Acta Polytechnica **65**(1):50–64, 2025
https://doi.org/10.14311/AP.2025.65.0050

Published by the Czech Technical University in Prague

STDWEB: SIMPLE TRANSIENT DETECTION PIPELINE FOR THE WEB

Sergey Karpov

Czech Academy of Sciences, Institute of Physics, Na Slovance 1999/2, 182 00 Prague, Czech Republic
correspondence: karpov@fzu.cz

Abstract. We present a simple web-based tool, STDWeb, for a quick-look photometry and transient detection in astronomical images. It tries to implement a self-consistent and mostly automatic data analysis workflow that would work on any image uploaded to it, allowing to perform basic interactive masking, object detection, astrometric calibration of the image, and building the photometric solution based on a selection of catalogues and supported filters, optionally including the colour term and positionally varying zero point. It also allows you to do image subtraction using either user-provided or automatically downloaded template images, and do a forced photometry for a specified target in either original or difference images, as well as transient detection with basic rejection of artefacts. The tool may be easily deployed allowing its integration into the infrastructure of robotic telescopes or data archives for an effortless analysis of their images.

Keywords: Photometric pipelines, transients, image processing.

1. Introduction

The study of energetic astronomical phenomena, such as supernovae, gamma-ray bursts, or electromagnetic counterparts of gravitational wave events, relies heavily on the ability to detect and analyze transient objects in the vast amount of data produced by modern large-scale sky surveys, as well as smaller-scale follow-up observations. While sky survey experiments usually have dedicated photometric and transient detection pipelines [1–4] tailored specifically for their data and regime of operations, the observations made with "normal" telescopes, especially smaller ones, in follow-up regime often lack such telescope-specific data analysis. This is especially important for the projects involving data from multiple telescopes at the same time, such as collaborative telescope networks[1] or observational efforts involving amateur astronomers.[2]

Unified processing of data from heterogeneous sources (e.g. telescope networks) requires the creation of analysis tools that allow a high level of automatisation, configurability, and overall quality control (visual and user-friendly) in all steps of the pipeline. While the building blocks, both low- and higher-level – like routines in Image Reduction and Analysis Facility [9] (IRAF), DAOPHOT [10] SExtractor [11] or modern Python packages like AstroPy [12] and photutils [13] – are readily available for most of the required tasks, their integration into a reliable pipeline is still a non-trivial effort requiring both the understanding of photometry and the programming affinity.

For the specific image processing requirements of the GRANDMA telescope network [5–7, 14, 15] we initially created a high-level Python library – STDPipe [16] – that aimed towards combining a lot of these building blocks into a set of high-level routines suitable for a rapid construction of telescope-specific pipelines, but it turned out insufficient for handling really heterogeneous data in a controllable and reproducible way. So, highly motivated by the success of popular web-based Astrometry.Net service [17], we decided to try a similar approach of a "heterogeneous data processing portal" for photometry and transient detection tasks.

As a first step towards this goal, we created STDWeb – simple web-based tool for a quick-look photometry and transient detection in astronomical images, that can be either locally deployed, or integrated into the infrastructure of data archive or robotic telescope. Here, we describe the details of its implementation and workflow (Section 2), along with specific details about astrometric and photometric calibration (Section 3), and transient detection (Section 4). We also outline some directions for its further development in Section 5.

2. Implementation and workflow

STDWeb[3] is implemented as a Django [19] server-side application that handles data uploading and presentation, user authorisation and interactions, and presentation of processing results. The actual data processing is handled in the background by Celery [20] distributed task queue that runs custom Python code based on STDPipe [16] library.

[1] An example of such a project actively using STDWeb is GRANDMA network [5–7].

[2] An example of such a professional-amateur observational partnership using STDWeb is the Kilonova-Catcher project [8].

[3] The code of STDWeb is available at [18].

STDPipe[4] is a set of Python routines for astrometry, photometry, and transient detection related tasks, intended for quick and easy implementation of custom pipelines, and for interactive data analysis, implemented as a library that tries to both wrap in pythonic way the access to common command-line tools, such as SExtractor [11], SCAMP [23], SWarp [24], HOTPANTS [25] or Astrometry.Net [17], and to provide extensive set of higher-level routines e.g. for aperture photometry, photometric calibration or transient candidate filtering. STDWeb uses these routines for performing its operation, while adding a set of heuristics on top of that for guessing best parameters for various steps, as well as implements some routines that are specific for its operation and not yet included in the STDPipe. An overview of external tools and libraries used by STDWeb is listed in Table 1.

Name	Purpose	Ref.
External tools		
SExtractor	Object detection	[11]
Astrometry.Net	Astrometric solution	[17]
SCAMP	Astrometric refinement	[23]
SWarp	Image re-projection	[24]
HOTPANTS	Image subtraction	[25]
Python libraries		
Astropy	Overall infrastructure	[12]
Matplotlib	Visualization	[26]
Astro-SCRAPPY	Cosmic ray masking	[27]
Photutils	Aperture photometry	[13]
Astroquery	Catalogue access	[28]
Reproject	Image re-projection	[29]
Scikit-Learn	Machine Learning	[30]
Django	Web interface	
Celery	Backend processing	

Table 1. External software and libraries that STDWeb uses at various steps of its operation. Upper half of the list contains external binaries that should be installed separately while deploying the application, while lower part lists Python libraries that are installed automatically during the deployment.

STDWeb allows the user to either upload the image in FITS format via a web browser, or to select one from local storage (e.g. telescope or project-specific data archive available in the file system) via built-in file browser. Upon importing, a dedicated task is created for the image that will hold the configuration, intermediate processing results and operation logs related to it. The task may be re-visited at any time later in order to either review the results or initiate image re-processing. The task page contains several sub-sections corresponding to different steps of the image processing, with each section having a set of controls for adjusting the parameters relevant for that step, textual log of its execution containing various diagnostic outputs, such as details of running external commands or links to output files, and a set of diagnostic images and plots, each available for better viewing in a pop-up window or for download. Pop-ups for FITS images specifically support adjusting the pixel value scaling and stretching, zooming in, and overplotting of coordinate grids, positions of user-specified targets, detected objects, and catalogue stars. An overall view of the user interface with its various pages is shown in Figure 1. Figure 2 shows an overview of the image processing steps which are discussed in more details below.

3. Image calibration

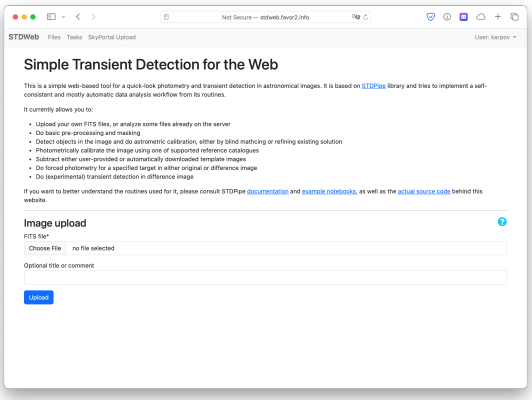
Prior to transient detection, the image – we assume it to be science-ready, i.e. already passed through the instrument-specific pre-processing, such as bias and dark current removal and flat-fielding – has to be properly masked and astrometrically and photometrically calibrated. These steps are described in more detail below.

3.1. Initial image inspection and masking

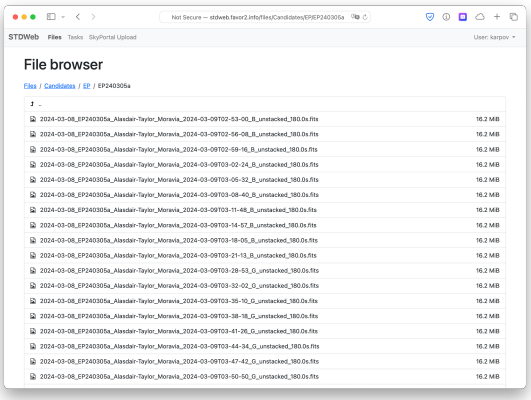
The software will automatically analyse the image trying to extract relevant information from its FITS header using a set of common keywords that define the image timestamp, its saturation level, gain settings, etc. All these settings may also be directly provided by the user, if the information is absent in the header, or is wrong – the software performs some basic checks of whether the values are compatible with actual data range of the pixel values, in order to detect stacked or re-scaled images, and informs the user if it detects some potential problem with it. The software also tries to sanitise the astrometric information in the header, removing or modifying some keywords produced by popular software that break AstroPy WCS module.

