This quantity, R_{SB}, the "starburstiness", is used later to generate the IR photometry.

Then, we split this SFR between obscured and non-obscured components. The obscured component generates the IR fluxes, while the non-obscured component emerges naturally in the UV. To do so, we use the evolution of IRX $\equiv L_{\mathrm{IR}}/L_{\mathrm{UV}}$ observed in the *Herschel* stacks of Schreiber et al. (2015) (see also Heinis et al. 2014), which gives

$$\mathrm{IRX} = \frac{L_{\mathrm{IR}}}{L_{\mathrm{UV}}} = \begin{cases} 15.8 \times \left(\frac{M_*}{3 \times 10^{10}}\right)^{0.45\,z + 0.35} & \text{for } z < 3 \\ 15.8 \times \left(\frac{M_*}{3 \times 10^{10}}\right)^{1.7} & \text{for } z > 3. \end{cases} \tag{7}$$

From there is it then simple to recover L_{IR} and L_{UV}, and therefore the obscured and non-obscured part of the SFR. Passive galaxies are given zero SFR.

2.4 Generating optical colors

To generate UV to near-IR fluxes, we first need to choose an optical SED for each galaxy. To do so, we choose to start from the UVJ color-color diagram. In this diagram, passive galaxies occupy a well defined region (red cloud), while star-forming galaxies form a "sequence", which is actually generated by a combination of attenuation and age (see e.g. Williams et al. 2009, Fig. 8). This is useful, because it is known that both age and attenuation (e.g. Pannella et al. 2014) correlate strongly with the stellar mass. We used this fact to create a simple recipe to associate colors to active and passive galaxies, knowing only their redshift and masses.

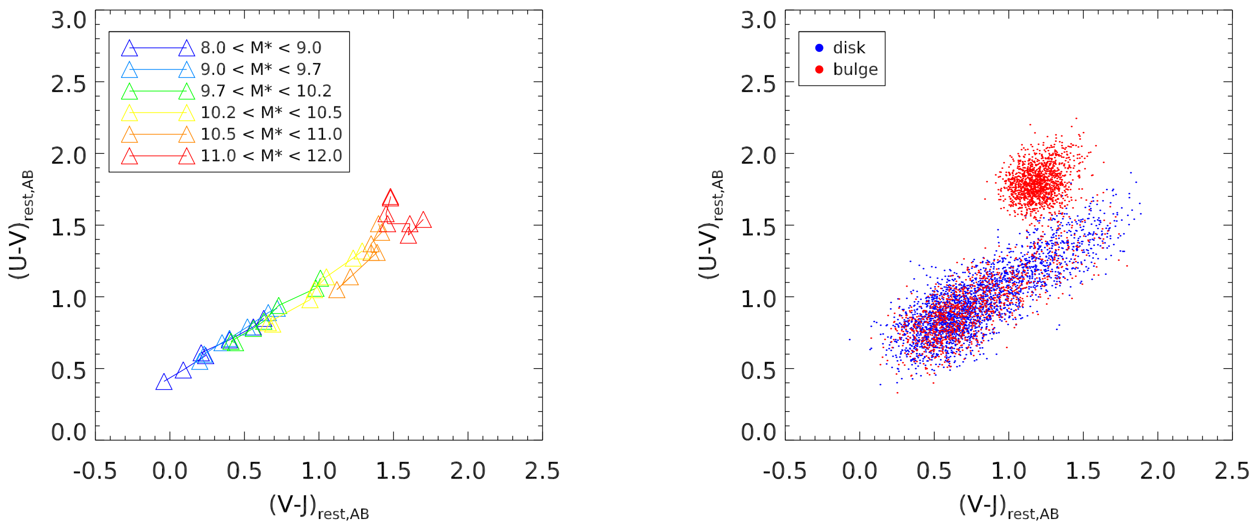


Figure 2: **Left:** Observed median colors of galaxies of different masses, for different redshift (from $z = 0.3$ to $z = 3.0$). The trend is that galaxies move diagonally toward the bottom-left corner when going to higher redshifts. **Right:** Generated UVJ colors of disk (blue) and bulge (red) components of galaxies with $M_* > 10^9\,\mathrm{M}_\odot$ and $0.8 < z < 1.2$.

We find that passive galaxies are well condensed in a fixed region, close to $V - J = 1.25$ and $U - V = 1.85$, with a very small trend with stellar mass. The principle is to put all passive galaxies at this position, shift them along the attenuation vector direction according to their stellar mass, and add some Gaussian noise to the generated colors. The final colors are chosen following

$$A = 0.1 \times (\log_{10}(M_*/\mathrm{M}_\odot) - 11) + G(0.1)\,, \tag{8}$$

$$(V - J)_{\rm passive} = 1.25 + A + G(0.1)\,, \tag{9}$$

$$(U - V)_{\rm passive} = 1.85 + 0.88 \times A + G(0.1)\,. \tag{10}$$

Note that the “shift” A is clamped to the range $[-0.1, 0.2]$ so that galaxies do not leave the red cloud.

For star-forming galaxies, one needs to be a bit more subtle because their colors vary a lot more. As can be seen, e.g., in Fig. 1 from Schreiber et al. (2015), star-forming galaxies populate different regions of the UVJ diagram depending on the stellar mass and redshift: massive galaxies are preferentially located on the top-right corner (red $U - V$ and $V - J$ colors), while low-mass galaxies are at the bottom-left (blue in $U - V$ and $V - J$), and they are shifted to bluer colors at higher redshift. We can parametrize this evolution.

To do so, we took a sample of UVJ star-forming galaxies in GOODS–South, and split them in mass bins. We further decompose each of these bins by slicing in redshift, and compute the median $U - V$ and $V - J$ colors. This produces a set of tracks in the UVJ diagram, which are reproduced in Fig. 2 (left). It turns out that these tracks fall roughly on a fixed line of slope 0.65, so reproducing these trend is relatively easy. We end up with the following formula

$$A_0 = 0.58 \times \mathrm{erf}(\log_{10}(M_*/\mathrm{M}_\odot) - 10) + 1.39\,, \tag{11}$$

$$A_s = \begin{cases} -0.34 + 0.3 \times \log_{10}\left(\frac{M_*}{2.2\times10^{10}\,\mathrm{M}_\odot}\right) & \text{for } M_* > 2.2\times10^{10}\,\mathrm{M}_\odot, \\ -0.34 & \text{for } M_* < 2.2\times10^{10}\,\mathrm{M}_\odot, \end{cases} \tag{12}$$

$$\tag{13}$$

$$A_1 = A_0 + A_s \times z\,, \tag{14}$$
$$A = A_1 + G(0.1)\,, \tag{15}$$
$$(V-J)_{\rm active} = 0.0 + A \times \cos(\theta) + G(0.12)\,, \tag{16}$$
$$(U-V)_{\rm active} = 0.45 + A \times \sin(\theta) + G(0.12)\,. \tag{17}$$

with A_1 being limited to at most 2, and $\theta = \arctan(0.65)$.

This parametrization will generate a UVJ diagram very similar to the observed one, with the same redshift and mass trends. However, the observed UVJ diagram is made out of the *total* light of the galaxy: here we need to decompose the galaxy into a bulge and a disk component, and both have usually different colors. The way we chose to handle this issue is to always use the "active" UVJ colors for disk components, always use the "passive" UVJ colors for bulges of bulge-dominated galaxies ($B/T > 0.6$), and randomly use either the "passive" or the "active" UVJ colors for the bulges of intermediate galaxies ($B/T < 0.6$) with 50% probability each. These prescriptions are lacking any direct observational constraints, and were therefore chosen somewhat arbitrarily so as to both reflect intuition and reproduce the observed color distribution.

The resulting UVJ colors are shown in Fig. 2 (right).

2.5 Choosing an optical SED

We then use these colors to associate a full optical SED to the galaxies. The idea is to consider that there is an average SED at each position on the UVJ diagram, and that one can attribute this average SED to the galaxies that are located at this position.

Therefore we have binned the UVJ plane into small buckets of about 0.1 mag, and computed the observed average rest-frame SED of all the galaxies that fall inside each bucket, assuming no redshift dependence. These rest-frame SEDs are actually generated by FAST with Bruzual & Charlot (2003) stellar population models, assuming a delayed exponentially declining star formation history. The result is a wide library of about 850 reference SEDs, all normalized per unit stellar mass.

Then the procedure is simply to pick one of these SEDs depending on the position of the galaxy in the UVJ diagram. We run this procedure for both disk and bulge components, multiply the chosen SEDs by the respective stellar mass of each component, redshift them to the redshift of the galaxy, and finally integrate the resulting SED over the chosen UV-NIR passbands to generate the corresponding fluxes.

2.6 Choosing an IR SED

The generation of the IR fluxes is the same as the one we used to generate the *Herschel* images with the previous Astrodeep mock catalog. Basically, we use the Chary & Elbaz (2001) library of FIR SEDs, normalize them to unit $L_{\rm IR}$, and attribute one of these SEDs to every galaxy, from its redshift and "starburtiness" (see Section 2.3). At higher redshifts, galaxies have warmer dust temperatures (Magdis et al. 2012), and the dust temperature also correlates with the offset of a galaxy from the Main Sequence (Magnelli et al. 2014). We use here the redshift evolution that was observed in the stacked *Herschel* SEDs of Schreiber et al. (2015).

Then, as for the optical flux computation, the chosen SED is multiplied by the $L_{\rm IR}$ of the galaxy, redshifted, and integrated over the chosen IR passbands to produce the final fluxes. For simplicity, we chose to attribute all of the FIR flux to the "disk" component. This should not matter, since at these wavelengths we usually do not have the resolution to disentangle between bulge and disk.

2.7 Generating sky positions

The final step is to generate a position on the sky for each galaxy. Here we make very simplistic assumptions. First, we assume the same angular correlation at all redshifts, which means that galaxies will be clustered on the same angular scale. This angular scale will correspond to a smaller proper distance at $z = 0.5$ than at $z = 1$, so it will somehow mimic the increase of proper distance clustering with time. Second, we consider that there is no sub-population of galaxies that is more clustered than the rest. E.g., massive early-type galaxies are treated the same way as dwarf star-forming galaxies. While this is probably wrong, it should be a sufficient approximation for now, and we can easily improve this later if need be. In fact, clustering is a relatively minor ingredient, and for our purposes it is only important that we generate catalogs with realistic sky-projected galaxy densities, with voids and peaks. The dependence on galaxy colors and properties is a second order effect.

We use these assumptions to measure the correlation function in the real GOODS–South catalog. We take into account that this correlation function is blurred by photometric redshift uncertainties, and use it to generate the position of each galaxies within a given redshift slices in the mock catalog using the Soneira & Peebles algorithm (power law index equal to 0.4, number of levels $N_{\rm level} = 4$). Doing so, one gets the right two-point correlation slope, but not the right amplitude: the correlation is too strong at all scale. To fix this, one has to say that there is a fraction (60%) of the sample which is not clustered, and we assign them uniformly random sky positions. This way, we reproduce the observed two-point correlation function over the whole field.

3 Results

We use two diagnostics to assess the quality of this mock catalog in each photometric band. The first one is the flux distribution of all galaxies, and the second is the pixel distribution of simulated images (only for confused FIR images where blending is important).

In what follows, we use a mock catalog generated with 90% completeness in H-band down to $H = 29$, from $z = 0.01$ to $z = 6$. Over 17×17 arcmin, this represents 104 000 galaxies. The minimum stellar mass goes as low as $5 \times 10^4\,{\rm M}_\odot$ at $z = 0.01$, and rises with redshift to reach $7 \times 10^6\,{\rm M}_\odot$ at $z = 1$, and $10^8\,{\rm M}_\odot$ at $z = 4$.

3.1 Optical magnitudes

Fig. 3 is showing the agreement of the total magnitude distribution, in multiple bands. This agreement is very good in the NIR. Since these wavelengths are most closely correlated to the stellar mass of the galaxies, and since the mock catalog was built to reproduce exactly the stellar mass function in GOODS–South, this should not come as a surprise. Still, this shows that the procedure works well. Generating the UV-optical (F435W and F606W) fluxes is more complex, because these bands actually trace the emerging UV light coming from star formation. Nevertheless, the agreement here is also very good.

We quantify the differences using the χ^2 statistics, and assuming only Poisson uncertainties (i.e., statistical fluctuations in the histograms, but not flux measurement uncertainties). For each of these bands, we measure reduced χ^2 of, respectively (from top-left to bottom-right), 4.23, 3.66, 2.22, 3.75 2.31 and 8.20 (for magnitudes brighter than 27, 27, 26, 26, 25 and 25, respectively). If our simulation was a perfect match to the data, and the observed differences were only due to statistical fluctuations, we would obtain $\chi^2 \sim 1$. The fact that we do not reach this value indicates that there are, of course, more subtle mechanisms in the real Universe than what we introduced here. In particular, the χ^2 of the IRAC channel 4 magnitudes is particularly high. We suspect this is due to the peculiar position of the observed $8\,\mu$m, which is probing dust emission at low-redshifts, and stellar emission at higher redshifts. The simulation here can be improved by introducing a better treatment of the junction point between these two wavelength regimes, and by choosing more carefully the IR SED. This is currently work in progress.

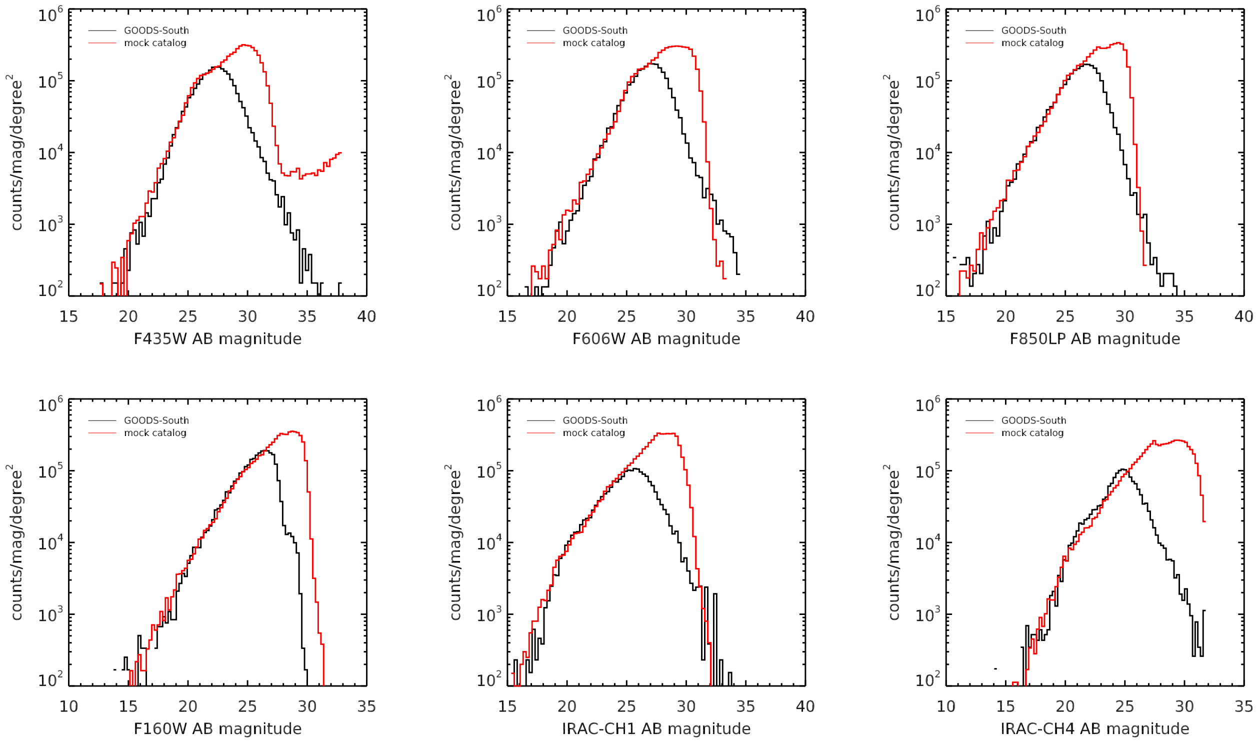


Figure 3: Total magnitude distribution of the real GOODS–South catalog (black) and the mock catalog (red), in different *HST* bands and *Spitzer* IRAC.

3.2 FIR fluxes

Fig. 4 shows the same plots, this time with the FIR fluxes. Again, the agreement is excellent. The χ^2 values are, respectively, 4.29, 2.25, 1.65, 1.10, 0.47, 1.50. Because the available observations are less extensive than for the optical magnitudes, these χ^2 are less stable, but still we do find values very close to 1. The worst case is that of the MIPS 24 μm, which is likely related to the IRAC 8 μm issue we reported in the previous section.