Taking into account the saturation level and gain, the software then creates a mask of pixels to be excluded from the consequent analysis, corresponding to saturated pixels and cosmic ray hits. For detecting the latters, AstroSCRAPPY [27] code implementing the original LACosmic [31] algorithm is used. As we make no prior assumptions about the pre-processing already applied to the image (e.g. was it background-subtracted or not), the software automatically constructs the noise model to be used for these routines based on the empirically estimated background noise of the image plus Poissonian noise contribution for the sources above the background. In case the algorithm misbehaves and starts masking normal stars as well (e.g. for undersampled images), this step may be disabled by the user. Moreover, the user may interactively create an additional "custom mask" to be applied on top of the automatically created one, in order to mask, for example, unusable regions of the image that cannot be easily detected by the software (heavily vignetted regions, overscans, imaging artefacts or significant reflections).

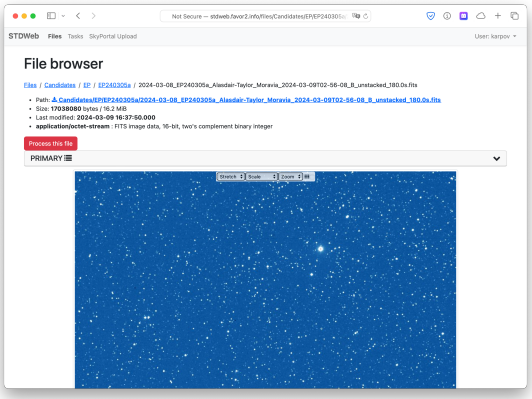
[4]STDPipe is available at [21] and is documented at [22].



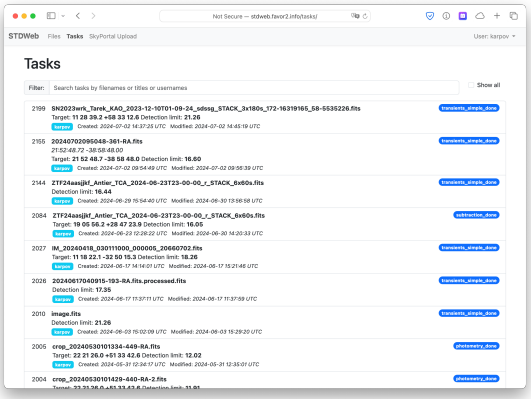
(A). Image upload page.



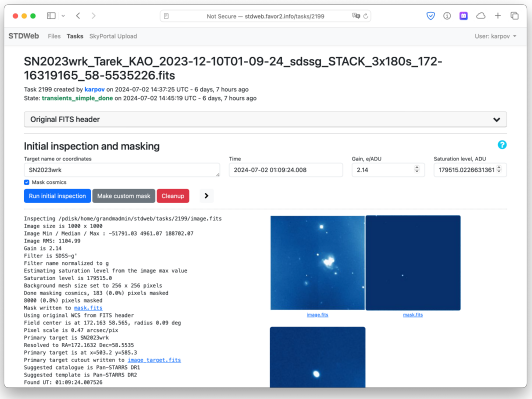
(B). Built-in file browser for accessing local images.



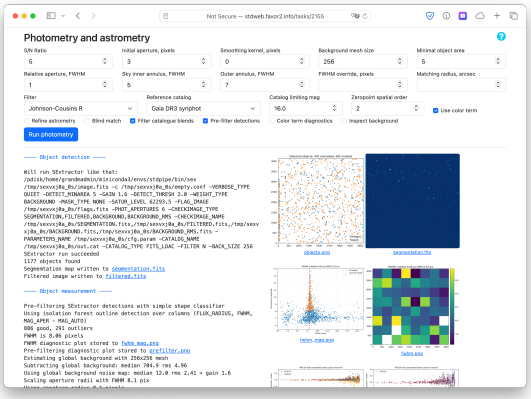
(C). File viewer displaying local FITS image.



(D). List of previously created tasks.



(E). Task page with the options for initial inspection, text output, and diagnostic images.



(F). Task page showing the section with options for object detection and astrometric and photometric calibration.

Figure 1. User interface of STDWeb.

In this step, the user may also provide a list of targets, either as coordinate strings in common formats or as Simbad [32] or TNS [33] resolvable names, to be used for forced photometry later.

3.2. Object detection

While photutils [13] does contain an extensive set of routines for object detection written in pure Python, we decided, primarily for performance reasons, to keep using SExtractor, which is highly optimised and memory efficient even for largest images, for this step in STDWeb. To run it, we construct a minimal set of configuration parameters based on the values optionally specified by the user, such as smoothing ("filtering", in SExtractor terminology) kernel size, and background mesh size. As SExtractor does not internally support masking, we set the image values corresponding to user-specified "custom mask" to `NaN`

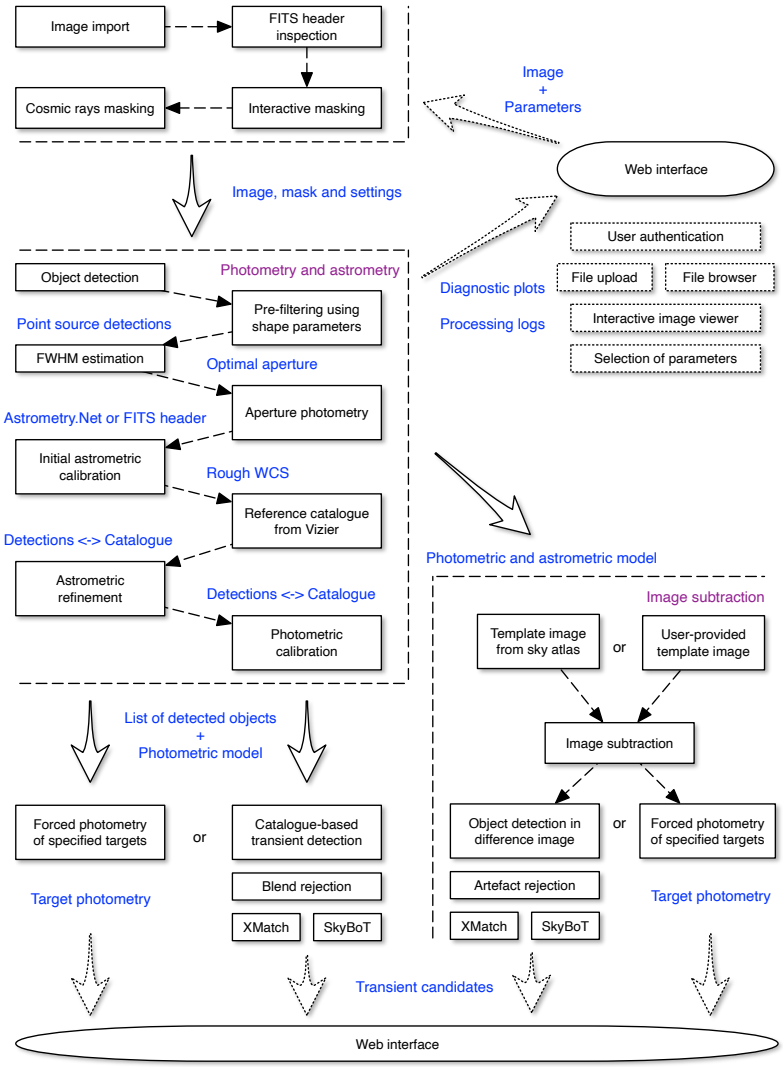


FIGURE 2. Flowchart of the image processing steps implemented in STDWEB.

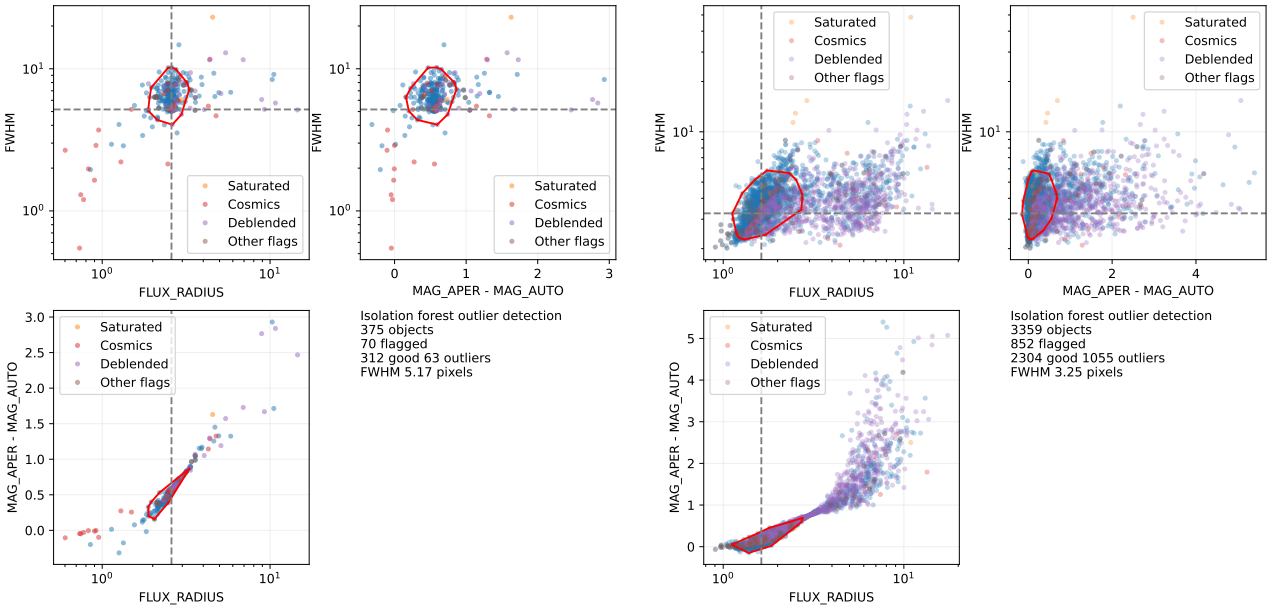
to exclude them from the background estimation. The objects with footprints containing the pixels masked by automatic routines (so, either saturated or marked as cosmic rays) are still properly detected, but flagged in the output so that consecutive processing can ignore them. For the visual quality control, we keep and present to the user the segmentation map, estimated background and background RMS maps, as well as the filtered image produced by SEXTRACTOR.