We also analyze in Fig. 5 the pixel histogram distribution of the simulated maps against the observed maps. This second test is important because of the blending, which sometimes pollutes the measured flux catalogs (two sources are combined into a single one), which tends to produce more bright fluxes than there actually is in the real Universe. By analyzing the map statistics directly, one gets rid of this issue of the counter part identification. This comparison also takes into account the clustering, which will tend to increase the contrast of the map without actually changing the fluxes of individual galaxies. The downside is that the bright pixel counts are very sensitive to statistical fluctuations, and a single very bright (but usually rare) object can drastically impact the measured distribution. Yet, here also the agreement is good. We find χ^2 = 3.03, 1.47, 4.98, 3.28, 2.71, and 1.09.

3.3 Generate images

Finally, we give an example is the simulated images we have produced in Fig. 6. This illustrates the power of our simulations, which are now physically consistent from the UV to the far-IR.

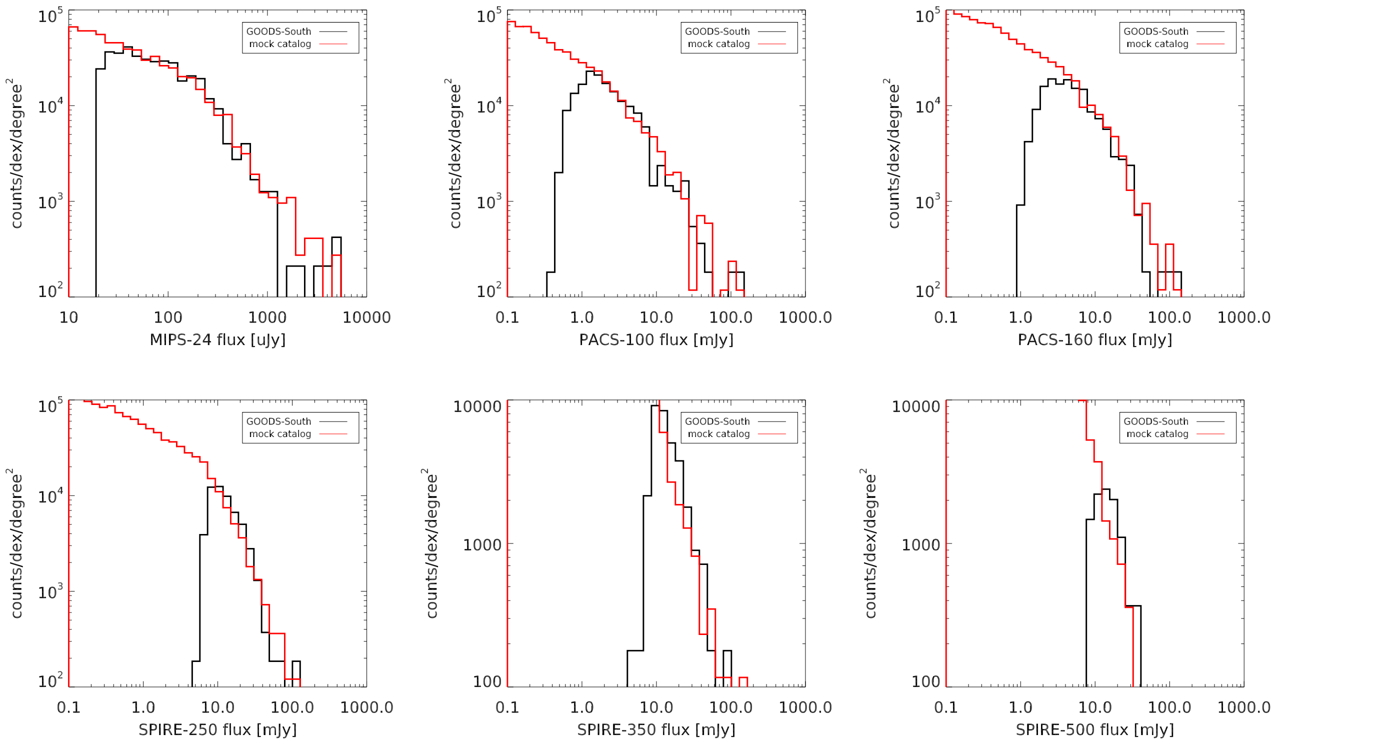


Figure 4: Total flux distribution in the MIR to FIR of the real GOODS–South catalog (black) and the mock catalog (red).

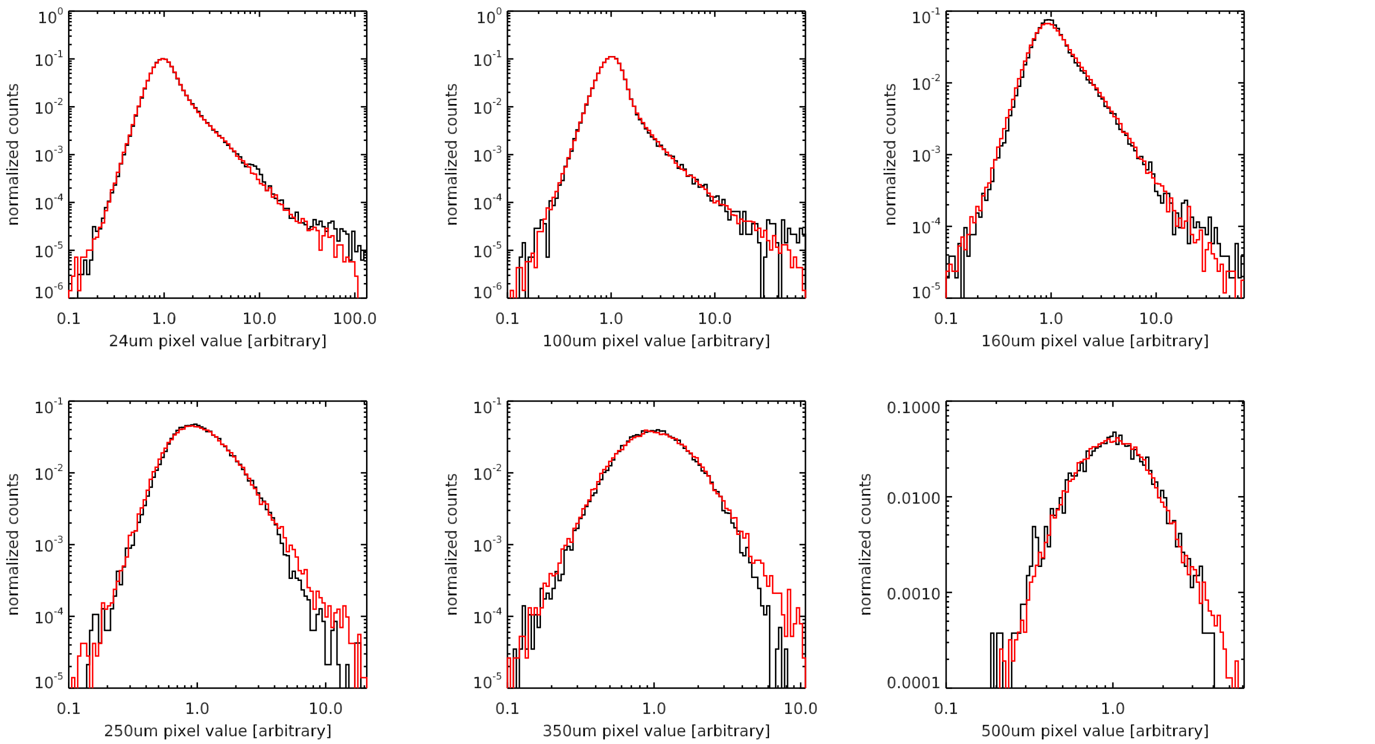


Figure 5: Pixel histogram distribution of the simulated FIR images versus real images in GOODS–South.

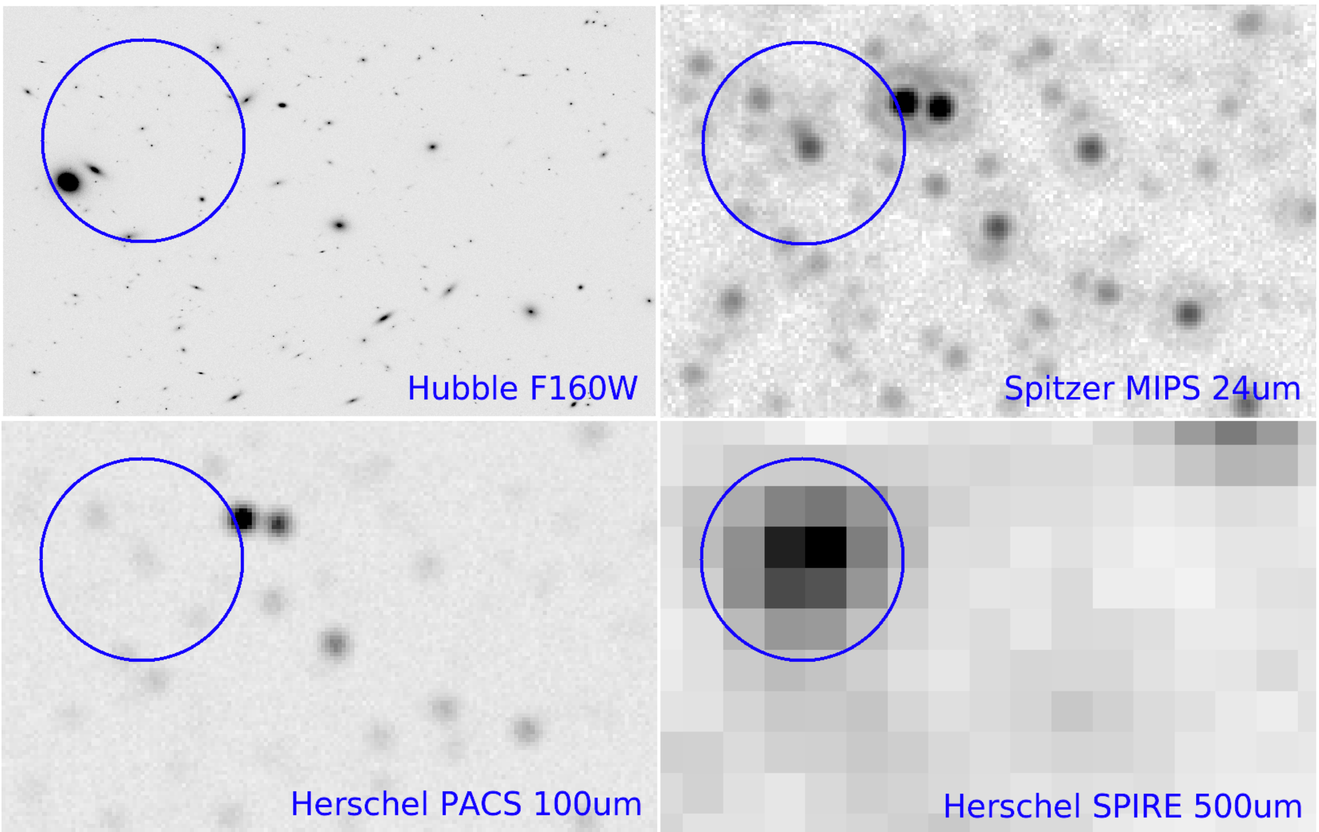


Figure 6: Simulated maps in the *Hubble* H band (top-left), *Spitzer* 24 μm (top-right), *Herschel* PACS 100 μm (bottom-left) and SPIRE 500 μm (bottom-right).

INSTALLING `gencat`

1 Forewords

`gencat` is written in C++ and has a few dependencies. I have tried to keep the number of these dependencies as low as possible, and in fact for the moment there are four:

- *phy*$_{++}$, a library for numerical analysis that I have developed during my PhD,
- cfitsio, for handling FITS files,
- WCSlib, for handling sky-to-pixel conversions,
- and CMake, for managing the building process (dependencies, and platform specific stuff).

2 Install dependencies

If your operating system comes with a package manager, this should be very easy. Apart from *phy*$_{++}$ that we will address in the next section, these dependencies are standard libraries and tools that should be available in all the package managers.

- Mac users:

  `sudo port install cfitsio wcslib cmake`

  or

  `sudo brew install cfitsio wcslib cmake`

- Linux/Ubuntu users:

  `sudo apt-get install libcfitsio3-dev wcslib-dev cmake`

- Other Linux distributions: You get the point. Use `yum`, `apt`, `pacman`, or whatever package manager is supported by your distribution.
- Windows users: Install Ubuntu and go to point 2.

If you don't have a package manager, then you have to compile these tools and libraries yourself... I hope it doesn't come to that, because you may loose a lot of time figuring this out. But in the eventuality, here are the links to places where you can download the source code. Follow the build instructions given on their respective each web page.

- cfitsio: `http://heasarc.gsfc.nasa.gov/fitsio/fitsio.html`
- WCSlib: `http://www.atnf.csiro.au/people/mcalabre/WCS/`
- CMake: `http://www.cmake.org/download/` (they also offer binaries, check this out first)

3 Install *phy*$_{++}$ and gencat

The rest is a little bit harder, but not that much. Thanks to CMake, the installing process is the same on all computers. In the following, I will assume that you have a directory somewhere on your computer where you keep all your programming related stuff (e.g., the source code of TPHOT if you have tried to compile it).

1. Download the following archives and extract them inside this directory:

   - `https://github.com/cschreib/phypp/archive/master.tar.gz`
   - `https://github.com/cschreib/gencat/archive/master.tar.gz`
   - `https://github.com/cschreib/filter-db/archive/master.tar.gz`

   This bash script will do that for you:

   ```
   wget https://github.com/cschreib/phypp/archive/master.tar.gz
   tar -xvzf master.tar.gz && rm master.tar.gz
   wget https://github.com/cschreib/gencat/archive/master.tar.gz
   tar -xvzf master.tar.gz && rm master.tar.gz
   wget https://github.com/cschreib/filter-db/archive/master.tar.gz
   tar -xvzf master.tar.gz && rm master.tar.gz
   ```

   In the end, this should create three directories:

   ```
   filter-db-master
   gencat-master
   phypp-master
   ```

2. Open a terminal and navigate to the `phypp-master` directory. Then, if you are using cfitsio and WCSlib from your package manager, run the following commands:

```
mkdir build && cd build
cmake ../
```

If instead you have installed one of these two libraries by hand, in a non-standard directory, you have to provide this directory to the CMake script. This is done by replacing the command :

```
mkdir build && cd build
cmake ../ -DWCSLIB_ROOT_DIR=... -DCFITSIO_ROOT_DIR=...
```

The "..." have to be replaced by the actual directory in which each library was installed. For example, if you have installed cfitsio in the `/opt/local/share/cfitsio` directory, then the "..." after `-DCFITSIO_ROOT_DIR` in the above command has to be replaced by "`opt/local/share/cfitsio`.

If all goes well, this will configure the *phy*$_{++}$ library and prepare it for installation. The script will most likely warn you about missing dependencies, but this is ok since none of these are needed for `gencat`. Just make sure that cfitsio and WCSlib are found correctly, then install the library with the following command:

```
sudo make install
source ~/.phypprc
```

3. Using the terminal, navigate now inside the `gencat-master` directory. Similarly, run the following commands:

```
mkdir build && cd build
cmake ../
```

This will generate an error if, somehow, there was an issue in the installation of the *phy*$_{++}$ library. Else, this will configure `gencat` and make it ready to be built. Finally, run the last command:

```
make install
```

The gencat binary will be created in `../bin`. See, not that hard!

4 Making sure everything works

Navigate into the `gencat-master/bin` directory and call:

```
./gencat verbose maglim=27 filter_db=../../filter-db-master/db.dat
```

This will take a few seconds to run. In the end, you should get something like:

```
note: initializing filters...
note: 15 optical bands and 8 IR bands
note: initializing redshift bins...
note: min dz: 0.05, max dz: 0.773951
note: 36 redshift slices
note: estimating redshift-dependend mass limit...
note: will generate masses from as low as 4.56944, up to 12
note: reading mass functions...
note: found 10 redshift bins and 181 mass bins
note: generating redshifts...
note: generated 50142 galaxies
note: generating masses...
note: generating morphology...
note: generating SFR...
note: assigning optical SEDs...
[--------------------------------------------------] 900 100%, 285ms elapsed, 0ns left, 285ms total
[--------------------------------------------------] 900 100%, 304ms elapsed, 0ns left, 304ms total
note: assigning IR SED...
note: computing fluxes...
note: computing optical fluxes...
[--------------------------------------------------] 50142 100%, 8s elapsed, 0ns left, 8s total
[--------------------------------------------------] 49883 100%, 8s elapsed, 0ns left, 8s total
note: computing IR fluxes...
[--------------------------------------------------] 50142 100%, 5s elapsed, 0ns left, 5s total
note: generating sky positions...
[--------------------------------------------------] 36 100%, 1s elapsed, 0ns left, 1s total
note: saving catalog...
```

Also, a file called `gencat-2015xxxx.fits` (e.g., for me it was `gencat-20150323.fits`) weighting about 20MB will be created in the same directory. This is the output catalog, in FITS format. You can open it in IDL to check its content with the following IDL command:

```
; Load the catalog
cat = mrdfits('gencat-2015xxxx.fits', 1)
; Look at its content
help, cat, /str
```

```
; Then do some plots
plot, cat.z, cat.m, psym=3, xtit='redshift', ytit='stellar mass'
```

There there remain to test the program that will tranlsate this catalog into a Skymaker-compatible catalog, one per band. Try:

```
./make_skymaker gencat-2015xxxx.fits band=f160w out=sky-f160w.cat
```

This should produce no output in the terminal, but create two files in the same directory, `sky-f160w.cat`, which is the Skymaker catalog, and `sky-f160w-hdr.txt`, which is the WCS header to feed to Skymaker.