We also output several shape-related parameters for detected objects that are then used for simple pre-filtering of non-point-source detections in the image. After some experiments, we selected three parameters produced by SEXTRACTOR that are most sensitive to shape variations and flux distribution inside the object footprint:

- `FWHM_IMAGE` – Full Width at Half Maximum, in pixels,
- `FLUX_RADIUS` – Half Flux Radius, in pixels,
- `(MAG_APER - MAG_AUTO)` – the difference between magnitudes measured in fixed circular and Kron-like elliptical apertures.

The space formed by these parameters is shown in Figure 3 for the images with different FWHM values and for various object flags. It is clear that these parameters represent slightly different shape aspects, and thus may be used to filter out various problematic detections corresponding to artefacts or blended sources. To do so, we implemented simple outlier rejection procedure based on isolation forest anomaly detection algorithm [34] as implemented in SCIKIT-LEARN package [30]. Figure 3 shows that the outlier score produced by the algorithm nicely complement the flags set by cosmic ray detection, saturation, and SEXTRACTOR for various artefacts. Therefore, we additionally flag the outliers of the algorithm so that later they may be excluded from various steps of the analysis as problematic detections. The summary of all object flags as implemented in STDWEB is given in Table 2.

As SEXTRACTOR aperture photometry is somewhat limited [35, 36], and as it does not allow to perform forced photometry at user-specified positions, we implemented an additional photometric measurement step based on PHOTUTILS routines [13]. It also allowed



(A). Oversampled images.

(B). Nearly critically sampled images.

Figure 3. Distribution of SExtractor-measured shape parameters for the objects detected in oversampled and nearly critically sampled images. Points of different colours represent various flagged objects – the ones containing saturated pixels or cosmic ray hits inside isophotal aperture, the ones marked as deblended, and having other extraction flags set by SExtractor. The red outline marks the region selected as "good" by the isolation forest outline rejection algorithm, as described in Section 3.2.

Bitmask	Meaning
Flags from SExtractor	
0x001	Aperture flux is significantly affected by nearby stars or bad pixels
0x002	Object is deblended
0x004	Object is saturated
0x008	Object footprint is truncated
0x010	Object aperture data are incomplete
0x020	Object isophotal data are incomplete
0x100	Object footprint contains masked pixels
Flags from aperture photometry	
0x200	Photometric aperture contains masked pixels
0x400	Local background annulus does not have enough good pixels
Flags from higher-level analysis	
0x800	Object is classified as outlier by pre-filtering routine

Table 2. Object flags as set at various stages of the analysis.

us to implement proper local background estimation inside an annulus of user-specified size. In order to facilitate mostly automatic use, we automatically select the aperture radius to be equal to image FWHM, which is a good compromise between the signal to noise (S/N) ratio and source confusion in denser stellar fields, and place the sky annulus between 5 and 7 FWHM values so that it is large enough for an accurate local background estimation. These values can also be adjusted by the user, if needed. As an estimate of image FWHM, we use the median value of 2·`FLUX_RADIUS` for all unflagged objects having S/N > 20, thus rejecting artefacts and non-point-source objects in a way similar to implemented e.g. in PSFEx [37]. Using an additional aperture photometry step allows us to also perform forced photometry at the positions specified by the user, for example, corresponding to transients with already known coordinates, with exactly the same parameters like aperture and background subtraction methods, thus ensuring common effective zero point for all measurements.

After the object measurement, we exclude the detections with sufficiently large photometric errors from the object list, thus keeping only the ones that have at least user-specified signal to noise ratio (by default 5, corresponding to magnitude errors smaller than 0.2). Together with the rejection of detections having too small footprints (`DETECT_MINAREA` parameter of SExtractor) and pre-filtering routine described above, it allows to reject obvious pixel-level and extended artefacts, as well as to keep only objects with properly measured flux.

3.3. Astrometric calibration

If the FITS header does not contain a usable astrometric solution, or if the user suspects it to be unreliable, the software initiates blind solving for it using a local instance of Astrometry.Net [17] code with 2MASS

indices that are sufficient for most of deeper images. In order to ensure that the astrometric solution uses the same positions as other steps of our analysis, as well as for optimising the performance, we run the code directly on the list of object positions detected in the previous step, additionally filtering out all flagged objects to avoid spurious matches due to non-stellar objects and artefacts. Optionally, the user can interactively provide the limits on the sky position and the range of pixel scales used for finding the solution, otherwise the full search for it is performed. We also run the code twice, in order to properly refine the solution to produce WCS with second-order SIP order distortions included.

As soon as the preliminary astrometric solution is known, either from blind matching or from the original FITS header, we request the catalogue from Vizier that will be used for the photometric calibration (see Section 3.4 for more details) and perform additional astrometric refinement using the SCAMP [23] code by passing both the catalogue and list of unflagged detected object positions to it. This way, we ensure that the astrometric solution is always correct, down to the accuracy required for the photometric calibration and image subtraction steps.

3.4. Reference catalogues and photometric calibration

In order to perform the photometric calibration, we use one of the larger catalogues covering large portions of the sky and available in CDS Vizier [38] – Pan-STARRS DR1 [39], SkyMapper Southern Survey DR4 [40], Gaia eDR3 [41], compilative all-sky ATLAS-REFCAT2 [42] catalogue, and the catalogue of synthetic photometry computed from Gaia low-resolution spectra (Gaia DR3 Syntphot [43]). The catalogues are requested from Vizier on the fly, using the ASTROQUERY [28] package, optionally limiting the range of magnitudes relevant for the image to restrict download size. In order to ensure a common photometric system for calibrations done using different catalogues, we derived a set of transformation equations[5] between their individual passbands, Pan-STARRS and Johnson-Cousins systems using a large set of Landolt standards assembled in [45]. When appropriate, these transformations were derived using two base colours (e.g. $g-r$ and $r-i$ for Pan-STARRS and SkyMapper magnitudes conversion to Johnson-Cousins system) to ensure the transformations are valid for both hot and cold stars. However, for converting between closer systems (e.g. Sloan or SkyMapper to Pan-STARRS), single colour regression is usually enough. This way, we have a set of catalogues covering all sky and sufficiently complete for both brighter (Gaia DR3 Syntphot) and fainter (Pan-STARRS and SkyMapper) stars, providing measurements in both the Pan-STARRS and Johnson-Cousins systems, all calibrated to common zero points. While technically it is also possible to derive a set of approximate transformations to and from Gaia eDR3 set of filters, we decided not to do it, leaving it as a separate option for calibrating unfiltered or very wide passband images using G magnitude and $BP-RP$ colour as a basis.

We perform a photometric calibration of the image by building the global zero point model applicable to all detected objects. To do so, we have to take into account various effects, such as the difference between instrumental and catalogue passbands, positional dependence of aperture correction, atmospheric extinction, and uncorrected detector sensitivity. To do so, we use the following photometric model:

$$m_{\mathrm{calib}} = m_{\mathrm{instr}} + \mathrm{ZP}(x,y) + C \cdot \mathrm{Color}, \qquad (1)$$

where

m_{calib} is the magnitude of the object in the catalogue system,

$m_{\mathrm{instr}} = -2.5 \cdot \log_{10}(\mathrm{ADU})$ is the instrumental magnitude computed from the measured object flux in the image,

$\mathrm{ZP}(x,y)$ is the positionally dependent zero-point,

C is the colour term and

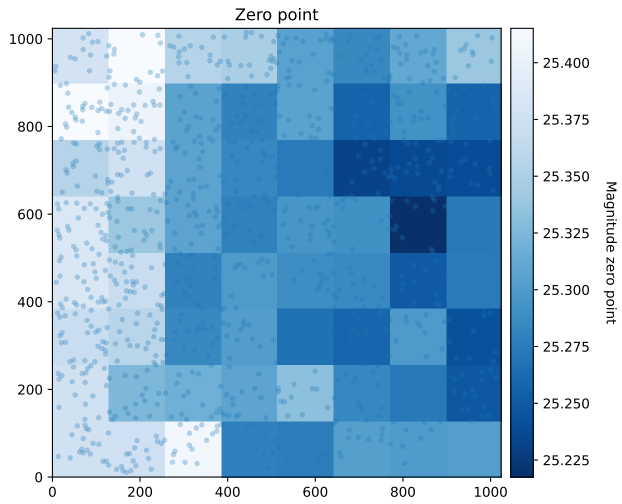
Color is the colour of the object in the catalogue.[6]

Positionally-dependent zero point is modelled as a spatial polynomial of an user-specified order in pixel coordinates, while the choice of catalogue colour depends on the desired passband used for the primary catalogue magnitude. So, for the primary magnitudes in Pan-STARRS *grizy* system, the $g-r$ colour is used (except for z filter, where we selected $r-i$ colour as more appropriate), while for Johnson-Cousins primary magnitudes, the colour is $B-V$, and $BP-RP$ is the one for the Gaia G primary filter. The choice of the primary filter depends on the filter used to capture the image, and can be set either automatically based on the FITS header keywords, or directly specified by the user alongside with the choice of the appropriate reference catalogue. Optionally, the software also performs colour term diagnostics by consecutively using all supported filters for a given catalogue as a primary, and computing the corresponding colour term. Then, the filter that minimizes the colour term – thus, corresponding to the passband closest to the instrumental system of the image – may be selected for the calibration.

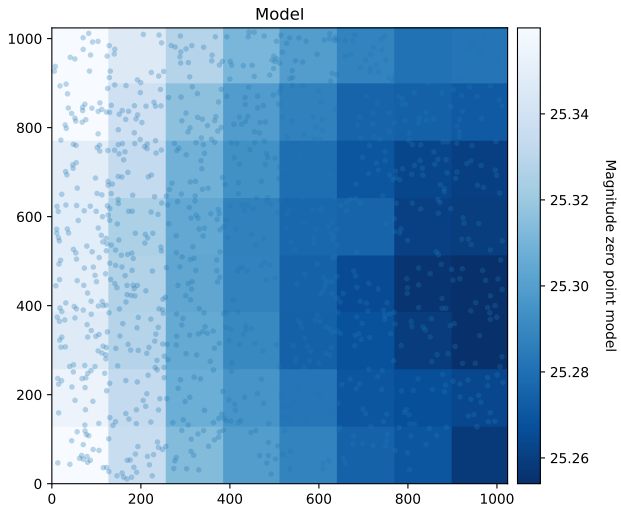
The model is fit using a robust fitter with iteratively rescaled errors to account for potentially underestimated uncertainties of the catalogue and measurements, with additional intrinsic scatter of measurements up to 0.01 mag also being fit as a free model

[5] Conversion coefficients are available directly in the code at GitHub at [44].

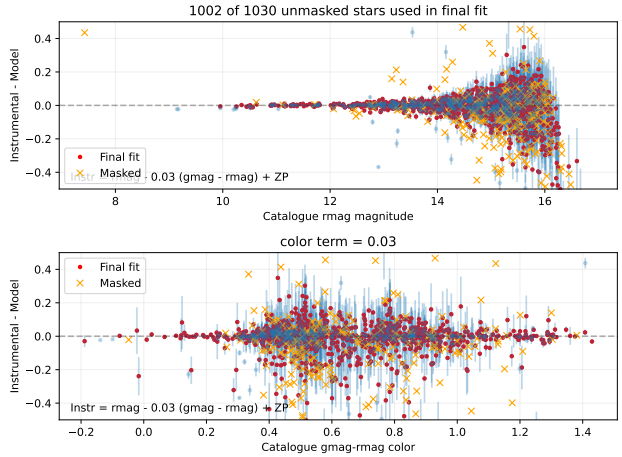
[6] While STDPIPE photometric solver also supports the fitting a position-dependent additive flux term, e.g. to account for a biased background estimation, we decided against using it in STDWEB for reducing the number of free parameters, which is critical for typical narrow-field images with not that many stars present.



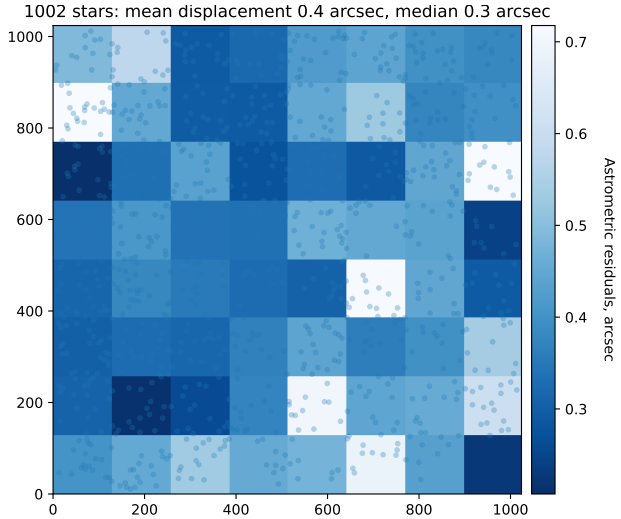
(A). Original map of differences between the instrumental and the catalogue magnitudes.



(B). Model built for it using the second order spatial polynomial and linear colour term.



(C). Residuals between the data and the model as a function of the catalogue magnitude and catalogue $g-r$ colour index.



(D). Map of astrometric displacement between the matched objects and the catalogue stars.

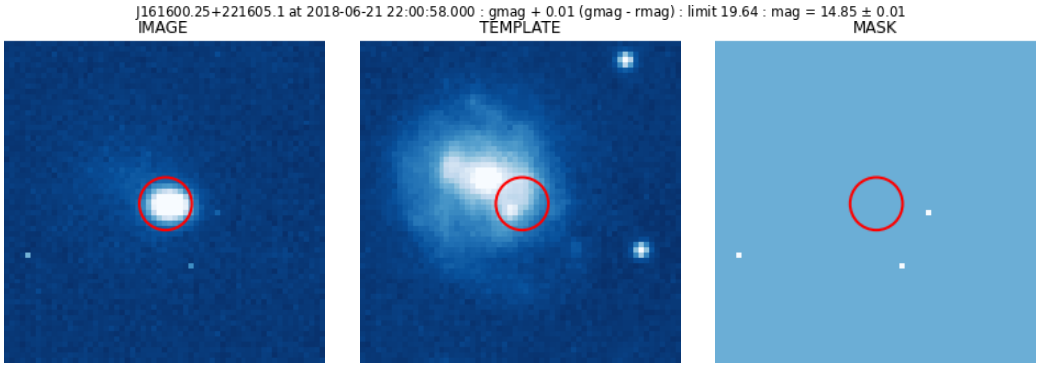
Figure 4. Example of some diagnostic plots generated by STDWeb while fitting the photometric model using both the positionally-dependent zero point and colour term.

parameter to account for uncorrected fixed pattern errors, for example. The fitting involves all pairs between detected objects and catalogue stars positionally matched within $0.5 \cdot \mathrm{FWHM}$ from each other. No additional criteria is applied to ensure the uniqueness of the matching, as non-unique matches just introduce the outliers in the fit that will be iteratively rejected. Instead, we pre-process the catalogue in order to exclude close pairs of stars (closer than $2 \cdot \mathrm{FWHM}$ to each other) that would potentially correspond to blended objects in the image, to avoid using them in the fit. In addition, we exclude all flagged objects corresponding to problematic, saturated, or pre-filtered detections from the fit in order to further reduce the amount of outliers. The fitting produces an extensive set of diagnostic plots, shown in Figure 4 that may be used for evaluating the fit quality.

Once constructed, this photometric model is used to calibrate the measurements of all detected sources, as well as forced photometry at specified target positions (see Figure 5).