Astronomy & Astrophysics manuscript no. tphot˙1˙working
May 12, 2015

T-PHOT*: a new code for PSF-matched, prior-based, multiwavelength extragalactic deconfusion photometry

E. Merlin[1], A. Fontana[1], H. C. Ferguson[2], J. S. Dunlop[3], D. Elbaz[4], N. Bourne[3], V. A. Bruce[3], F. Buitrago[3,10,11], M. Castellano[1], C. Schreiber[4], A. Grazian[1], R. J. McLure[3], K. Okumura[4], X. Shu[4,8], T. Wang[4,9], R. Amorín[1], N. Cappelluti[5], A. Comastri[5], S. Derriere[6], S. M. Faber[7], and P. Santini[1]

[1] INAF - Osservatorio Astronomico di Roma, Via Frascati 33, I - 00040 Monte Porzio Catone (RM), Italy e-mail: `emiliano.merlin@oa-roma.inaf.it`
[2] Space Telescope Science Institute, 3700 San Martin Drive, Baltimore, MD 21218, USA
[3] SUPA**, Institute for Astronomy, University of Edinburgh, Royal Observatory, Edinburgh, EH9 3HJ, U.K.
[4] Laboratoire AIM-Paris-Saclay, CEA/DSM/Irfu - CNRS - Université Paris Diderot, CEA-Saclay, pt courrier 131, F-91191 Gif-sur-Yvette, France
[5] INAF - Osservatorio Astronomico di Bologna, Via Ranzani 1, I - 40127, Bologna, Italy
[6] Observatoire astronomique de Strasbourg, Universit de Strasbourg, CNRS, UMR 7550, 11 rue de lUniversit, F-67000 Strasbourg, France
[7] UCO/Lick Observatory, University of California, 1156 High Street, Santa Cruz, CA 95064, USA
[8] Department of Physics, Anhui Normal University, Wuhu, Anhui, 241000, China
[9] School of Astronomy and Astrophysics, Nanjing University, Nanjing, 210093, China
[10] Instituto de Astrofísica e Ciências do Espaço, Universidade de Lisboa, OAL, Tapada da Ajuda, PT1349-018 Lisbon, Portugal
[11] Departamento de Fìsica, Faculdade de Ciências, Universidade de Lisboa, Edificio C8, Campo Grande, PT1749-016 Lisbon, Portugal

Preprint online version: May 12, 2015

Abstract

Context. The advent of deep multiwavelength extragalactic surveys has led to the necessity for advanced and fast methods for photometric analysis. In fact, codes which allow analyses of the same regions of sky observed at different wavelengths and resolutions are becoming essential to thoroughly exploit current and future data. In this context, a key issue is the *confusion* (i.e. blending) of sources in low resolution images.
Aims. We present T-PHOT, a publicly available software package developed within the ASTRODEEP project. T-PHOT is aimed at extracting accurate photometry from low resolution images, where the blending of sources can be a serious problem for the accurate and unbiased measurement of fluxes and colours.
Methods. T-PHOT can be considered as the next generation to TFIT, providing significant improvements over and above it and other similar codes (e.g. CONVPHOT). T-PHOT gathers data from a high resolution image of a region of the sky, and uses this information (source positions and morphologies) to obtain priors for the photometric analysis of the lower resolution image of the same field. T-PHOT can handle different types of datasets as input priors: namely, i) a list of objects that will be used to obtain cutouts from the real high resolution image; ii) a set of analytical models (as `.fits` stamps); iii) a list of unresolved, point-like sources, useful e.g. for far infrared wavelength domains.
Results. By means of simulations and analysis of real datasets, we show that T-PHOT yields accurate estimations of fluxes within the intrinsic uncertainties of the method, when systematic errors are taken into account (which can be done thanks to a flagging code given in the output). T-PHOT is many times faster than similar codes like TFIT and CONVPHOT (up to hundreds, depending on the problem and the method adopted), whilst at the same time being more robust and more versatile. This makes it an optimal choice for the analysis of large datasets. When used with the same parameter sets as for TFIT it yields almost identical results (albeit in a much shorter time), but in addition we show how the use of different settings and methods significantly enhances the performance.
Conclusions. T-PHOT proves to be a state-of-the-art tool for multiwavelength optical to FIR image photometry. Given its versatility and robustness, T-PHOT can be considered the preferred choice for combined photometric analysis of current and forthcoming extragalactic imaging surveys.

Key words. Galaxy, photometry, multiwavelength, software

1. Introduction

Combining observational data from the same regions of the sky in different wavelength domains has become common practice in the past few years (e.g. Agüeros et al. 2005; Obrić et al. 2006; Grogin et al. 2011, and many oth-

* T-PHOT is publicly available for downloading from *www.astrodeep.eu/t-phot/* .
** Scottish Universities Physics Alliance

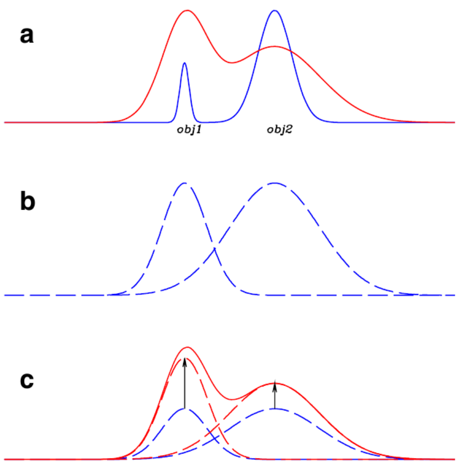


Figure 1. A schematic representation of the PSF-matched algorithm implemented in T-PHOT. Top: two objects are clearly detected and separated in the high resolution detection image (blue line). The same two objects are blended in the low resolution measurement image and have different colors (red line). Middle: the two objects are isolated in the detection image and are individually smoothed to the PSF of the measurement image, to obtain normalized model templates. Bottom: the intensity of each object is scaled to match the global profile of the measurement image. The scaling factors are found with a global χ^2 minimisation over the object areas. Image from De Santis et al. (2007).

ers). However, the use of both space-based and ground-based imaging instruments, with different sensitivities, pixel scales, angular resolutions, and survey depths, raises a number of challenging difficulties in the data analysis process.

In this context, it is of particular interest to obtain detailed photometric measurements for high redshift galaxies in the near infrared (NIR; corresponding to rest-frame optical) and far infrared (FIR) domains. In particular, great attention must be paid to bandpasses containing spectral features which allows for thorough physical investigation of the sources, to disentangle degenerate observational features and to obtain crucial clues to the understanding of the galactic physics (e.g. Daddi et al. 2004; Fontana et al. 2009). For example, at $z > 3$ photometry longward of H-band is needed to locate and measure the size of the Balmer break. A passive galaxy at $z \simeq 6$ (having the Balmer break lying longward of the K-band) can have H band and 3.6μm fluxes compatible with e.g. a star forming, dusty galaxy at $z \simeq 1$, and $K-$band photometry is necessary to disentangle the degeneracy. However, the limited resolution of the ground based $K-$band observations can impose severe limits on the reliability of traditional aperture or even PSF-fitting photometry. Also, IRAC photometry is of crucial importance to obtain reliable photometric redshifts for red and high-z sources, and to derive robust stellar mass estimates.

To address this, a high resolution image (HRI), obtained e.g. from the Hubble Space Telescope in the optical domain, can be used to retrieve detailed information on the positions and morphologies of the sources in a given region of the sky. Such information can be subsequently used to perform the photometric analysis of the lower resolution image (LRI), using the HRI data as priors. However, simply performing aperture photometry on the LRI at the positions measured in the HRI can be dramatically affected by neighbour contamination for reasonably sized apertures. On the other hand, performing source extraction on both images and matching the resulting catalogs is compromised by the inability to deblend neighbouring objects, and may introduce significant inaccuracies in the cross-correlation process. PSF-matching techniques that degrade high-resolution data to match the low resolution data discard much of the valuable information obtained in the HRI, reducing all images to the "lowest common denominator" of angular resolution. Moreover, crowded-field, PSF-fitting photometry packages such as DAOPHOT (Stetson 1987) perform well if the sources in the LRI are unresolved, but are unsuitable for analysis of even marginally resolved images of extragalactic sources.

A more viable approach consists of taking advantage of the morphological information given by the HRI, to obtain high resolution cutouts or models of the sources. These priors can then be degraded to the resolution of the LRI using a suitable *convolution kernel*, constructed by matching the PSFs of the HRI and of the LRI. Such low resolution templates, normalized to unit flux, can then be placed at the positions given by the HRI detections, and the multiplicative factor that must be assigned to each model to match the measured flux in each pixel of the LRI will give the measured flux of that source. Such an approach, although relying on some demanding assumptions as described in the following Sections, has proven to be efficient. It has been implemented in such public codes as TFIT (Laidler et al. 2007) and CONVPHOT (De Santis et al. 2007), and has already been utilized successfully in previous studies (e.g. Guo et al. 2013; Galametz et al. 2013).

In this paper we describe a new software package, T-PHOT, developed at INAF-OAR as part of the ASTRODEEP project[1]. T-PHOT can be considered as a new, largely improved version of TFIT, supplemented with many of the features of CONVPHOT. Moreover, it adds many important

[1] ASTRODEEP is a co-ordinated and comprehensive program of i) algorithm/software development and testing; ii) data reduction/release, and iii) scientific data validation/analysis of the deepest multi-wavelength cosmic surveys. To get more information, visit *http://astrodeep.eu* .

new options, including the possibility of adopting different types of priors (namely, real images, analytical models, or point-sources). In particular, it is possible to use T-PHOT on FIR and sub-millimetric (sub-mm) datasets, as a competitive alternative to the existing dedicated software such as FastPhot (Béthermin et al. 2010) and DesPhot (Roseboom et al. 2010; Wang et al. 2014). This makes T-PHOT a versatile tool, suitable for the photometric analysis of a very broad range of wavelengths from UV to sub-mm.

T-PHOT is a robust and easy-to-handle code, with a precise structural architecture (a Python envelope calling `C/C++` core codes) in which different routines are encapsulated, implementing various numerical/conceptual methods, to be chosen by simple switches in a parameter file. While a standard default "best choice" usage mode is provided and suggested, the user is allowed to select their own preferred way of obtaining their dataset.

One of the main advantages of T-PHOT is a significant saving of computational time with respect to both TFIT and CONVPHOT (see Sect. 4). This has been achieved with the use of fast `C` modules and an efficient structural arrangement of the code. In addition to this, we demonstrate how different choices of parameters influence the performace, and can be optimized to significantly improve the final results with respect to e.g. TFIT.

The plan of the paper is as follows. Sect. 2 provides a general introduction to the code, its mode of operation and its algorithms. Sect. 3 presents a comprehensive set of tests, based on both simulated and real datasets, to assess the performance of the code and to fully illustrate its capabilities and limitations. Sect. 4 briefly discusses the computational performances of T-PHOT and provides some reference computational timescales. Finally, in Sect. 5 the key features of T-PHOT are summarized, and outstanding issues and potential complications are briefly discussed.

2. General description of the code

As described above, T-PHOT uses spatial and morphological information gathered from a HRI to measure the fluxes in a LRI. To this aim, a linear system is built and solved via matricial computing, minimizing the χ^2 (in which the numerically determined fluxes for each detected source are compared to the measured fluxes in the LRI, summing the contributions of all pixels). Moreover, the code produces a number of diagnostic outputs and allows for an iterative re-calibration of the results. Fig. 1 shows a schematic depiction of the basic PSF-matched fitting algorithm used in the code.

As HRI priors T-PHOT can use i) real cutouts of sources from the HRI, ii) models of sources obtained e.g. with Galfit or similar codes, iii) a list of coordinates where PSF-shaped sources will be placed; or a combination of these three types of priors.

For a detailed technical description of the mode of operation of the code, we refer the reader to the Appendix and to the documentation included in the downloadable tarball. Here, we will briefly describe its main features.

2.1. Pipeline

The pipeline followed by T-PHOT is outlined in the flowchart given in Fig. 2. The following paragraphs give a short description of the pipeline.

2.1.1. Input

The input files needed by T-PHOT vary depending on the type(s) of priors used.

If *"true" high-resolution priors* are used, e.g. for optical/NIR ground-based or IRAC measurements using HST cutouts, T-PHOT needs:

- the detection, high resolution image (HRI) in `.fits` format;
- the catalog of the sources in the HRI, obtained e.g. using SExtractor or similar codes (the required format is described in Appendix A);
- the segmentation map of the HRI, in `.fits` format, again obtained e.g. using SExtractor or similar codes, having the value of the `id` of each source in the pixels belonging to it, and zero everywhere else;
- a convolution kernel K, in the format of a `.fits` image or of a `.txt` file, matching the PSFs of the HRI and the LRI so that $PSF_{LRI} = K * PSF_{HRI}$ ($*$ is the symbol for convolution). The kernel *must* have the HRI pixel scale.

If *analytical models priors* are used as priors (e.g. Galfit models), T-PHOT needs:

- the stamps of the models (one per object, in `.fits` format);
- the catalog of the models (the required format is described in Appendix A);
- the convolution kernel K matching the PSFs of the HRI and the LRI, as in the previous case.

If models have more than one component, one separate stamp per component, and catalogs for each component are needed (e.g. one catalog for bulges and one catalog for disks).

If *unresolved, point-like priors* are used, T-PHOT needs:

- the catalog of positions (the required format is described in Appendix A);
- the LRI PSF, in the LRI pixel scale.

In this case, a potential limitation to the reliability of the method is given by the fact that the prior density usually needs to be optimised with respect to FIR/sub-mm maps, as discussed e.g. in Shu et al. (2015, in preparation) and Elbaz et al. (2011) (see also Wang et al. 2015; Bourne et al. 2015, in preparation). The optimal number of priors turns out to be around 50-75% of the numbers of beams in the map. The key problem is to identify which of the many potential priors from e.g. an HST catalogue one should use. This is a very complex issue and we do not discuss it in this paper.

If mixed priors are used, T-PHOT obviously needs the input files corresponding to each of the different types of priors in use.

Finally, in all cases T-PHOT needs

- the measure LRI, *background subtracted* (see next paragraph), in `.fits` format, with the same orientation as the HRI (i.e., no rotation allowed); the pixel scale can be equal to, or an integer multiple of, the HRI pixel scale, and the origin of one pixel must coincide; it should be in surface brightness units (e.g. counts/s/pixel, or Jy/pixel for FIR images, and not PSF-filtered);

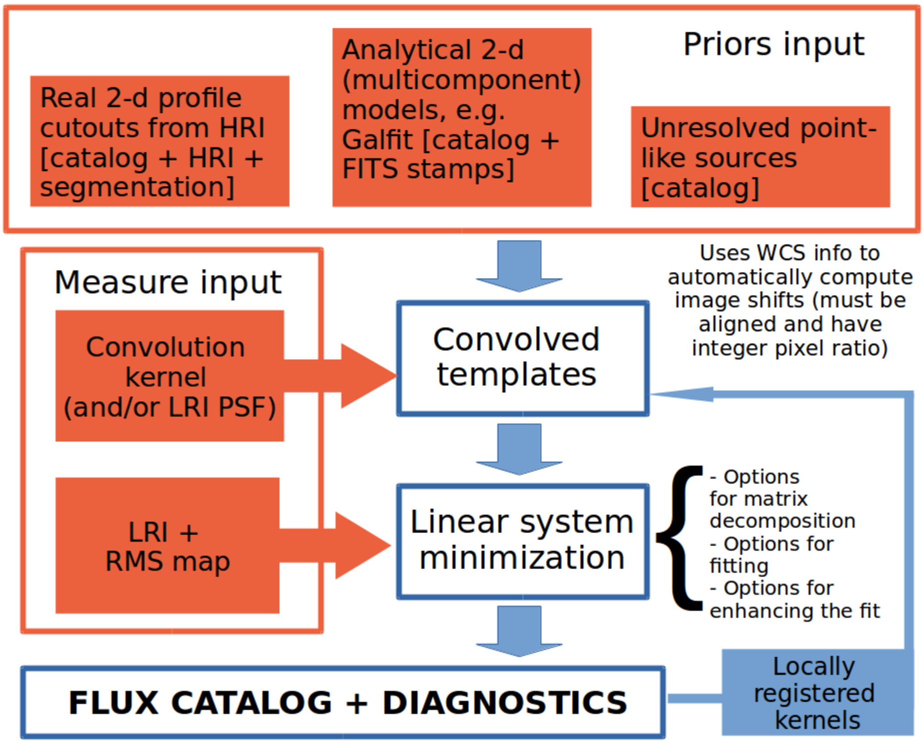


Figure 2. Schematic representation of the workflow in T-PHOT.

	Real cutouts	Analytical models	Point-sources
Priors	HRI Segmentation Catalog	HRI Model Stamps Catalog	Positions Catalog
Transformation	Convolution Kernel	Convolution Kernel	PSF_{LRI}
Measure	LRI RMS_{LRI}	LRI RMS_{LRI}	LRI RMS_{LRI}

Table 1. The input files needed by T-PHOT for different settings. See text for details.

– the LRI RMS map, in `.fits` format, with the same dimensions and WCS of the LRI.

Table 1 summarizes the input requirements for the different choices of priors just described.

All the input images *must* have the following keywords in their headers: `CRPIXn`, `CRVALn`, `CDn_n`, `CTYPEn` (n=1,2).

2.1.2. On the background subtraction

As already mentioned, the LRI *must* be background subtracted before being fed to T-PHOT. This is of particular interest when dealing with FIR/sub-mm images, where the typical standard is to use zero-mean. To estimate the background level in optical/NIR images, one simple possibility is to take advantage of the option to fit point-like sources to measure the flux for a list of positions chosen to fall within void regions. The issue is more problematic in such confusion-limited FIR images where there are no empty sky regions. In such cases, it is important to separate the fitted sources (those listed in the prior catalogue) from the background sources, which contribute to a flat background level behind the sources of interest. The priors should be chosen so that these two populations are uncorrelated. The average contribution of the faint background source population can then be estimated e.g. by (i) injecting fake sources into the map and measuring the average offset (output-input) flux; or (ii) measuring the modal value in the residual image after a first pass through T-PHOT (see e.g. Bourne et al. 2015, in preparation).

2.1.3. Stages

T-PHOT goes through "stages", each of which performs a well defined task. The best results are obtained performing two runs ("pass 1" and "pass 2"), the second one using locally registered kernels, produced during the first one. The possible stages are the following:

– `priors`: creates/organizes stamps for sources as listed in the input priors catalog(s);
– `convolve`: convolves each high resolution stamp with the convolution kernel K to obtain models ("templates") of the sources at LRI resolution. *The templates are normalized to unit total flux.* If the pixel scale of the

images is different, transforms templates accordingly. Convolution is preferably performed in Fourier space, using fast `FFTW3` libraries; however the user can choose to perform it in real pixel space, ensuring a more accurate result at the expense of a much slower computation.
- `positions`: if an input catalog of unresolved sources is given, creates the PSF-shaped templates listed in it, and merges it with the one produced in the `convolve` stage;
- `fit`: performs the fitting procedure, solving the linear system and obtaining the multiplicative factors to match each template flux with the measured one;
- `diags`: selects the best fits[2] and produces the final formatted output catalogs with fluxes and errors, plus some other diagnostics, see Sect. 2.3;
- `dance`: obtains local convolution kernels for the second pass; it can be skipped if the user is only interested in a single pass run;
- `plotdance`: plot some diagnostics for the dance stage; it can be skipped for any other purpose than diagnostics;
- `archive`: archives all results in a subdirectory whose name is based on the LRI and the chosen fitting method (to be only used at the end of the second pass).

The exact pipeline followed by the code is specified by a keyword in the input parameter file. See the Appendix for a more detailed description of the whole procedure.

2.1.4. Solution of the linear system

The search for the LRI fluxes of the objects detected in the HRI is performed by creating a linear system

$$\sum_{m,n} I(m,n) = \sum_{m,n}\sum_{i}^{N} F_i P_i(m,n) \tag{1}$$

where m and n are the pixel indexes, I contains the pixel values of the fluxes in the LRI, P_i is the normalized flux of the template for the i-th objects in the (region of the) LRI being fitted, and F_i is the multiplicative scaling factor for each object. In physical terms, F_i represent the flux of each object in the LRI (that is, it is the unknown to be determined).

Once the normalized templates for each object in the (region of interest within the) LRI have been generated during the `convolve` stage, the best fit to their fluxes can be simultaneously derived by minimizing a χ^2 statistic,

$$\chi^2 = \left[\frac{\sum_{m,n} I(m,n) - M(m,n)}{\sigma(m,n)}\right]^2 \tag{2}$$

where m and n are the pixel indexes,

$$M(m,n) = \sum_{i}^{N} F_i P_i(m,n) \tag{3}$$

and σ is the RMS value in the pixel.

The output quantities are the best-fit solutions of the minimization procedure, i.e. the F_i parameters and their relative errors. They can be obtained resolving the linear system

$$\frac{\partial \chi^2}{\partial F_i} = 0 \tag{4}$$

for $i = 0, 1, ..., N$.

In practice, the linear system can be rearranged into a matrix equation,

$$AF = B \tag{5}$$

where the matrix A contains the coefficients $P_i P_j/\sigma^2$, F is a vector containing the fluxes to be determined, and B is a vector given by $I_i P_i/\sigma^2$ terms. The matrix equation is solved via one of three possible methods as described in the next subsection.

2.1.5. Fitting options

T-PHOT allows for some different options to perform the fit:

- three different methods for solving the linear system are implemented: namely, the LU method (used by default in TFIT); the Cholesky method; and the Iterative Biconjugate Gradient method (used by default in CONVPHOT). They prove to yield similar results, the LU method being slightly more stable and faster;
- a threshold can be imposed so that only pixels with a flux higher than it will be used in the fitting procedure (see Sect. 3.1.4);
- sources fitted with a large, unphysical negative flux ($f_{meas} < -3\sigma$, where σ is their nominal error, see below) can be excluded from the fit, and in this case a new fitting loop will be performed without considering these sources.

The fit can be performed (i) on the entire LRI as a whole, producing a single matrix containing all the sources (this is the method adopted in CONVPHOT); (ii) subdividing the LRI into an arbitrary grid of (overlapping) small cells, perfoming the fit in each of such cells separately, and then choosing the "best" fit for each source, using some convenient criteria to select it (because sources will be fitted more than once, if the cells overlap. This is the method adopted in TFIT); (iii) ordering objects by decreasing flux, building a cell around each source including all its potential contaminants, solving the problem in that cell and assigning to the source the obtained flux (*cells-on-objects* method; see the Appendix for more details).

While the first method is the safest and more accurate because it does not introduce any bias or arbitrary modifications, it may often be unfeasible to process at once large or very crowded images. Potentially large computational time saving is possible using the *cells-on-objects* method, depending on the level of blending/confusion in the LRI: if the latter is very high, most sources will be overlapping, so the cells will end up being very large. This ultimately results in repeating many times the fit on regions with dimensions comparable to the whole image (a check is implemented in the code, to automatically change the method from *cells-on-objects* to single fit if this is the case). If the confusion is not dramatic, a saving in computational time up to two orders of magnitude can be achieved. The results obtained using the *cells-on-objects* method prove to be virtually identical to those obtained with a single fit on the whole image (see Sect. 3.1.2). On the other hand, using the arbitrary cells method is normally the fastest option, but can introduce potentially large errors to the flux estimates, due to wrong assignments of peripheral flux from sources located outside a given cell to sources within the cell (again, see Sect. 3.1.2 and Appendix).

[2] Each source is fitted more than once if an arbitrary grid is used, as in the standard TFIT approach.

2.1.6. Post-fitting stages: kernel registration

After the fitting procedure is completed, T-PHOT will produce the final output catalogs and diagnostic images (see 2.3). Among these, a *model* image is obtained by adding all the templates, scaled to their correct total flux after fitting, in the positions of the sources. This image will subsequently be used if a second pass is planned, during a stage named `dance`: a list of positional shifts is computed, and a set of shifted kernels are generated and stored. The `dance` stage consists of three conceptual steps:

- the LRI is divided into cells of given size (specified by the keyword `dzonesize`) and a linear $\Delta x, \Delta y$ shift is computed within each cell, cross-correlating the model image and the LRI in the considered region[3];
- interpolated shifts are computed for the regions where the previous registration process gives spuriously large shifts, i.e. above the given input threshold parameter `maxshift`;
- the new set of kernels is created using the computed shifts to linearly interpolate their positions, while catalogs reporting the shifts and the paths to kernels are produced.

2.1.7. Second pass

The registered kernels can subsequently be used in the second pass run, to obtain more astrometrically precise results. T-PHOT automatically deals with them provided the correct keyword is given in the parameter file. If unresolved priors are used, the list of shifts generated in the `dance` stage will be used by the `positions` routine during the second pass to produce correctly shifted PSFs and generate new templates.

2.2. Error budget

During the fitting stage, the covariance matrix is constructed and output. Errors for each source are assigned as the square root of the diagonal element of the covariance matrix relative to that source. It must be pointed out that using any cell method for the fitting, rather the single fitting option, will affect this uncertainty budget, since a different matrix will be constructed and resolved in each cell.

It is important to stress that this covariance error budget is a *statistical* uncertainty, relative to the RMS fluctuations in the measurement image, and is not related to any possible *systematic* error. The latter can instead be estimated by flagging potentially problematic sources, to be identified separately from the fitting procedure. There can be different possible causes for systematic offsets of the measured flux with respect to the true flux of a source. T-PHOT assigns the following flags:

- +1 if the prior has saturated or negative flux;
- +2 if the prior is blended (the check is performed on the segmentation map);
- +4 if the source is at the border of the image (i.e. its segmentation reaches the limits of the HRI pixels range).

[3] FFT and direct cross-correlations are implemented, the latter being the preferred default choice because it gives more precise results at the expense of a slightly slower computation.

2.3. Description of the output

T-PHOT output files are designed to be very similar in format to those produced by TFIT. They provide:

- a "best" catalog containing the following data, listed for each detected source (as reported in the catalog file header):
  - `id`;
  - `x` and `y` positions (in LRI pixel scale and reference frame, FITS convention where first pixel is centered at 1,1);
  - `id` of the cell in which the best fit has been obtained (only relevant for arbitrary grid fitting method);
  - x and y positions of the object in the cell and distance from the center (always equal to 0 if the *cells-on-objects* method is adopted);
  - fitted flux and its uncertainty (square root of the variance, from the covariance matrix). These are the most important output quantities;
  - flux of the object as given in the input HRI catalog or, in the case of point-sources priors, measured flux of the pixel at the x, y position of the source in the LRI;
  - flux of the object as determined in the `cutout` stage (it can be different to the previous one, e.g. if the segmentation was dilated); in case of point-sources priors, measured flux of the pixel at the x, y position of the source in the LRI;
  - *flag* indicating a possible bad source as described in the previous subsection;
  - number of fits for the object (only relevant for arbitrary grid fitting method, 1 in all other cases).
  - `id` of the object having the largest covariance with the present source;
  - *covariance index*, i.e. the ratio of the maximum covariance to the variance of the object itself; this number can be considered an indicator of the reliability of the fit, since large covariances often indicate a possible systematic offset in the measured flux of the covarying objects (see Sect. 3.1.2).
- two catalogs reporting statistics for the fitting cells and the covariance matrices (they are described in the documentation);
- the model `.fits` image, obtained as a collage of the templates, as already described;
- a diagnostic residual `.fits` image, obtained by subtracting the model image from the LRI;
- a subdirectory containing all the low resolution model templates;
- a subdirectory containing the covariance matrices in graphic (`.fits`) format;
- a few ancillary files relating to the shifts of the kernel for the second pass and a subdirectory containing the shifted kernels.

All fluxes and errors are output in units consistent with the input images.

Figs. 3, 4 and 5 show three examples of T-PHOT applications on simulated and real data, using the three different options for priors.

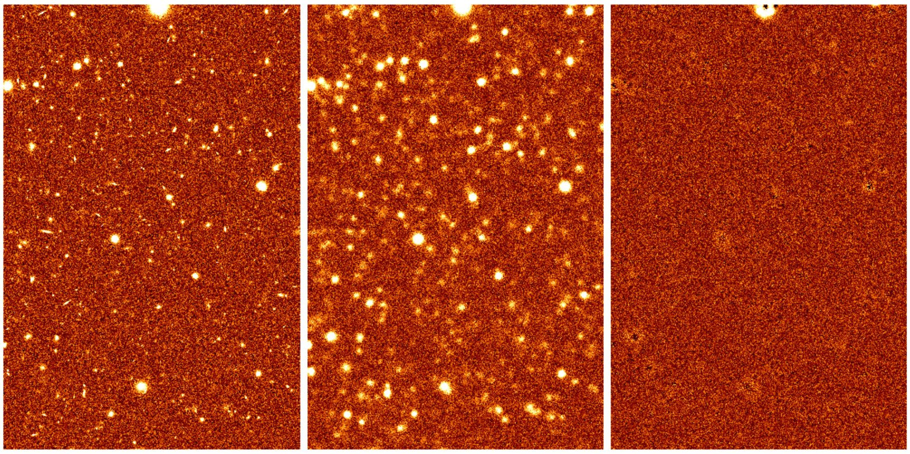


Figure 3. Example of the results of a standard T-PHOT run using extended priors. Left to right: HRI (FWHM=0.2″), LRI (FWHM=1.66″), and residuals image for a simulated dataset. LRI and residual image are on the same grayscale.

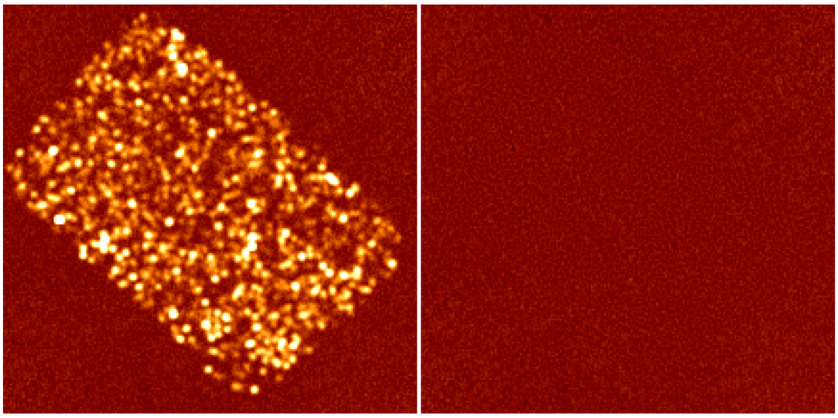


Figure 4. Example of the results of a standard T-PHOT run using point-source priors. Left to right: LRI (FWHM=25") and residuals image (same grayscale) for a simulated dataset.

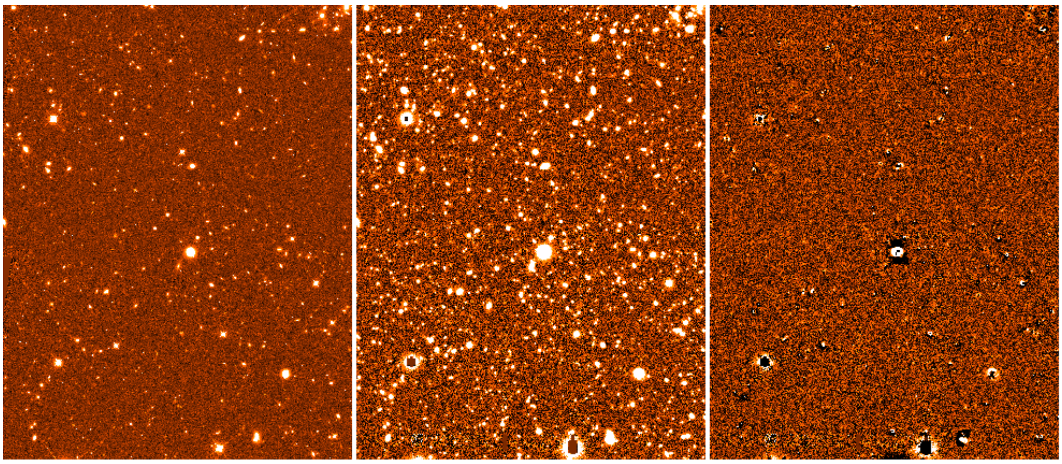


Figure 5. Example of the results of a standard T-PHOT run using analytical priors. Left to right: CANDELS COSMOS *H*-band (HRI), *R*-band (LRI) and residuals image obtained using GALFIT two-component models. LRI and residual image are on the same grayscale.

3. Validation

To assess the performance of T-PHOT we set up an extensive set of simulations, aimed at various different and complementary goals.