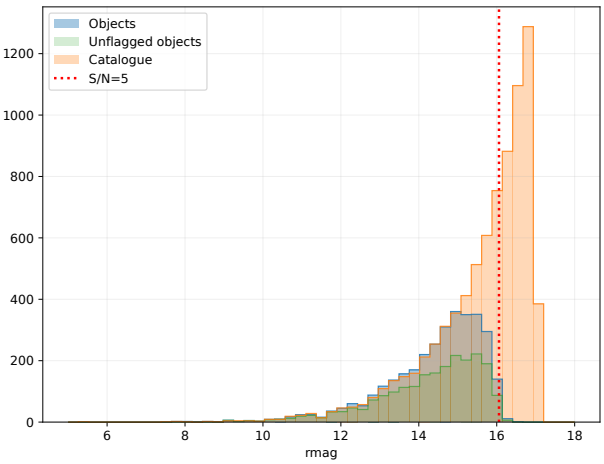
3.5. Detection limit

There are several possible approaches for defining the detection limit – the brightness of the faintest detectable source – of the image, from synthetic source injection to the analysis of noise within the photometric aperture. In STDWeb, we adopted the one based on the studying the dependence of signal to noise ratio on the magnitude in a list of objects detected and measured in the image, as shown in the right panel of Figure 6. To derive it, the software fits the set of corresponding S/N values for all objects with the parametric model consisting of constant background

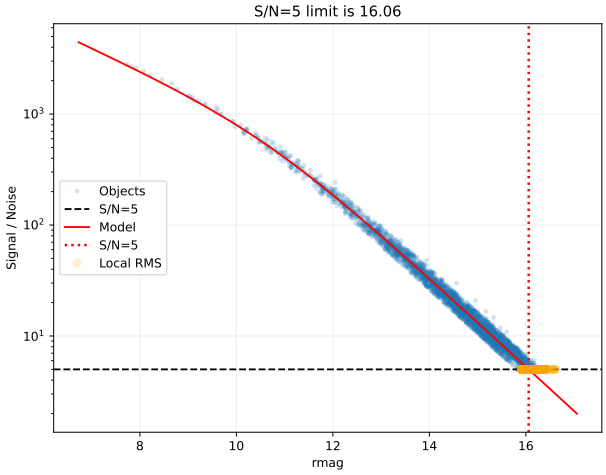


(A). Original image. (B). Atlas image. (C). Mask image.

FIGURE 5. Diagnostic plot for the forced photometry at a user-specified position, containing three cutouts centred on it – original image, atlas image for a visual comparison (either Pan-STARRS or SkyMapper images from `HiPS2FITS` [46] service are used, depending on the hemisphere), and mask image that marks the positions of masked pixels. Reticle (red circle) marks the position of the target in all panels. The title summarises the relevant information, such as calibrated magnitude of the source, effective passband, local detection limit, and timestamp.



(A). The histograms of calibrated magnitudes for detected objects (both total and unflagged ones only), and catalogue magnitudes for stars inside the image.



(B). Signal to noise ratio vs calibrated magnitude plot of the detected objects, fitted with the noise model. Detection limit is defined as an intersection of the model curve with the selected level (S/N = 5 in this example). Also plotted (orange dots) are the detection limits estimated for individual objects solely from the positionally-dependent background flux errors inside the aperture.

FIGURE 6. Example of diagnostic plots generated by STDWEB to determine the detection limit.

noise and Poissonian noise of the object flux that is proportional to the flux square root. While the model is essentially the same as used by photometric routines for estimating the measurement error, the plot still has some scatter due to the positional dependence of the zero point. Thus, the estimated global detection limit corresponds to a mean value over the whole image, while local values may slightly deviate from that. The code also tries to estimate the limits for individual objects solely from the (positionally dependent) noise inside the aperture multiplied by a given S/N ratio (thus, corresponding to the flux that would be detectable with that significance above the background noise), but this approach slightly overestimates the limit due to not taking into account noise contribution from the object flux itself, as seen in the right panel of Figure 6 (orange dots).

The histogram of the magnitudes of the detected objects in comparison with the ones for catalogue stars inside the image may also be used for both estimating the detection limit, as well as completeness and purity of the detections, as shown in the left panel of Figure 6. In this specific example, which corresponds to the same image as the right panel of Figure 3, most of the flagged objects represent the blended ones that are marked as outliers by the pre-filtering routine.

4. TRANSIENT DETECTION

Much of the effort in creating the STDWEB has gone into implementing transient detection routines, as they are the primary motivation behind the original STDPIPE. The code implements two different approaches for detecting the transients in the image –

the one based solely on comparing the lists of the detected objects with different catalogues, and the one involving subtracting the reference image in order to find pixel-level differences between them. Details of their implementations are given below.

4.1. Simple catalogue-based transient detection

For detecting brighter transients that are not situated on a complex background or in a dense stellar fields, the approach based on the direct comparison of objects detected in the image with the lists of known catalogue stars may be sufficient. It is implemented in STDWeb as a "Simple transient detection" section of task analysis, and involves the following steps.

The analysis is based on the list of objects detected in the original object detection step, that includes only the ones that have aperture flux measured with signal-to-noise ratio better than the user-specified threshold. They all have astrometrically calibrated positions and magnitudes calibrated to a standard (Pan-STARRS or Johnson-Cousins) system, with a instrumental photometric system characterised by a linear colour term. First, the code rejects all the objects that have any flag set (see Table 2), except for `0x002` (deblended), or `0x100` (the isophotal footprint contains masked pixels), thus rejecting all saturated ones, the ones marked as cosmic rays, and the ones rejected by the shape-based classifier (see Section 3.2). The latter criterion can be optionally relaxed, in order to be able to detect the transients that are in close proximity of other objects, or just in the images where the shape-based classifier rejects too many good objects, for example.

Then, the user is able to restrict the search to just a portion of the sky by specifying the centre and radius corresponding to, for example, a known error box where the transient is expected. If so, filtering is performed using these parameters, and the objects falling outside the specified cone are excluded from the following analysis. Next, a simple "multi-image mode" can also be enabled – if several images covering the same sky position at approximately the same time are uploaded to STDWeb, then the user can specify the request to reject any object that is not apparent in all these images simultaneously. This is done by specifying task IDs of corresponding additional images that should be already photometrically calibrated, performing the positional cross-match on them, and rejecting the objects that have no counterparts in each of them.

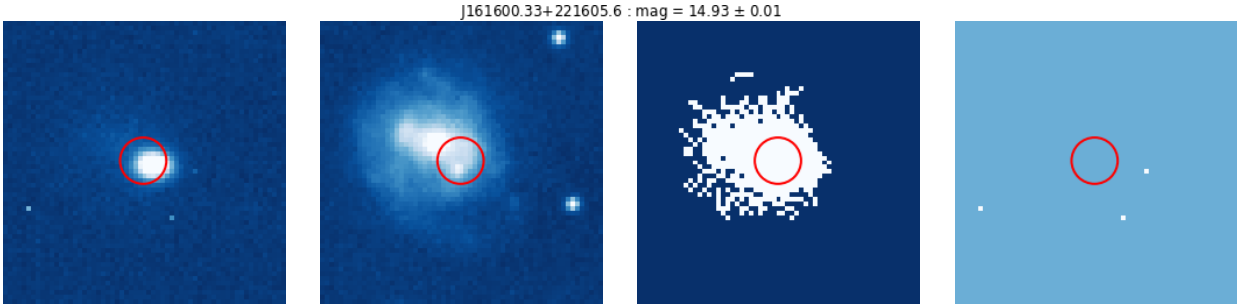
Next, the remaining list of objects is cross-matched with a list of catalogues (Gaia eDR3, then Pan-STARRS DR1, then SkyMapper DR4) using the CDS XMatch [47] service within $2 \cdot \mathrm{FWHM}$ radius. As the list of catalogue fields returned is a bit limited compared to regular CDS Vizier [38], we do not perform a proper augmentation of cross-match results with Pan-STARRS or Johnson-Cousins magnitudes, but instead just select the closest catalogue passband, apply AB to the Vega zero point difference if necessary, and compare the brightness of the detected source with it. We reject all the matches where the object is not brighter than the catalogue star by a specified limit (2 magnitudes by default). This way, we reject all positional matches except for the ones where the catalogue star is too faint compared to the detected transient candidate in order to both avoid the spurious matches with faint background catalogue sources and keep the transients related to the significant increase of the flux from known objects (e.g. nuclear transients or stellar flares). As a final step, if the timestamp of the image is known, the remaining transient candidates are checked against the positions of known Solar System objects at that moment using the IMCCE SkyBoT [48] service, and the candidates positionally matched within $10''$ are rejected. No brightness comparison is performed between the candidate and the Solar System object in this step. Instead, this check can be completely disabled by the user.

The results of the search are presented to the user as a set of candidates, each accompanied by the cutouts from the original image centred on it, the reference image from the relevant sky survey (either Pan-STARRS or SkyMapper) in the closest passband, the detected object footprint, and the mask image, as shown in Figure 7. The whole list is also directly downloadable in a tabular format.

4.2. Image subtraction

Image differencing, or reference image subtraction, is a now *de facto* standard way of detecting transients in astronomical images. There are two primary approaches for that being used in modern large scale sky surveys and their pipelines – the one based on the determination of an optimal convolution kernel that matches the PSF of one image with the other directly in pixel space [49] (often called Alard-Lupton method), and the one that, instead, derives the optimal point source detection statistics [50] (Zackay-Ofek-Gal-Yam, or ZOGY, method). In STDWeb, we support both of these methods, with the primary one being the former, as implemented in the publicly available HOTPANTS [25] code, due to its ability to work without prior knowledge of exact PSFs and relative flux scale of the images.