We used SKYMAKER (Bertin 2009), a public software tool, to build synthetic `.fits` images. The code ensures direct control on all the observational parameters (the magnitude and positions of the objects, their morphology, the zero point magnitude, the noise level, and the PSF). Model galaxies are built by summing a de Vaucouleurs and an

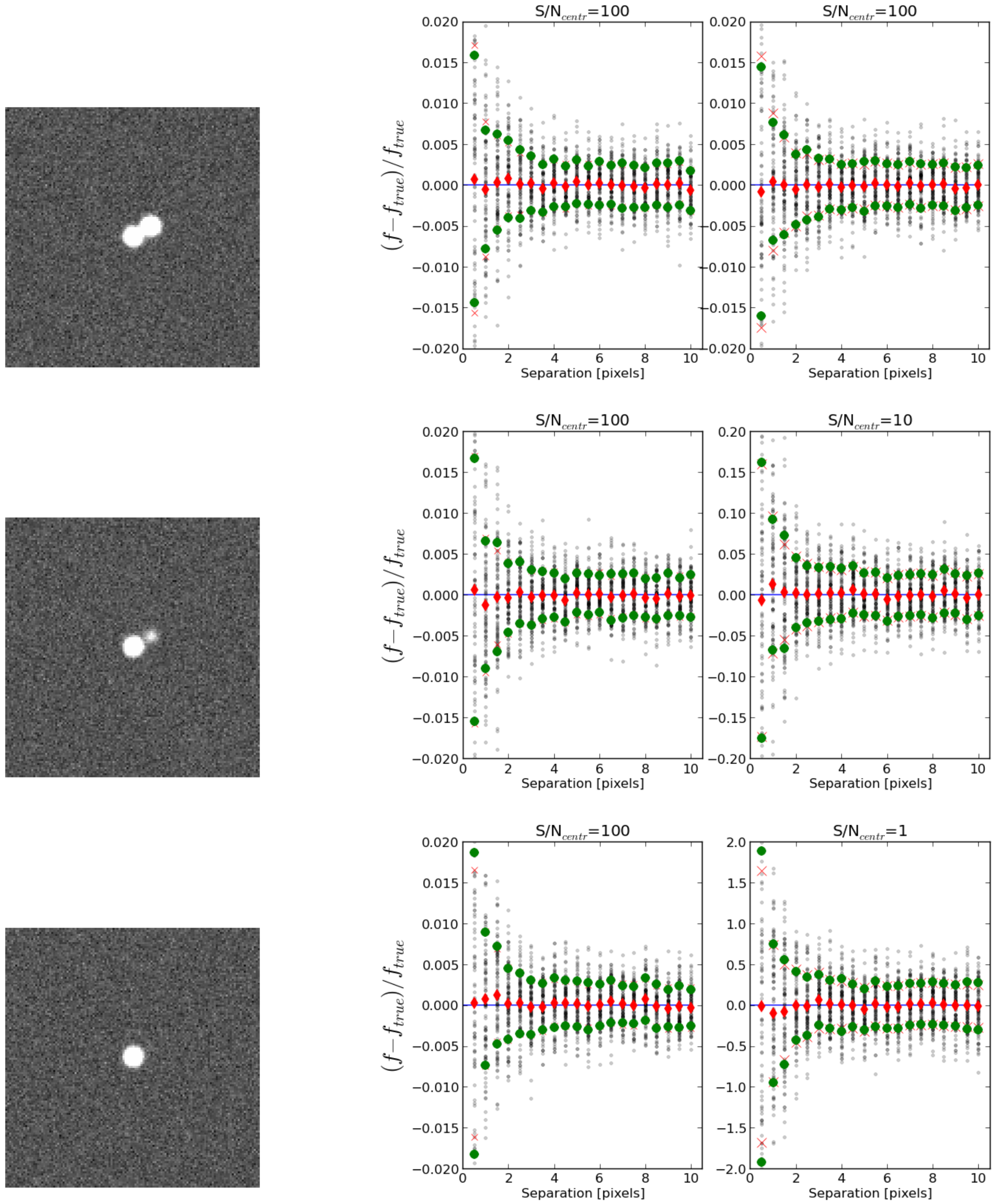


Figure 6. Accuracy check on idealized PSF-shaped objects. 100 realizations of the same image containing two PSF-shaped objects at varying positions and signal-to-noise ratios have been produced and the fluxes have been measured with T-PHOT. In each row, the left image shows one of the 100 realizations with the largest considered separation (10 pixels). On the right, the first panel refers to the central object, and the second (on the right) to the shifted object; central signal-to-noise (S/N_{centr}) ratios are, from top to bottom, 100-100, 100-10, and 100-1 for the two objects respectively. In each panel, as a function of the separation interval between the two sources the faint grey points show each of the 100 flux measurements (in relative difference with respect to the "true" input flux), the red diamonds are the averages of such 100 measurements, the red crosses show the nominal error given by the covariance matrix in T-PHOT and the green dots the standard deviation of the 100 measurements. See text for more details.

exponential light profile in order to best mimick a realistic distribution of galaxy morphologies. These models are generated using a variety of bulge-to-total light ratios, component sizes and projection angles.

All tests have been run using ideal (i.e. synthetic and symmetric) PSFs and kernels.

Moreover, we also perform tests on real datasets taken from the CANDELS survey (in these cases using real PSFs).

Some of the tests were performed using both T-PHOT and TFIT, to cross-check the results, ensuring the perfect equivalency of the results given by the two codes when used with the same parameter sets, and showing how appropriate

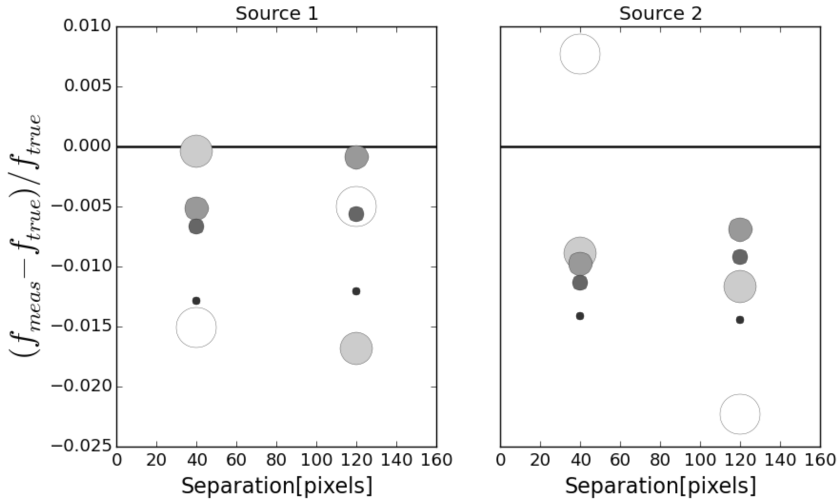


Figure 7. Effects of segmentation on the measured flux of two isolated objects with the same flux and signal-to-noise ratio (two cases with different relative separations of 40 and 120 pixels). The shades and dimensions of the dots is a function of the radius of the segmentation, with darker and smaller dots corresponding to smaller segmentations. See text for more details.

settings of the T-PHOT parameters can ensure remarkable improvements.

For simplicity, here we only show the results from a restricted selection of the tests dataset, which are representative of the performance of T-PHOT in standard situations. The results on the other simulations globally resemble the one we present, and are omitted for the sake of conciseness.

3.1. Code performance and reliability on simulated images

3.1.1. Basic tests

As a first test, we checked the performance of the basic method by measuring the flux of two PSF-shaped synthetic sources, with varying separation and signal to noise ratios. One hundred realizations with different noise maps of each parameter set were prepared, and the averages on the measured fluxes were computed. The aim of this test was twofold: on the one hand, to check the precision to which the fitting method can retrieve "true" fluxes in the simplest possible case - two sources with ideal PSF shape; on the other hand, to check the reliability of the nominal error budget given by the covariance matrix, comparing it to the real RMS of the 100 measurements. Fig. 6 shows three examples of the setup and the results of this test. Clearly, in both aspects the results are reassuring: the average of the 100 measurement (red diamonds) is always in very good agreement with the "true" value, with offset in relative error always well under the $1/(S/N)_{centr}$ limit ($(S/N)_{centr}$ is the value of the signal to noise ratio in the central pixel of the source, corresponding to roughly one third of the total S/N); and the nominal error (red crosses) given by the covariance matrix is always in good agreement with the RMS of the 100 measurements (red circles).

When dealing with extended objects rather than with point-like sources, one must consider the additional problem that the entire profile of the source cannot be measured exactly, because the segmentation is limited by the lowest signal-to-noise isophote. The extension of the segmentation therefore comes to play a crucial role and defining it correctly is a very subtle issue. Simply taking the isophotal area as reported by SEXTRACTOR as `ISOAREA` often underestimates the real extension of the objects. Accordingly, the segmentation of the sources should somehow be enlarged to include the faint wings of sources. To this aim, specific software called DILATE has been developed at OAR and used in the CANDELS pipeline for the photometric analysis of GOODS-S and UDS IRAC data (Galametz et al. 2013). DILATE enlarges the segmentation by a given factor, depending on the original area; it has proven to be reasonably robust in minimizing the effects of underestimated segmentated areas.

Fig. 7 shows the effects of artificially varying the dimensions of the segmentation relative to two bright, extended and isolated sources in a simulated HRI, on the flux measured for that source in a companion simulated LRI. Note how enlarging the segmented area normally results in larger measured fluxes, because more and more light from the faint wings of the source are included in the fit. However, beyond a certain limit the measurements begin to lose accuracy due to the inclusion of noisy, too low signal-to-noise regions (which may cause a lower flux measurement).

In principle, using extended analytical models rather than real high resolution cutouts should cure this problem more efficiently, because models have extended wings which are not signal-to-noise limited. Tests are ongoing to check the performance of this approach, and will be presented in a forthcoming paper.

3.1.2. Tests on realistic simulations

The next tests were aimed at investigating less idealized situations, and have been designed to provide a robust analysis of the performance of the code on realistic datasets. We used the code GENCAT (Schreiber et al. 2015, in preparation) to produce mock catalogs of synthetic extragalactic sources, with reasonable morphological features and flux

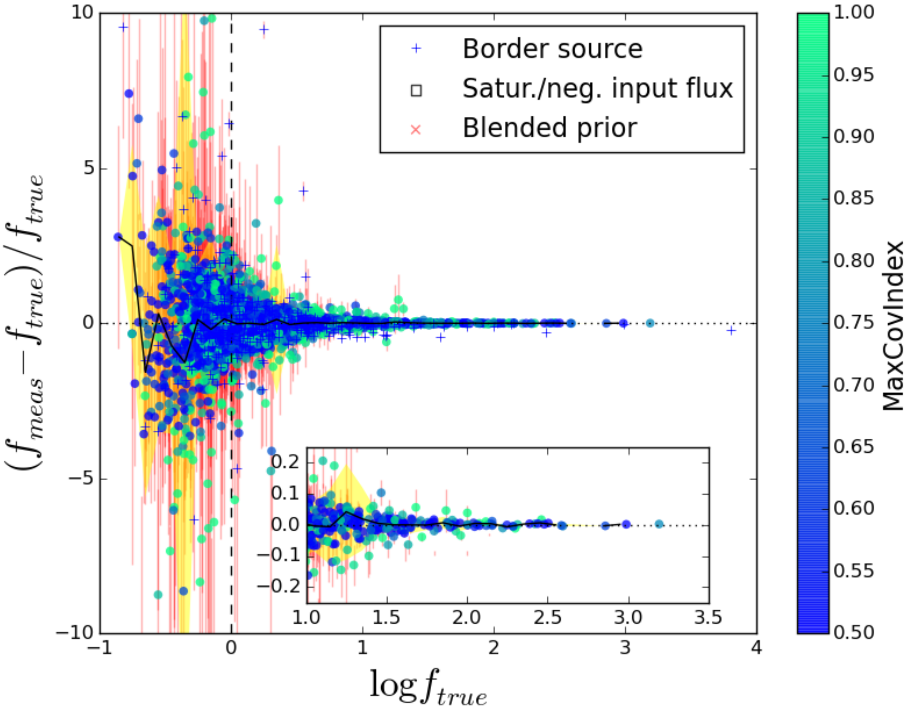


Figure 8. Accuracy of the flux determination in a simulation containing non-overlapping, PSF-shaped sources and "perfect" detection. Relative measured flux difference $(f_{meas} - f_{true})/f_{true}$ is plotted versus logarithm of the input flux f_{true}, for a simulated image populated with PSF-shaped sources (FWHM=1.66″). Each dot corresponds to a single source, with different symbols and colors referring to various diagnostics as explained in the legend and in the colorbar. The black solid line is the average in bins, the yellow shade is the standard deviation. The vertical dashed line shows the limiting flux at 1σ, $f = 1$. The inner panel shows a magnification of the brighter end of the distribution. The fit has been performed on the whole image at once. See text for more details.

distribution[4]. Then, a set of images were produced using such catalogs as an input for SkyMaker. A "detection" HRI mimicking an HST H band observation (FWHM = 0.2″) was generated from the GenCat catalog, using output parameters to characterize the objects' extended properties. Then a set of measure LRI's were produced: a first one was populated with PSF-shaped sources, having FWHM = 1.66" (the typical IRAC-ch1/ch2 resolution, a key application for T-PHOT), while other LRIs were created from the input catalog, mimicking different ground-based or IRAC FWHMs. Finally, we created another HRI catalog removing all of the overlapping sources[5]. This "non-overlapping" catalog was used to create parallel detection and measurement images, to obtain insight into the complications given by the presence of overlapping priors. In all these images, the limiting magnitude was set equal to the assigned zero point, so that the limiting flux at 1σ is 1. Also, the fits were always performed on the LRI as a whole, if not otherwise specified.

Fig. 8 shows the results relative to the first test, i.e. the fit on the image containing non-overlapping, PSF-shaped sources, with a "perfect" detection (i.e. the prior catalog contains all sources above the "true" detection limit), obtained with a single fit on the whole image. The figure shows the relative error in the measured flux of the sources, $(f_{meas} - f_{true})/f_{true}$, versus the log of the real input flux f_{true}; the different symbols refer to the flag assigned to each object, while the color is a proxy for the covariance index, as described in more detail in the next paragraph.

In this case, the only source of uncertainty in the measurement is given by the noise fluctuations, which clearly becomes dominant in the faint end of the distribution. Looking at the error bars of the sources, which are given by the nominal error assigned by T-PHOT from the covariance matrix, one can see that almost all sources have measured flux within 2σ from their "true" flux, with only strongly covariant sources (covariance index $\simeq 1$, greener colors) having $|f_{meas} - f_{true}|/f_{true} > 1\sigma$. The only noticeable exceptions are sources that have been flagged as potentially unreliable, as described in Sect. 2.2. Also note how the average $\Delta f/f$ (solid black line) is consistent with zero down to $f_{true} = S/N \simeq 0.63$.

[4] GenCat is another software package developed within the ASTRODEEP project. It uses GOODS-S CANDELS statistics to generate a realistic distribution of masses at different redshifts, for two populations of galaxies (namely, active and passive) using observed mass-functions. It then estimates all the other physical properties using statistical recipes: morphology, star formation rate, attenuation, optical colors, and sky-projected position.

[5] We proceeded as follows. First, we created a "true" segmentation image using the input catalog and assigning to each object all the pixels in which the flux was $1.005 \times f_{background}$. Then, starting from the beginning of the list, we included each source in the new catalog if its segmented area did not overlap the segmented area of another already inserted source.

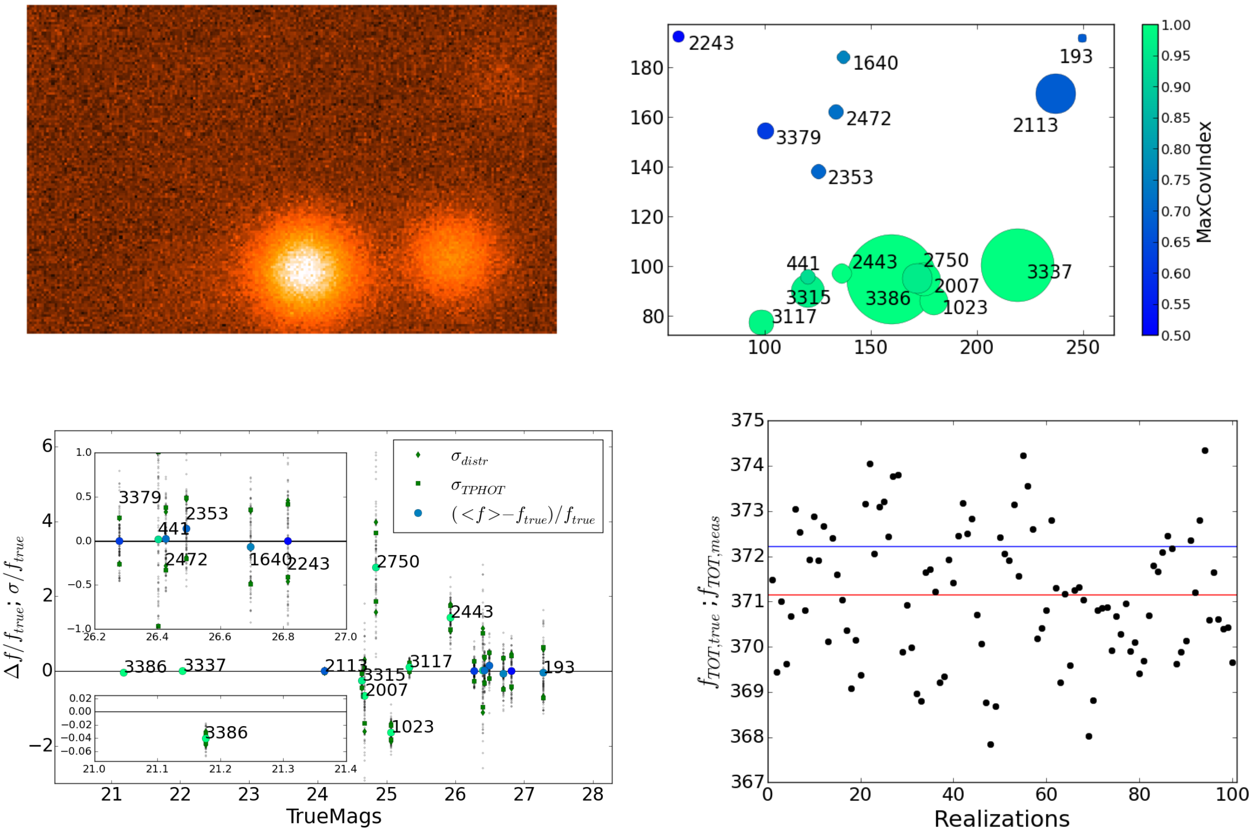


Figure 9. Analysis of a small region including a strongly covarying group of sources. Upper left panel: one of the 100 realizations with different noise maps of the region. Upper right panel: "true" spatial position of all the sources in the region (the color of the dots refer to the covariance index of the sources, as indicated in the colorbar, while their size is proportional to their true flux). Bottom left panel: relative deviation of measured flux from the true flux for each source in the region, as a function of their true magnitude (big dots show the average relative deviation, and their colors refer to their covariance index as in the previous panel; green squares show the nominal uncertainty given by T-PHOT, to be compared with the RMS of the distribution of the 100 measurements (diamonds); small grey dots are the single 100 measurements. Small inner panels show magnifications of regions of interest). Bottom right panel: each dot shows the sum of the measured fluxes for each of the 100 realizations, and the average of this sum (red line) to be compared with the "true" sum (blue line), showing that an overall consistency is guaranteed by the method. See text for more details.