Image subtraction requires a reference, or template, image with comparable or preferably better detection limit. Modern large-scale sky survey experiments produce such templates as part of their routine observations, by stacking series of historical images covering the same sky position, but for smaller telescopes that perform just occasional follow-up observations, this approach is not available. Therefore, the transient detection on such images requires either acquiring the reference image by observing the same location after the transient faded down, which is impractical in most cases, or falling back to publicly available archival images from some of the large scale sky surveys that



(A). Original image. (B). Atlas image. (C). Detection footprint. (D). Mask image.

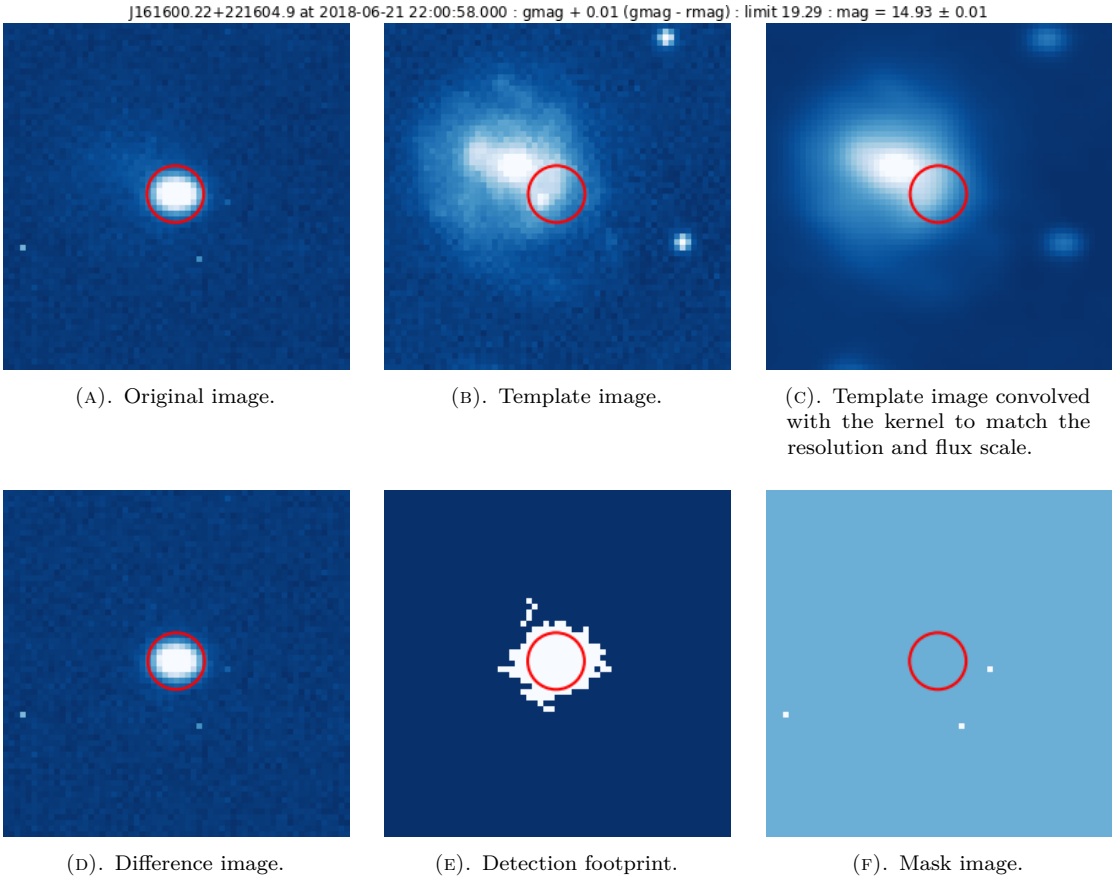
FIGURE 7. Diagnostic plot for the candidate selected by the simple transient detection routine, containing four cutouts centred on its position – the original image, atlas image for visual comparison (either Pan-STARRS or SkyMapper images from the `HiPS2FITS` [46] service are used, depending on the hemisphere), the detection footprint from the SEXTRACTOR segmentation map, and mask image that marks the positions of masked pixels. Reticle (red circle) marks the position of the target in all panels. The title summarises the relevant information, such as calibrated magnitude of the source, effective passband, local detection limit, and timestamp. In this example, the code properly detects the transient, but due to it being superimposed on top of the extended galaxy, the candidate centroid position is biased, and the isophotal footprint includes the flux from the galaxy itself, thus affecting the object's shape parameters that may lead to it being rejected by the shape-based classifier, as described in Section 3.2.

produced sky atlases, such as the Pan-STARRS 3Pi survey [39], or DESI Legacy Surveys [51], with obvious limitations being different epochs of observations, different passbands, and sometimes non-trivial imaging artefacts in these data. STDWEB implements acquiring template images from both of these data sources, automatically performing their re-projection and mosaicking, as well as handling some quirks of their data formats (such as non-linear scaling of Pan-STARRS images) and producing the corresponding masks. However, as neither of these two surveys cover the whole Southern sky, as an optimistic fallback data source, STDWEB also supports getting the data from several surveys like Dark Energy Survey, SkyMapper or ZTF, available through the HiPS2FITS [46] service, the same that powers e.g. Aladin Sky Atlas [52]. The service conveniently allows getting FITS images already projected onto the requested WCS pixel grid, but lacks any means of acquiring corresponding mask information, thus limiting the usability of this approach. Also, STDWEB allows the user to directly upload a custom template image as a separate FITS file that is automatically aligned with original image for the subtraction.

Upon acquisition and re-projection of the template image, the code tries to automatically select the best set of parameters for HOTPANTS based on the image FWHM (it is estimated as part of the object detection step, see Section 3.2), saturation level, similar to how it is done in [1]. Moreover, it constructs the noise model for both the image and the template by combining their estimated background RMS with the corresponding gain values, thus allowing the code to work properly on background-subtracted images (and, to some extent, even on the images with unknown or incorrect gain value). It uses masks from the image and the template to exclude bad image regions from the fitting, and directly passes to the code the positions of good objects to be used for selecting the best positions for placing the sub-stamps used to derive the convolution kernel. We also set up HOTPANTS to always convolve the template to match the image, even if it is not optimal, in order to keep the same zero point for the aperture photometry on the difference image. However, the user can directly modify all input parameters used to call the code, and so revert this behaviour, as well as adjust any other aspect of the HOTPANTS operation.

In order to reduce the impact of a positionally-dependent PSF shape or improper flat-fielding, the software automatically splits the image into a set of chunks of approximately specified size with some overlaps to avoid any position inside it being too close to the edge, processes these chunks separately, and then combines the results together. Specifically, chunk processing includes getting the difference image and noise model for it from the HOTPANTS output, performing a noise-weighted object detection in the difference image using SEXTRACTOR, applying the same prefiltering of detections as described in Section 3.2 using the classifier derived from the original image, and performing the aperture photometry at candidate positions using the same aperture settings as used in the original image in order to keep the same zero point. This way, we derive the list of transient candidates that can be further filtered, like in Section 4.1, using positional criteria, or cross-matching with Vizier catalogues or Solar System object positions (both disabled by default, as most of the stationary objects are subtracted and not being detected as candidates). The final list of candidates is directly available for downloading in a tabular format, and is presented to the user such as is shown in Figure 8.

Image subtraction is not perfect, and prone to various kinds of artefacts related to both numerical instabilities of the kernel fitting routine (typically pro-



(A). Original image. (B). Template image. (C). Template image convolved with the kernel to match the resolution and flux scale.

(D). Difference image. (E). Detection footprint. (F). Mask image.

Figure 8. Diagnostic plot for the candidate selected by the image subtraction based transient detection routine, containing six cutouts centred on its position – the original image, template image, template image convolved with the kernel to match the resolution and flux scale of the original image, difference image, the detection footprint from the SExtractor segmentation map, and the mask image that marks the positions of masked pixels. Reticle (red circle) marks the position of the target in all panels. The title summarises the relevant information, such as calibrated magnitude of the source, effective passband, local detection limit, and timestamp. The candidate shown here is the same as in Figure 7 – it is clearly visible that this time, its position is unbiased and the footprint does not include extra regions, as the underlying galaxy was properly subtracted.

ducing ring-shaped artefacts) and slight differences of object positions between the images, e.g. due to astrometric misalignments or proper motions of the stars (typically producing characteristic "dipole" artefacts, consisting of overlapped positive and negative components of a similar amplitude). While both of them may be, in most cases, successfully filtered out by a shape-based classification (see Section 3.2), we have an additional routine implemented to specifically target the "dipole" artefacts. The routine is based on a sub-pixel adjustment of the relative shift, along with a minor adjustment of the flux scale, between the convolved template and original image in order to minimise the difference between them in the region around the candidate. If the routine (limited to a maximum of 1 pixel shift, and no more than 30 % flux scale adjustment) is able to reduce the χ^2 of the difference by more than three times, or if the final p-value is exceeding 0.01, as shown in Figure 9, then the candidate is considered spurious and is also rejected. Allowing minor optimisation of the flux scale hinders the ability to detect low-amplitude variability of the objects, but we consider it a good trade-off with overcoming the artefacts due to the passband differences between the image and the template. While this algorithm is, by no means, a proper replacement for machine-learning artefact rejection methods, such as convolutional neural networks [53, 54], it is generic enough to be used on arbitrary images without any preliminary training.