Fig. 9 shows the analysis of a case study in which the fluxes of a clump of highly convariant objects are measured with poor accuracy, and some of the nominal uncertainties are underestimated: a very bright source (ID 3386, $m_{true} = 21.17$) shows a relative difference $(f_{meas} - f_{true})/f_{true} > 3\sigma$. To cast light on the reason for such a discrepancy, the region surrounding the object was replicated 100 times with different noise realizations, and the results were analyzed and compared. The upper panels show (left) one of the 100 measurement images and (right) the position of all the sources in the region (many of which are close to the detection limit). The color code refers to the covariance index of the sources. The bottom left panel shows the relative error in the measured flux for all the sources in the region, with the inner panels showing magnifications relative to the object ID 3386 and to the bunch of objects with $m_{true} \sim 26.5$. Looking at the colors of their symbols, many objects in the region turn out to be strongly covariant. Indeed, while the "bluer" sources in the upper part of the region all have covariant indexes lower than 0.5, the "greener" ones in the crowded lower part all have covariance index larger than 1 (indeed larger than 2 in many cases). This means that their flux measurements are subject to uncertainties not only from noise fluctuations, but also from systematic errors due to their extremely close and bright neighbors. As clearly demonstrated here, the covariance index can give a clue about which measurements can be safely trusted.

The bottom right panel gives the sum of the measured fluxes of all sources in each of the 100 realizations (the blue line is the true total flux and the red line is the mean of the 100 measured total fluxes). It can be seen that the total flux measured in the region is always consistent with the expected true one to within $\simeq 1\%$ of its value.

The bottom line of this analysis is that, although it is not possible to postulate a one-to-one relation (because in mane cases sources having a large covariance index have a relatively good flux estimate, see Fig. 8), the covariance index, together with the flagging code outputted by T-PHOT, can give clues about the reliability of measured flux, and should be taken into consideration during the analysis of the data. Measurements relative to sources having covariance index e.g. larger than 1 should be treated with caution.

In a subsequent more realistic test, we considered extended objects (including morphologies of objects from the GenCat catalog, using FWHM_{HRI}=0.2″ and FWHM_{LRI}=1.66″ and imposing $m_{true,LRI}$=$m_{true,HRI}$ =

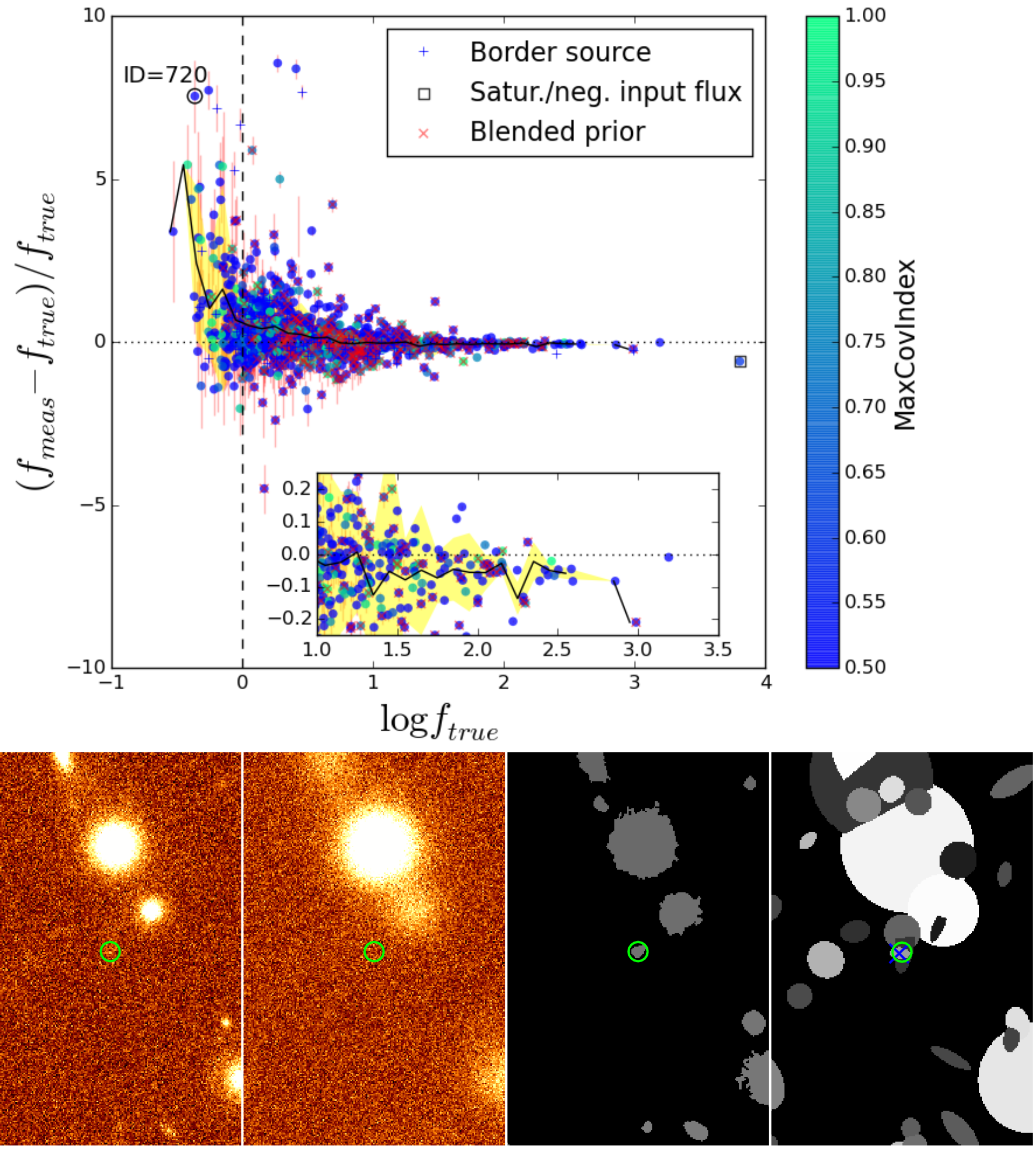


Figure 10. Accuracy of the flux determination in a simulation containing extended objects, overlapping priors and SEXTRACTOR detection. Top: relative flux difference $(f_{meas} - f_{input})/f_{input}$ versus logarithm of the input flux f_{input} for a simulated image populated with extended sources (FWHM=$1.66''$). Symbols and colors are as in Fig. 8. The inner panel shows a magnification of the brighter end of the distribution. The outlier marked with the open black circle, ID=720, is shown in the bottom panel: left to right, HRI (FWHM=$0.2''$), LRI, SEXTRACTOR segmentation map and "true" segmentation map. The green circles show the object detected via SEXTRACTOR, while the blue cross shows its "true" position. See text for more details.

$m_{H160,GenCat}$ for simplicity) and allowed for overlapping priors. To be consistent with the standard procedure adopted for real images, for this case we proceeded by producing an SEXTRACTOR catalog and segmentation map, which were then spatially cross-correlated with the "true" input catalog. The results for this test are shown in Fig. 10. Even in this much more complex situation, the results are reassuring: there is an overall good agreement between measured and input fluxes for bright ($S/N > 1$) sources, with only a few flagged objects clearly showing large deviations from the expected value. However, all fluxes are measured $\simeq 5\%$ fainter than the true values (see the inner box in the same Figure); this is very likely to be the effect of the limited segmentation extension, as already discussed in the previous Section. On the other hand, faint sources tend to have systematically overestimated fluxes, arguably because of contamination from undetected sources. To confirm this, we focus our attention on a single case study (the source marked as ID 720) which shows a large discrepancy from its true flux, but has a relatively small covariance index. An analysis of the real segmentation map shows how in reality the detected object is a superposition of two different sources, which have been detected as a single one, so that the measured flux is of course higher than expected.

One should also note that the uncertainties on the measured fluxes are smaller in this test, because there are fewer priors (only the ones detected by SExtractor are now present), implying a lower rank of the covariance matrix and a lower number of detected neighbors blending in the LRI. This causes a global underestimation of the errors.

3.1.3. Testing different fitting options: cell dimensions

We then proceeded to test the performance of the different fitting techniques that can be used in T-PHOT. To this aim, we repeated the fitting on the LRI, with different methods: using a regular grid of cells of 100×100 pixels, a regular grid of cells of 200×200 pixels, and the *cells-on-objects* method, comparing the results with those from the fit of the whole image at once. The results of the tests are shown in Figs. 11 and 12. The first figure compares the distributions of the relative errors in measured flux for the runs performed on the 100×100 pixels grid, on the 200×200 pixels grid, and on the whole image at once. Clearly, using any *regular* grid of cells worsens the results, as anticipated in Sect. 2.1.5. Enlarging the sizes of the cells yields improvement, but does not completely solve the problem. Note that the adoption of an arbitary grid of cells of any dimension in principle is prone to the introduction of potentially large errors, because (possibly bright) contaminating objects may contribute to the brightness measured in the cell, without being included as contributing sources. A mathematical sketch of this issue is explained in the Appendix (and see also Sect. 3.2).

The second histogram compares the differences between the fit on the whole image and the one with the *cells-on-objects* method. Almost all the sources yield identical results with the two methods, within $(f_{meas} - f_{true})/f_{true} < 0.001$. This proves how the *cells-on-objects* method can be considered a reliable alternative to the single fit method.

Fig. 12 compares the HRI, the LRI, and the residual images obtained with the four methods and their distributions of relative errors, showing quantitatively the difference between the analyzed cases.

In summary, it is clear that an incautious choice of cell size may lead to unsatisfactory and catastrophic outcomes. On the other hand, the advantages of using a single fit, and the equivalence of the results obtained with the single fit and the *cells-on-objects* technique, are evident. As already anticipated, one should bear in mind that the *cells-on-objects* method is only convenient if the overlapping of sources is not dramatic, as in ground-based optical observations. For IRAC and FIR images, on the other hand, the extreme blending of sources would cause the cells to be extended over regions approaching the size of the whole image, so that a single fit would be more convenient, although often still CPU-time consuming.

3.1.4. Testing different fitting options: threshold fitting

As described in Sect. 2.1.5, T-PHOT includes the option to impose a lower threshold on the normalized fluxes of templates so as to exclude from the fit low signal-to-noise pixels. Fig. 13 shows a comparison of the relative errors obtained with three different values of the `THRESHOLD` parameter: $t = 0$, $t = 0.5$ and $t = 0.9$ (this means that only pixels with normalized flux $f_{norm} > t \times f_{peak}$ in the convolved template will be used in the fitting procedure). The differences are quite small, however a non-negligible global effect can be noticed: all sources tend to slightly decrease their measurement of flux when using a threshold limit. This brings faint sources (generally overestimated without using the threshold) closer to their "true" value, at the same time making bright sources too faint. This effect deserves careful investigation which is beyond the scope of this study, and is postponed to a future paper.

3.1.5. Colors

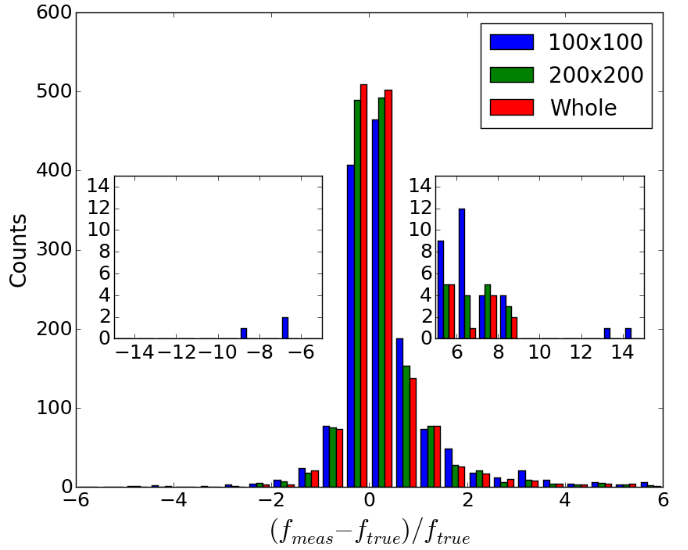
A final test was run introducing realistic colors, i.e. assigning fluxes to the sources in the LRI consistent with a realistic SED (as output by GenCat), instead of imposing them to be equal to the HRI fluxes. We took IRAC-ch1 as a reference filter for the LRI, consistently with the chosen FWHM of 1.66". Furthermore, we allowed for variations in the bulge-to-disc ratios of the sources to take into account possible effects of color gradients. We compared the results obtained with T-PHOT with the ones obtained with two alternative methods to determine the magnitudes of the sources in the LRI: namely, SExtractor dual mode aperture and `MAG_BEST` photometry (with HRI as detection image). The differences between measured and input magnitudes in the LRI, m_{meas}-m_{true}, are plotted in Fig. 14. Clearly, T-PHOT ensures the best results, with much less scattered measurements than both the other two methods, and very few outliers.

3.2. Direct comparison with TFIT on real data

It is instructive to compare the results of a T-PHOT run on real datasets already processed using previous releases of the TFIT package.

To address this, we compared the results of the TFIT CANDELS analysis on the UDS CANDELS *I*-band (Galametz et al. 2013) with a T-PHOT run obtained using the *cells-on-objects* method and different parameters in the kernel registration stage.

Fig. 15 shows the histograms of the differences in the photometric measurements between TFIT and T-PHOT on the same field (UDS Subaru I band) obtained using the *cells-on-objects* method. The differences are evident. Many sources end up with a substantially different flux, because of the two cited factors (a better kernel registration and the different fitting procedure). Note that the majority of the sources have *fainter* fluxes with respect to the previous measurements, precisely because of the effect described in Sect. 3.1.2: fitting using a grid of cells introduces systematic errors assigning light from sources which are not listed in a given cells but overlap with it to the objects recognized as belonging to the cell. To further check this point, Fig. 16 shows some examples of the difference between the residuals obtained with TFIT (official catalog) and those obtained with this T-PHOT run using *cells-on-objects* method, also introducing better registration parameters in the `dance` stage. Clearly, the results are substantially different, with many black spots (sources with spurious overestimated fluxes) disappearing. Also, the registrations appear to be generally improved.



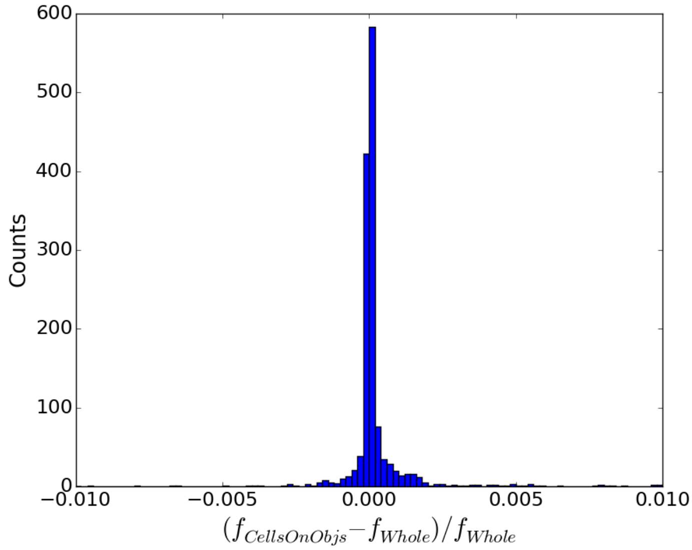


Figure 11. Accuracy of the flux determination. Top panel: for the same simulation described in Fig. 10, the histograms show the results for three different fitting methods: regular grid 100×100 pixels (standard TFIT approach), regular grid 200×200, single fit on the whole image. The small boxes show the extended wings of the histograms, magnified for better viewing. The accuracy increases enlarging the cells and, reaches the best result with the single fit on the whole image. Bottom panel: the histogram shows the relative measured flux difference between the single fit on the whole image and the *cells-on-objects* method. Differences above 1% are very rare.