If the position of the transient is known in advance, STDWeb also allows skipping the transient detection part of the image subtraction, and directly performing the forced photometry at its position in the difference image.

5. Conclusions and future development

We present STDWeb, a simple web-based photometric calibration and transient detection tool that allows a semi-automated and interactive analysis of arbitrary sky images in order to perform their astrometric and photometric calibration, forced photometry, and transient detection, presenting the user with a rich set of

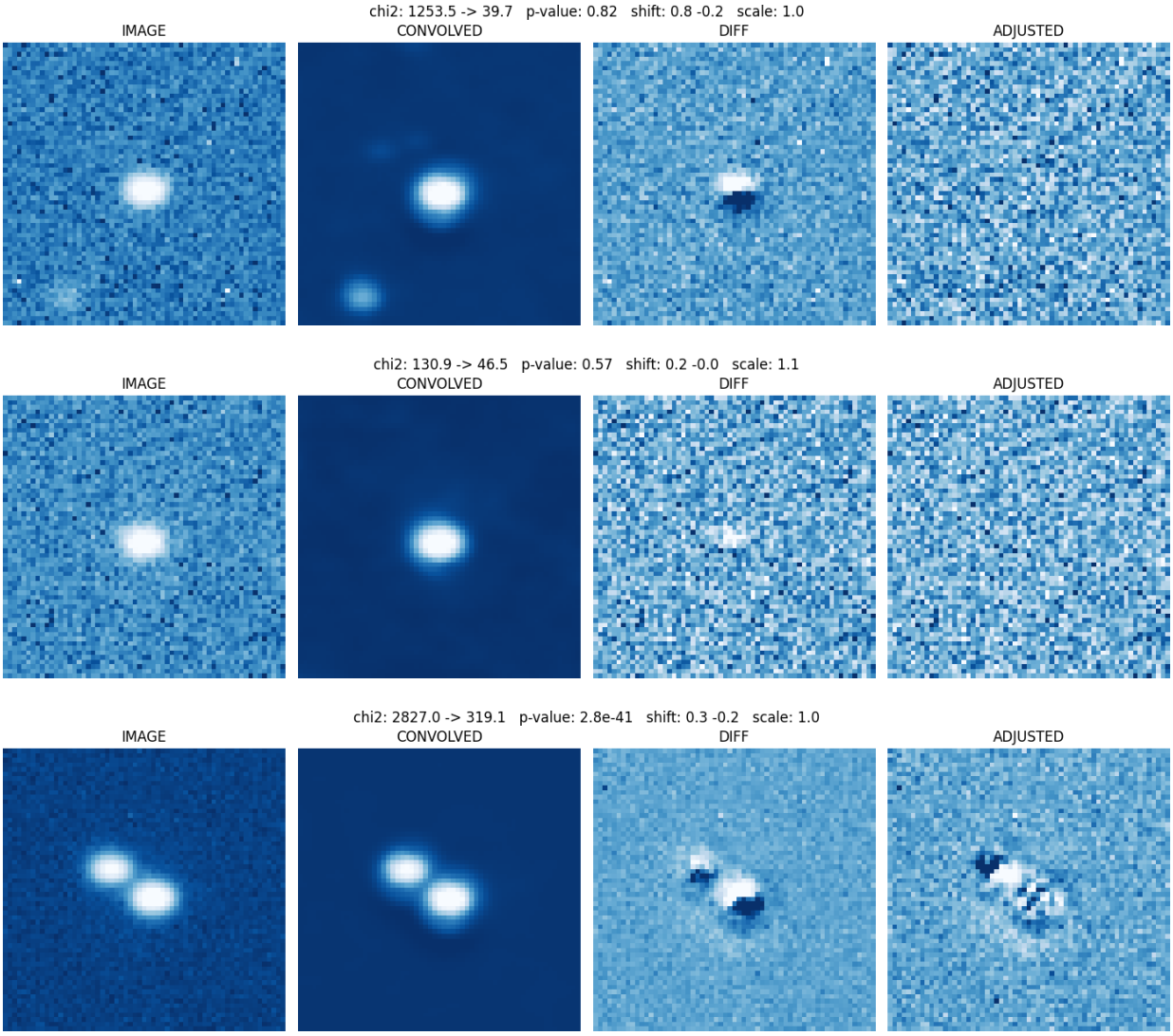


FIGURE 9. Examples of a successful cutout-level sub-pixel adjustment routine that is able to fully eliminate the "dipole" artefact related to the position misalignment between the images (upper row) and the candidate related to slight changes in object brightness (middle row), and mostly eliminate the candidate in a close pair where the relative position of the stars changed due to proper motion (so, after the adjustment, second component "dipole" became even move prominent). Every row shows the cutouts from the original image, template convolved with matching kernel, original difference, and adjusted difference images. The adjustment is performed within $2 \cdot \mathrm{FWHM} \times 2 \cdot \mathrm{FWHM}$ central part of the cutout only. The row title shows the values of chi^2 in this region before and after the adjustment, determined positional shift, and flux scaling.

controls and diagnostic outputs. It is intended to be used both as a simple user-level tool, and as a part of data infrastructure of projects involving diverse sets of images from various sources such as telescope networks. It is already deployed as a standard image processing tool within GRANDMA network [5–7].

The niche occupied by STDWEB lies between the diverse group of lower-level utilities and libraries, such as SEXTRACTOR [11], DAOPHOT [10] or PHOTUTILS [13], and powerful but complicated integrated environments such as IRAF [9], and is closer to more specialised batch processing pipelines, such as AUTOPHOT [55], PHOTOMETRYPIPELINE [56], or VASTT [57], that aim to solve some specific task in an automated manner. However, due to its flexibility and interactivity, STDWEB is also similar to much more user-friendly GUI-based packages, such as ASTROIMAGEJ [58], APERTURE PHOTOMETRY TOOL [59], PIXINSIGHT [60], or SIPS [61]. In contrast to them, STDWEB does not have advanced image stacking or processing capabilities and does not support image sequence analysis that is critical for variable stars photometry (typical application field of these packages), but, it is a web application focused more on transient detection and accurate ensemble photometry of a single image that does not require local installation and is accessible from any device. Moreover, STDWEB is open source and can be easily deployed and adapted for specific tasks and project environments, as opposed to, for example, the proprietary AAVSO VPHOT [62] web tool.

While already feature complete, the code of STDWeb continues being actively developed. Directions for future improvements include overcoming some of its shortcomings listed above, such as better cross-image analysis that is important both for checking whether the detected transient candidates are real, and a quick examination of their temporal behaviour, better rejection of image artefacts, and some of the advanced image pre-processing algorithms for removing low-frequency contaminations, such as fringes or nebulous background components. Another direction is the inclusion of PSF photometry when it is appropriate, as well as overall optimisation of the code so that it is usable even on very large images with many undersampled stars, such as the ones produced by wide-field sky survey telescopes. The user interface is also the subject of future improvements, especially in the part related to quick visual checking of the transient candidates, by, for example, allowing a quick access to sky atlas images and known catalogue objects around them, lists of nearby transients from Transients Name Server [33] and alert brokers such as Fink [63].

The code of STDWeb is publicly available, along with detailed installation and deployment instructions, on GitHub at [18].

Acknowledgements

This work was co-funded by the EU and supported by the Czech Ministry of Education, Youth and Sports through the project CZ.02.01.01/00/22_008/0004596 (SENDISO). This research made use of Astropy, a community-developed core Python package for Astronomy [12]. This research made use of Photutils, an Astropy package for the detection and photometry of astronomical sources [13]. This research made use of the data provided by Astrometry.net [17]. This research made use of hips2fits [46], a service provided by CDS.