4. Computational times

As anticipated, T-PHOT ensures a large saving of computational times compared to similar codes like TFIT and CONVPHOT, when used with identical input parameters. For example, a complete, double-pass run on the whole CANDELS UDS field at once (I band; ∼35000 prior sources; LRI 30720×12800 = 400 million pixels; standard TFIT parameters and grid fitting) is completed without memory swaps in ca. 2 hours (i.e., 1 hour per pass) on a standard workstation (INTEL I5, 3.20 GHz, RAM 8 Gb). A complete, double pass run on the GOODS-S Hawk-I W1 field (∼17500 prior sources, LRI 10700×10600 = 100 millions pixels, identical parameters) is completed in ∼20 minutes. For comparison, TFIT may require many hours (∼24) to complete a single pass on this Hawk-I field on the same machine. It must be said that TFIT by default produces cutouts and templates for all the sources in the HRI image; selecting the ones belonging to the LRI field and inputting an ad-hoc catalog would have reduced the computing time, say by a factor of two (i.e., 11 hours for the a single pass). It was not possible to process large images like the UDS field in a single run, because of RAM memory failing. CONVPHOT timings and memory problems are similar to the TFIT ones, although due to different causes (being written in `C`, computation is generally faster, but it employs a slower convolution method and the solution of the linear system in performed as a single fit instead of grid fitting like in TFIT, being much more time consuming).

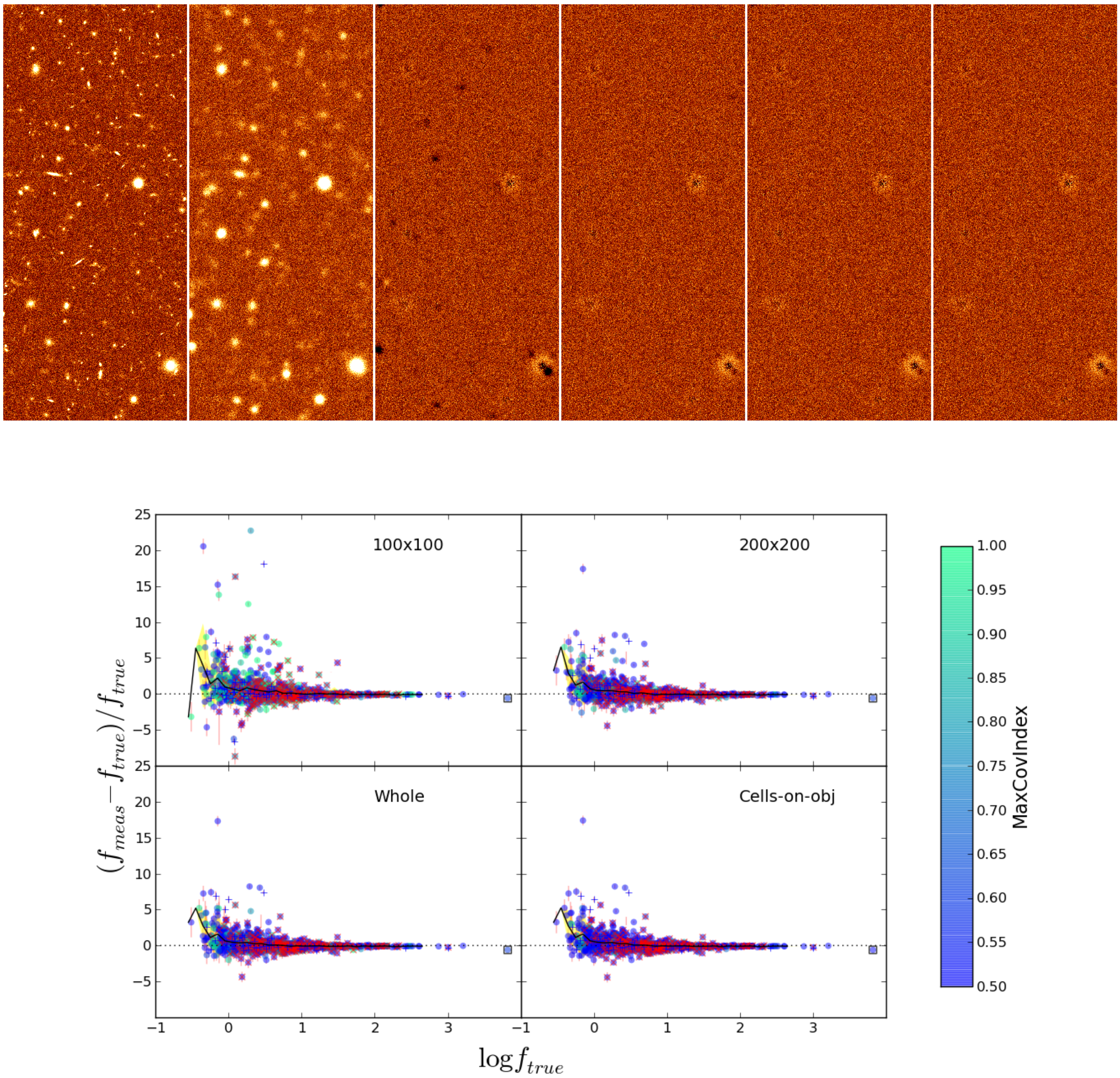


Figure 12. Accuracy of the flux determination. For the same simulation described in Fig. 10, the plots show the results for four different fitting methods. Top panel, left to right: HRI (FWHM=0.2″), LRI (FWHM=1.66″), residuals using a regular grid of 100×100 pixels cells (standard TFIT approach), a regular grid of 200×200 pixels cells, a single fit on the whole image, and the *cells-on-objects* method. The spurious fluctuations in the last two panels are due to segmentation inaccuracies, as in Fig. 10. Bottom panels, left to right and top to bottom: relative measured flux differences with respect to true fluxes, same order as above. Note that the values of the covariance index are different in each case, because of the varying sizes of the cells (and therefore of the relative matrix).

Adopting the *cells-on-objects* (Sect. 2.1.5) method increases the computational time with respect to the TFIT standard cell approach, but it is still far more convenient than the CONVPHOT standard single fit approach, and gives nearly identical results.

Table 2 summarizes the computational times for an extended tests on a set of simulated images having different detection depths (and therefore number of sources) and dimensions, with LRI FWHM=1.66". The simulations have been run on the same machine described above, using three different methods: whole image fitting, *cells-on-objects* and 100×100 pixels cells fitting.

5. Summary and conclusions

We have presented T-PHOT, a new software package developed within the ASTRODEEP project. T-PHOT is a robust and versatile tool, aimed at the photometric analysis of deep extragalactic fields at different wavelengths and spatial resolution, deconfusing blended sources in low resolution images.

T-PHOT uses priors obtained from a high resolution detection image to obtain normalized templates at the lower resolution of a measurement image, and minimizes a χ^2 problem to retrieve the multiplicative factor relative to each source, which is the searched quantity, i.e. the flux in the LRI. The priors can be either real cutouts from the HRI, or a list of positions to be fitted as PSF-shaped sources, or an-

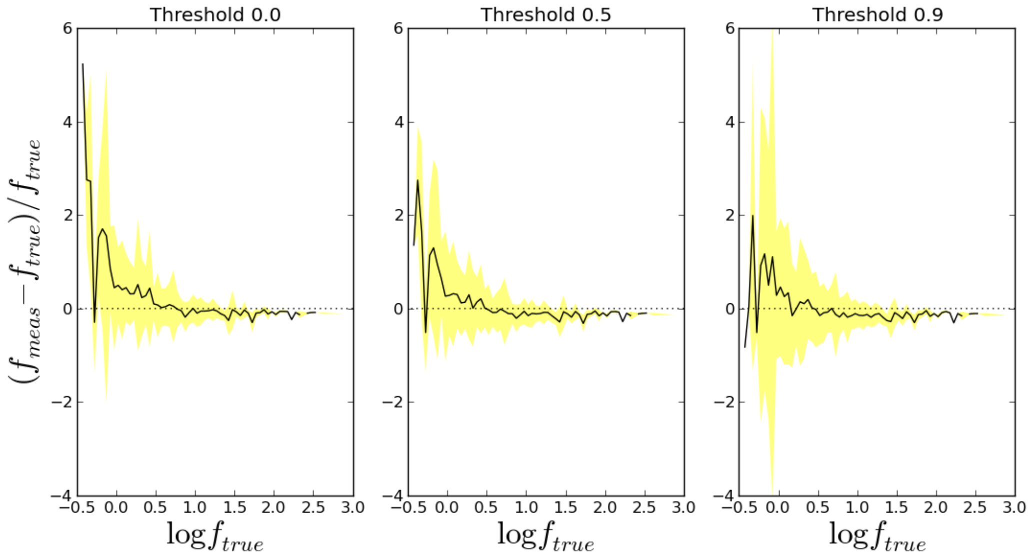


Figure 13. Effects of threshold fitting (Sect. 3.1.4). Mean relative error (black line) and standard deviation (yellow shaded area) for three simulations with different threshold values (0.0, 0.5, 0.9). Only pixels with normalized flux higher than the threshold values are included in the fit. Larger threshold values result in more accurate measurements for faint sources, at the expense of a systematic underestimation of the flux for brighter ones.

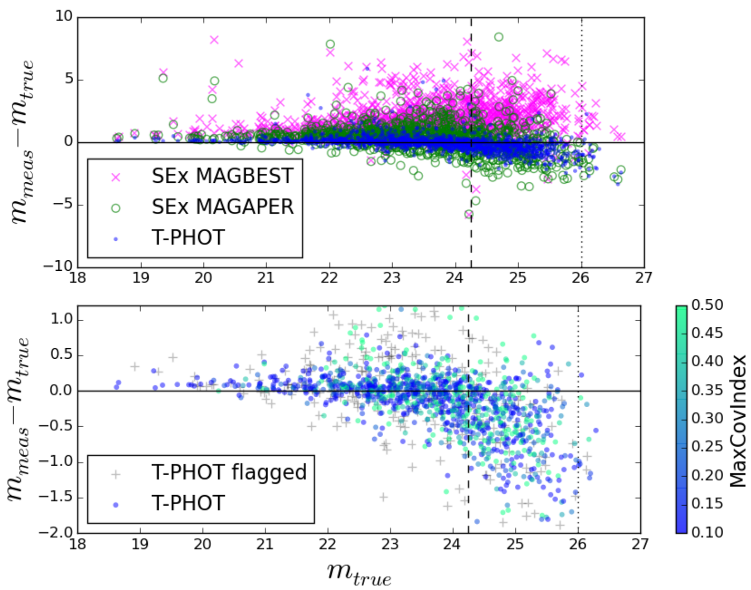


Figure 14. Top: measured magnitude differences ($m_{meas} - m_{true}$) versus "true" input magnitudes m_{true}, for a couple of simulated images populated with extended sources (HRI has FWHM=0.2″ and HST Hband-like fluxes, LRI has FWHM=1.66″ and IRACch1-like fluxes), using three different methods: SExtractor dual-mode aperture, SExtractor dual-mode "best", and T-PHOT. Vertical lines show the 5σ (dashed) and 1σ (dotted) limits of the simulated LRI. Bottom: magnification of the top panel, showing only T-PHOT results, color-coded as a function of the covariance index. See text for more details.

alytical 2-d models, or a mix of the three types. Different options for the fitting stage are given, including a *cells-on-objects* method which is computationally efficient while yielding accurate results for relatively small FWHMs. T-PHOT ensures a large saving of computational time as well as increased robustness with respect to similar public codes like its direct predecessors TFIT and CONVPHOT. With an appropriate choice of the parameter settings, greater accuracy is also achieved.

As a final remark, it should be pointed out that the analysis presented in this work deals with idealized situations, namely simulations or comparisons with the performances of other codes on real datasets. There are a number of subtle issues regarding complex aspects of the PSF-matching techinque, which become of crucial importance when work-

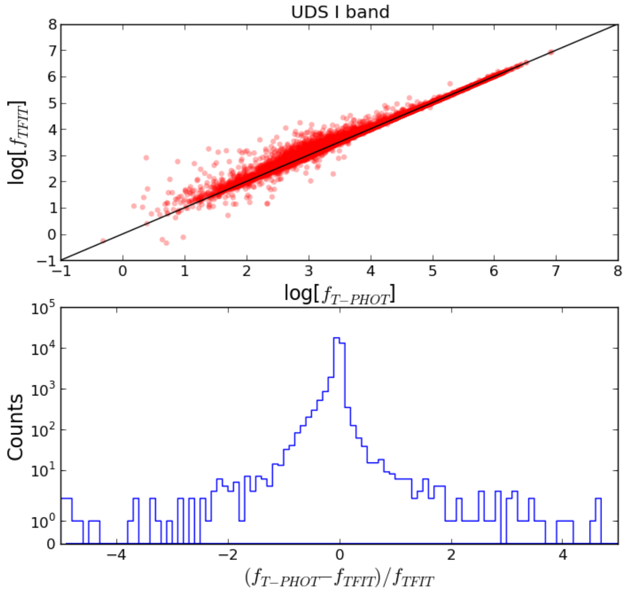


Figure 15. UDS I band TFIT vs. T-PHOT comparison. Top panel: compared measured fluxes. Bottom panel: histogram of relative measured flux difference.

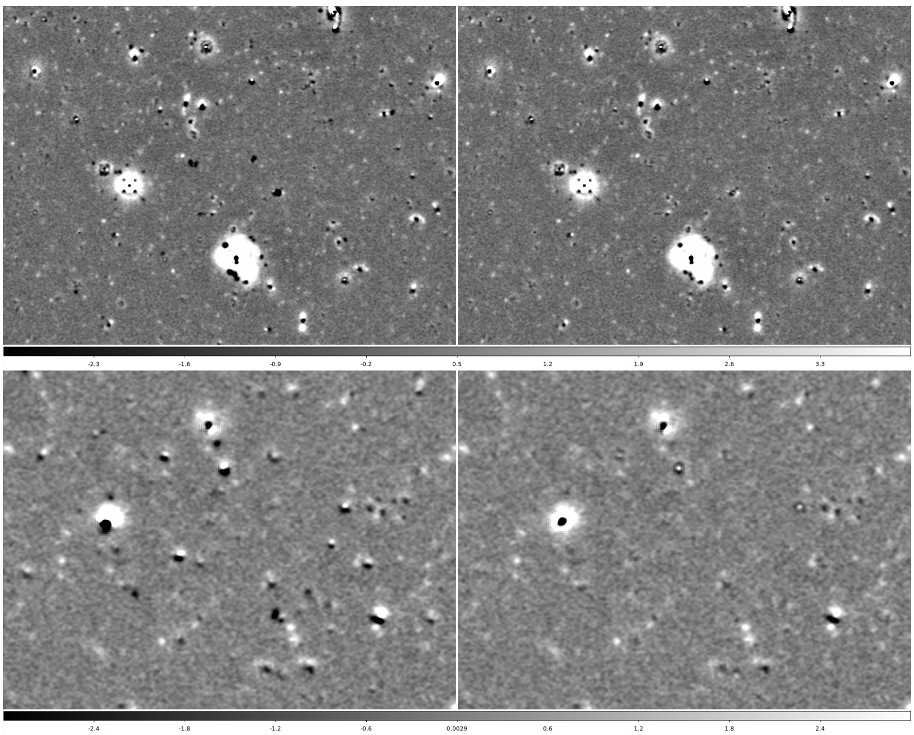


Figure 16. UDS I band TFIT vs. T-PHOT comparison. The panels on the left show two small patches of the "official" CANDELS residual image obtained using T-FIT. The residual images of the same regions are showed in the right panels, this time obtained using T-PHOT with *cells-on-objects* method and improved local kernel registration. Note the disappearence of many spurious black spots.

ing on real data. A simple foretaste of such complexity can be obtained considering the problem described in Sect. 3, i.e. the correct amplitude to be assigned to the segmented area of a source. Work is ongoing on this, and the full discussion will be presented in a subsequent companion paper.

As we have shown, T-PHOT proves to be an efficient tool for the photometric measurements of images on a very broad range of wavelengths, from UV to sub-mm, and is currently being routinely used by the ASTRODEEP community to analyse data from different surveys (e.g. CANDELS, Frontier Fields, AEGIS). Its main advantages with respect to similar codes like TFIT or CONVPHOT can be summarized as follows:

Size[pix] \ $\mathrm{mag}_{lim,det}$	27	28	29
Number of sources			
2500 × 2500	523	1070	1398
5000 × 5000	2104	4237	5561
10000 × 10000	8390	16807	22394
20000 × 20000	33853	65536	65536
Whole image fitting			
2500 × 2500	38" (2")	54" (10")	1'9" (20")
5000 × 5000	3'26" (1'1")	11'9" (7'41")	20'28" (16'1")
10000 × 10000	1h28'22" (1h15'46")	8h26'1" (7h58'10")	21h16'24" (20h27'53")
20000 × 20000	-	-	-
***Cells-on-objects* fitting**			
2500 × 2500	46" (4")	1'11" (16")	1'30" (33")
5000 × 5000	3'1" (18")	4'27" (1'8")	6'3" (2'20")
10000 × 10000	12'27" (1'12")	17'52" (4'31")	25'11" (9'52")
20000 × 20000	51'12" (6'1")	1h34'40" (35'8")	1h43'10" (41'2")
100×100 pixels cells fitting			
2500 × 2500	52" (3")	1'6" (7")	1'14" (9")
5000 × 5000	3'16" (14")	4'22" (29")	4'54" (41")
10000 × 10000	13'4" (56")	17'12" (1'54")	19'47" (2'53")
20000 × 20000	55'24" (6'19")	1h18'38" (15'53")	1h17'17" (17'58")

Table 2. Computational times test for T-PHOT runs on images of given dimensions and limiting magnitude in detection. Each entry of the table reports the total duration of run, the duration the fitting stage alone between parentheses, and the number of fitted sources. The `dance` stage takes most of the CPU time after the fitting routine.

- when used with the same parameter settings of TFIT, T-PHOT is many times faster (up to hundreds of times);
- T-PHOT is more robust, more user-friendly, and can handle larger datasets thanks to an appropriate usage of the RAM;
- T-PHOT can be used with three different types of priors (real high-resolution cutouts, analytical models and/or unresolved point sources) making it a versatile tool for the analysis of different datasets;
- T-PHOT offers many options to perform the fit in different ways, and with an appropriate choice of parameter settings it can give more accurate results.

Future applications might include the processing of EUCLID and CCAT data.

Acknowledgements. The authors acknowledge the contribution of the FP7 SPACE project ASTRODEEP (Ref.No: 312725), supported by the European Commission. JSD acknowledges the support of the European Research Council via the award of an Advanced Grant. FB acknowledges support by FCT via the postdoctoral fellowship SFRH/BPD/103958/2014 and also the funding from the programme UID/FIS/04434/2013. RJM acknowledges the support of the European Research Council via the award of a Consolidator Grant (PI McLure). The authors would like to thank Kuang-Han Huang, Mimi Song, Alice Mortlock and Michal Michalowski, for constructive help and suggestions.

References

Agüeros, M. A., Ivezić, Ž., Covey, K. R., et al. 2005, AJ, 130, 1022
Bertin, E. 2009, Mem. Soc. Astron. Italiana, 80, 422
Béthermin, M., Dole, H., Cousin, M., & Bavouzet, N. 2010, A&A, 516, A43
Bourne, N., Dunlop, J., Bruce, V., et al. 2015
Daddi, E., Cimatti, A., Renzini, A., et al. 2004, ApJ, 617, 746
De Santis, C., Grazian, A., Fontana, A., & Santini, P. 2007, New A, 12, 271
Elbaz, D., Dickinson, M., Hwang, H. S., et al. 2011, A&A, 533, A119
Fontana, A., Santini, P., Grazian, A., et al. 2009, A&A, 501, 15
Galametz, A., Grazian, A., Fontana, A., et al. 2013, ApJS, 206, 10
Grogin, N. A., Kocevski, D. D., Faber, S. M., et al. 2011, ApJS, 197, 35
Guo, Y., Ferguson, H. C., Giavalisco, M., et al. 2013, ApJS, 207, 24
Laidler, V. G., Papovich, C., Grogin, N. A., et al. 2007, PASP, 119, 1325
Obrić, M., Ivezić, Ž., Best, P. N., et al. 2006, MNRAS, 370, 1677
Roseboom, I. G., Oliver, S. J., Kunz, M., et al. 2010, MNRAS, 409, 48
Schreiber, C., Merlin, E., Elbaz, D., et al. 2015
Shu, X., Merlin, E., Elbaz, D., et al. 2015
Stetson, P. B. 1987, PASP, 99, 191
Wang, L., Viero, M., Clarke, C., et al. 2014, MNRAS, 444, 2870
Wang, T., Schreiber, C., Elbaz, D., et al. 2015

Appendix A: The parameter file

Below is a template of the standard first pass parameter file to be given as input to T-PHOT (similar templates for both the first and the second pass are included in the dowloadable tarball). It is very similar to the original TFIT parameter file, and part of the description is directly inherited from it.

A.1. Pipeline

Standard runs can be achieved setting `order standard` and `order standard2`.

A standard firstpass run includes the stages `priors, convolve, fit, diags, dance, plotdance`. The stage `priors` allows for an automatic re-construction of the pipeline depending on the input data given in the following sections (see the documentation included in the tarball). A standard second pass run includes the stages `convolve, fit, diags, archive`. The `archive` stage creates a directory after the name of the LRI, with some specifications, and archives the products of both runs.

```
# T-PHOT PARAMETER FILE

# PIPELINE

# 1st pass
order standard
#priors, convolve, fit, diags, dance, plotdance

# PRIORS STAGE

# Choose priors types in use:
usereal   True
usemodels True
useunresolved True

# Real 2-d profiles
hiresfile       HRI.fits
hirescat        HRI.cat
hiresseg        HRI.seg.fits
normalize       true
subbckg True
savecut true
cutoutdir       cutouts
cutoutcat       cutouts/_cutouts.cat

# Analytical 2-d models
modelscat models/models.cat
modelsdir models

culling false

# Unresolved point-like sources
poscat pos.cat
psffile  psf.fits

# CONVOLUTION STAGE

loresfile       LRI.fits
loreserr        LRI.rms.fits
errtype         rms
rmsconstant     1
relscale        1

FFTconv true
multikernels false
kernelfile      kernel.fits
kernellookup    ch1_dancecard.txt

templatedir     templates
templatecat templates/_templates.cat

# FITTING STAGE

# Filenames:
fitpars          tpipe_tphot.param
tphotcat         lores_tphot.cat_pass1
tphotcell        lores_tphot.cell_pass1
tphotcovar       lores_tphot.covar_pass1

# Control parameters:
fitting coo
cellmask        true
maskfloor       1e-9
writecovar      true

threshold 0.0
linsyssolver lu
clip false

# DIAGNOSTICS STAGES

modelfile       lores_collage_pass1.fits

# Dance:
dzonesize       100
maxshift        1.0
ddiagfile       ddiags.txt
dlogfile        dlog.txt
dancefft false
```

A.2. Priors

Each prior must have an *unique* identificative number (ID) to avoid errors. The user must be careful to give the correct information in this paramfile. Select the priors to be used by switching on/off the relative keywords: `usereal`, `usemodels`, `useunresolved`.

- `hiresfile`: the high resolution, detection image. If a catalog and a segmentation map are given in the two subsequent entries (`hirescat` and `hiresseg`), cutouts will be created out of this image. It is necessary if a catalog of real or model priors are be used. The catalog `hirescat` must be in a standard format: `id x y xmin ymin xmax ymax background SEx_flux`. (`x` and `y` are the coordinates of the source in HRI pixel reference frame, `xmin ymin xmax ymax` are the limits of the segmentation relative to the source in HRI pixel reference frame, `background` is the value of the local background and `SEx_flux` is a reference isophotal flux).
- `poscat`: a catalog of positions for unresolved, point-like sources. No HRI image/segmentation are needed, while the PSF to be used to create the models is mandatory (`psffile`). The catalog must be in the standard format `id x y`.
- `modelscat`: a catalog (with format `id x y xmin ymin xmax ymax background SEx_flux`, as for a standard HRI priors catalog) of model priors. `modelsdir` is the directory in which the stamps of the models are stored. Models with two or more components can be processed, but each component must be treated as a separated object, with a different ID, and a catalog for each component must be given. Catalogs for each component must have the same name, but ending with "_1", "_2" etc.; put the "_1" catalog in the paramfile. *Note that two components of the same object should not have exactly identical positions, to avoid numerical divergencies.*
- `culling`: if `True`, objects in the catalog (real priors and/or models) but not falling into the LRI frame will not be processed; if it is false, all objects in the catalog will be processed (useful for storing cutouts for future reuse on different datasets) and the selection of objects will be done before the convolution stage.
- `subbckg`: if `True`, subtract the value given in the input catalog from each cutout stamp.
- `cutoutdir`: the directory containing the cutouts.
- `cutoutcat`: the catalog of the cutouts, containing the flux measured within the cutout area (which may be different from the `SEx_flux` given in the input catalog, e.g. if the segmentation has been dilated). Note that these are output parameters if you start from the

priors/cutout stage; they are input parameters for the convolve stage.
– normalize: determines whether the cutouts will be normalized or not; it is normally set to true, so that the final output catalog will contain fluxes, rather than colors.

A.3. Convolution

– loresfile, loreserr: the LRI and RMS images. T-PHOT is designed to work with an RMS map as the error map, but it will also accept a weight map, or a single constant value of the RMS from which an RMS map is generated. The errtype specifies which kind of error image is provided. For best results, use a source-weighted RMS map, to prevent the bright objects from dominating the fit.
– relscale: the relative pixel scale between the two images. For example if the HRI has a pixel scale of 0.1 arcsec/pixel and the LRI has a pixel scale of 0.5 arcsec/pixel, the value of relscale should be 5. If the LRI has been manipulated to match the HRI pixel scale and WCS data (e.g. using codes like SWARP by E. Bertin), put relscale 1.
– kernelfile: the convolution kernel file. The kernel must be a FITS image on the same pixel scale as the high res image. It should contain a centered, normalized image.
– FFTconv is True if the convolution of cutouts with the smoothing kernel is to be done in Fourier space (via FFTW3).
– kerntxt may be explicitely put True if one wishes to use a text file containing the kernel instead of a .fits one. T-PHOT supports the use of multiple kernels to accommodate a spatially varying PSF. To use this option, set the multikernels value to true, and provide a kernellookup file (it is automatically produced during the dance stage in the first pass, but it can also be fed externally) that divides the LRI into rectangular zones, specified as pixel ranges, and provides a local convolution kernel filename for each zone. Any objects which fall in a zone not included in the lookup file will use the transfer kernel specified as kernelfile.
– templatedir: the directory containing the templates created in the convolve stage, listed in the catalog templatecat. Note that these are output parameters for the convolve stage, and an input parameter for all subsequent stages.

A.4. Fitting stage

– fitpars, tphotcat, tphotcell, tphotcovar: these are all output parameters. The tfitpars file specifies the name of the special parameter file for the fitting stage that will be generated from the parameters in this file. The others are filenames for the output catalog, cell, and covariance files, respectively.
– fitting: this keyword tells T-PHOT which method to use to perform the fitting (see also Appendix B):
  – coo or 0 for *cells-on-objects*;
  – single or -1 for single fit;
  – single! or -10 for optimized single fit (the LRI is divided in square cells containing roughly 10000 sources each);
  – cell_xdim, cell_ydim, cell_overlap for arbitrary grid of cells.
– cellmask: if true, uses a mask to exclude pixels from the fit which do not contain a value of at least maskfloor in at least one template.
– writecovar: if true, writes the covariance information out to the tphotcovar file.
– threshold: forces to use a threshold on the flux, to only use the central parts of the objects.
– linsyssolver: the chosen solution method, i.e. LU, Cholesky or Iterative Biconjugate Gradient (IBG). LU is default.
– clip: tells whether to loop on the sources excluding negative solutions.

A.5. Diagnostic stages

– modelfile: the .fits file that will contain the collage made by multiplying each template by its best flux and dropping it into the right place. An additional diagnostic file will be created: it will contain the difference image (LRI - modelfile). Its filename will be created by prepending resid_ to the modelfile.
– dzonesize specifies the size of the rectangular zones over which the pixels cross-correlation between LRI and modelfile will be calculated during the dance stage. It should be comparable to the size over which misregistration should be roughly constant; but, it must be large enough to contain enough objects to provide a good signal to the cross-correlation.
– maxshift specifies the maximum size of the x,y shift, in LRI pixel frame, which is considered to be valid. Any shift larger than this is considered spurious and dropped from the final results, and replaced by an interpolated value from the surrounding zones. Ideally, maxshift $\simeq$ 1pixel $\times$ FWHM$_{LRI}$/FWHM$_{HRI}$.
– ddiagfile is an output parameter for the dance stage, and an input parameter for the plotdance stage.
– dlogfile is an output parameter; it simply contains the output from the cross-correlation process.
– danceFFT: if True cross-correlation is to be performed using FFT techniques rather than in real pixels space.

Appendix B: The cells-on-objects algorithm

Experiments on simulated images (see Sect. 3) clearly show that fitting small regions (cells) of the LRI, as done by default in TFIT, may lead to potentially large errors. This is particularly true if the dimensions of the cells are chosen to be smaller than an ideal size, which changes from case to case, and should however always be greater than ∼10 times the FWHM. However, it can be mathematically shown that the "arbitrary cells" method intrinsecally causes the introduction of errors in the fit, as soon as a source is excluded from the cell (e.g., because its center is outside the cell) but contributes with some flux in some of its pixels.

Consider a cell containing N sources. For simplicity, assume that each source i only overlaps with the two neighbours $i-1$ and $i+1$. Furthermore, assume that a $(N+1)$-th source is contaminating the N-th source, but is excluded from the cell for some reason, for example (as in TFIT) because the centroid of the source lies outside the cell.

The linear system for this cell $AF = B$ will consist of a matrix A with only the elements on the diagonal and those with a ± 1 offset as non-zero elements (a symmetric *band* matrix), and the vector B will contain the products of templates of each source with the real flux in the LRI (as a summation on all pixels), as described in Sect. 2.1.4. Given the above assumptions, this means that the N-th term of B will be higher than it should be (because it is contaminated by the external source).

Using the Cramer rule for the solution of squared linear systems, the flux for the object i is given by

$$f_i = \frac{\det A_i}{\det A} \tag{B.1}$$

with A_i a square matrix in which the i-th columns is substituted with the vector B. If for example $N = 3$, for $i = 1$ this gives

$$f_1 = [B_1(A_{22}A_{33} - A_{23}^2) - A_{12}(B_2A_{23} - B_3A_{22})]/\det A \tag{B.2}$$

and since B_3 is larger than it should, f_1 will be overestimated (slightly, if A_{12} is not large, i.e. if sources 1 and 2 do not strongly overlap). On the other hand, for $i = 3$ we have

$$f_3 = [A_{11}(A_{22}B_3 - A_{23}B_2) - A_{12}^2B_3 + B_1(A_{12}A_{23})]/\det A \tag{B.3}$$

and in this case again A_{12}^2 might be small, but the first term given by $A_{11}A_{22}$ will be certainly large, resulting in catastrophic overestimation of f_3. f_2 will of course be underestimated, as it would be easy to show.

From this simple test case it is clear that arbitrarily dividing the LRI into regions will always introduce errors (potentially non-negligible) in the fitting procedure, unless some method to remove dangerous contaminating sources is devised.

The *cells-on-objects* algorithm aims at ensuring the accuracy of the flux estimate while at the same time drastically decreasing computational times and memory requirements. As explained in Sect. 2.1.4, when this method is adopted a cell is centered around each detected source, and enlarged to include all its “potential” contaminant neighbors, and the contaminant of the contaminants, and so on. To avoid an infinite loop, the process of inclusion is interrupted when one of the following criteria is satisfied:

- the flux of the new neighbor is lower than a given fraction f_{flux} of the flux of the central object (the considered fluxes are: if real priors are used, the ones given in the HRI catalog; if unresolved priors are used, the ones read in the pixels of the LRI containing the coordinates of the sources; if analytical models are used, the ones of the models as reported in the HRI models catalog), or
- the template of the neighbor overlaps with its direct previous contaminant for a fractional area lower than f_{area}.

Experiments on simulations have shown that good results are obtained with $f_{flux} = 0.9$ and $f_{area} = 0.25$, and these values are used as constants in the source code.

Note that if a cell is enlarged to more than 75% of the dimensions of the total LRI, T-PHOT automatically switches to the single fit on the whole image.

Appendix C: Suggested best options

Of course, different problems require different approaches to obtain their best possible solution, and users are encouraged to try different options and settings. However, some indicative guidelines to optimize a run with T-PHOT can be summarized as follows.

- Be sure all the required input files exist, have correct format, and paths are correctly given in the parameter file.
- Whenever possible, fit the whole image at once (i.e. put `fitting single` in the parameter file). The more the sources, and the more severe their blending, the more CPU time will be required (see Sect. 4). If the blending is not dramatic, it's safe to switch to the *cells-on-objects* method (i.e. put `fitting coo` in the parameter file). On the other hand, if blending is severe this option would result in redundant fittings because cells would be enlarged to include as many as possible neighbors, increasing the total computing time. In this case, either stick to the whole image fitting, or (depending on the desired degree of accuracy) switch to the TFIT-like cells fitting.
- Spend some time in checking the output catalog, e.g. considering with caution fits relative to sources having flags > 0 and covariance indeces larger than 1.