References

[1] Y. Cao, P. E. Nugent, M. M. Kasliwal. Intermediate Palomar Transient Factory: Realtime image subtraction pipeline. *Publications of the Astronomical Society of the Pacific* **128**(969):114502, 2016. https://doi.org/10.1088/1538-3873/128/969/114502

[2] E. C. Bellm, S. R. Kulkarni, M. J. Graham, et al. The Zwicky Transient Facility: System overview, performance, and first results. *Publications of the Astronomical Society of the Pacific* **131**(995):018002, 2019. https://doi.org/10.1088/1538-3873/aaecbe

[3] E. A. Magnier, W. E. Sweeney, K. C. Chambers, et al. Pan-STARRS pixel analysis: Source detection and characterization. *The Astrophysical Journal Supplement Series* **251**(1):5, 2020. https://doi.org/10.3847/1538-4365/abb82c

[4] D. Steeghs, D. K. Galloway, K. Ackley, et al. The Gravitational-wave Optical Transient Observer (GOTO): Prototype performance and prospects for transient science. *Monthly Notices of the Royal Astronomical Society* **511**(2):2405–2422, 2022. https://doi.org/10.1093/mnras/stac013

[5] S. Agayeva, V. Aivazyan, S. Alishov, et al. The GRANDMA network in preparation for the fourth gravitational-wave observing run. In D. S. Adler, R. L. Seaman, C. R. Benn (eds.), *Observatory Operations: Strategies, Processes, and Systems IX*, vol. 12186, p. 121861H. International Society for Optics and Photonics, SPIE, 2022. https://doi.org/10.1117/12.2630240

[6] V. Aivazyan, M. Almualla, S. Antier, et al. GRANDMA observations of ZTF/Fink transients during summer 2021. *Monthly Notices of the Royal Astronomical Society* **515**(4):6007–6022, 2022. https://doi.org/10.1093/mnras/stac2054

[7] I. Tosta e Melo, J.-G. Ducoin, Z. Vidadi, et al. Ready for O4 II: GRANDMA observations of Swift GRBs over eight weeks in spring 2022. *Astronomy & Astrophysics* **682**:A141, 2024. https://doi.org/10.1051/0004-6361/202347938

[8] D. Turpin. Kilonova-catcher: A new citizen science project to explore the multi-messenger transient sky. In J. Richard, A. Siebert, E. Lagadec, et al. (eds.), *SF2A-2022: Proceedings of the Annual meeting of the French Society of Astronomy and Astrophysics*, pp. 153–157. 2022.

[9] D. Tody. The IRAF data reduction and analysis system. In D. L. Crawford (ed.), *Instrumentation in astronomy VI*, vol. 627, pp. 733–748. International Society for Optics and Photonics, SPIE, 1986. https://doi.org/10.1117/12.968154

[10] P. B. Stetson. DAOPHOT: A computer program for crowded-field stellar photometry. *Publications of the Astronomical Society of the Pacific* **99**(613):191, 1987. https://doi.org/10.1086/131977

[11] E. Bertin, S. Arnouts. SExtractor: Software for source extraction. *Astronomy and Astrophysics Supplement Series* **117**(2):393–404, 1996. https://doi.org/10.1051/aas:1996164

[12] The Astropy Collaboration. Astropy: A community Python package for astronomy. *Astronomy & Astrophysics* **558**:A33, 2013. https://doi.org/10.1051/0004-6361/201322068

[13] L. Bradley, B. Sipőcz, T. Robitaille, et al. Astropy/photutils: 1.0.0, 2020. https://doi.org/10.5281/zenodo.4044744

[14] S. Antier, S. Agayeva, V. Aivazyan, et al. The first six months of the Advanced LIGO's and Advanced Virgo's third observing run with GRANDMA. *Monthly Notices of the Royal Astronomical Society* **492**(3):3904–3927, 2020. https://doi.org/10.1093/mnras/stz3142

[15] S. Antier, S. Agayeva, M. Almualla, et al. GRANDMA observations of advanced LIGO's and advanced Virgo's third observational campaign. *Monthly Notices of the Royal Astronomical Society* **497**(4):5518–5539, 2020. https://doi.org/10.1093/mnras/staa1846

[16] S. Karpov. STDPipe: Simple Transient Detection Pipeline. *Astrophysics Source Code Library* ascl:2112.006, 2021.

[17] D. Lang, D. W. Hogg, K. Mierle, et al. Astrometry.net: Blind astrometric calibration of arbitrary astronomical images. *The Astronomical Journal* **139**(5):1782, 2010. https://doi.org/10.1088/0004-6256/139/5/1782

[18] S. Karpov. STDWeb – Simple Transient Detection for the Web, 2025. [2025-02-26]. https://github.com/karpov-sv/stdweb

[19] Django Software Foundation and individual contributors. The web framework for perfectionists with deadlines – Django, 2025. [2025-02-26]. https://www.djangoproject.com

[20] Ask Solem & contributors. Celery – Distributed task queue – Celery 5.4.0 documentation, 2023. [2025-02-26]. https://docs.celeryq.dev/

[21] S. Karpov. STDPipe – Simple Transient Detection Pipeline, 2025. [2025-02-26]. https://github.com/karpov-sv/stdpipe

[22] S. Karpov. STDPipe – Simple Transient Detection Pipeline – STDPipe 0.2.2 documentation, 2022. [2025-02-26]. https://stdpipe.readthedocs.io/

[23] E. Bertin. Automatic astrometric and photometric calibration with SCAMP. In C. Gabriel, C. Arviset, D. Ponz, S. Enrique (eds.), *Astronomical Data Analysis Software and Systems XV*, vol. 351 of *Astronomical Society of the Pacific Conference Series*, p. 112. 2006.

[24] E. Bertin. SWarp: Resampling and co-adding FITS images together. *Astrophysics Source Code Library* ascl:1010.068, 2010.

[25] A. Becker. HOTPANTS: High Order Transform of PSF ANd Template Subtraction. *Astrophysics Source Code Library* ascl:1504.004, 2015.

[26] J. D. Hunter. Matplotlib: A 2D graphics environment. *Computing in Science & Engineering* **9**(3):90–95, 2007. https://doi.org/10.1109/MCSE.2007.55

[27] C. McCully, M. Tewes. Astro-SCRAPPY: Speedy Cosmic Ray Annihilation Package in Python. *Astrophysics Source Code Library* ascl:1907.032, 2019.

[28] A. Ginsburg, B. M. Sipőcz, C. E. Brasseur, et al. Astroquery: An astronomical web-querying package in Python. *The Astronomical Journal* **157**(3):98, 2019. https://doi.org/10.3847/1538-3881/aafc33

[29] T. Robitaille, C. Deil, A. Ginsburg. Reproject: Python-based astronomical image reprojection. *Astrophysics Source Code Library* ascl:2011.023, 2020.

[30] F. Pedregosa, G. Varoquaux, A. Gramfort, et al. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research* **12**:2825–2830, 2011.

[31] P. G. van Dokkum. Cosmic-ray rejection by Laplacian edge detection. *Publications of the Astronomical Society of the Pacific* **113**(789):1420, 2001. https://doi.org/10.1086/323894

[32] M. Wenger, F. Ochsenbein, D. Egret, et al. The SIMBAD astronomical database. The CDS reference database for astronomical objects. *Astronomy and Astrophysics Supplement Series* **143**(1):9–22, 2000. https://doi.org/10.1051/aas:2000332

[33] A. Gal-Yam. The TNS alert system. *Bulletin of the American Astronomical Society* **53**(1):2021n1i423p05, 2021.

[34] F. T. Liu, K. M. Ting, Z.-H. Zhou. Isolation forest. In *2008 Eighth IEEE International Conference on Data Mining*, pp. 413–422. 2008. https://doi.org/10.1109/ICDM.2008.17

[35] A. C. Becker, N. M. Silvestri, R. E. Owen, et al. In pursuit of LSST science requirements: A comparison of photometry algorithms. *Publications of the Astronomical Society of the Pacific* **119**(862):1462, 2007. https://doi.org/10.1086/524710

[36] M. Annunziatella, A. Mercurio, M. Brescia, et al. Inside catalogs: A comparison of source extraction software. *Publications of the Astronomical Society of the Pacific* **125**(923):68, 2013. https://doi.org/10.1086/669333

[37] E. Bertin. Automated Morphometry with SExtractor and PSFEx. In I. N. Evans, A. Accomazzi, D. J. Mink, A. H. Rots (eds.), *Astronomical Data Analysis Software and Systems XX*, vol. 442 of *Astronomical Society of the Pacific Conference Series*, p. 435. 2011.

[38] F. Ochsenbein, P. Bauer, J. Marcout. The VizieR database of astronomical catalogues. *Astronomy and Astrophysics Supplement Series* **143**(1):23–32, 2000. https://doi.org/10.1051/aas:2000169

[39] K. C. Chambers, E. A. Magnier, N. Metcalfe, et al. The Pan-STARRS1 surveys. *arXiv* 1612.05560, 2016. https://doi.org/10.48550/arXiv.1612.05560

[40] C. A. Onken, C. Wolf, M. S. Bessell, et al. SkyMapper Southern Survey: Data release 4. *arXiv* 2402.02015, 2024. https://doi.org/10.48550/arXiv.2402.02015

[41] Gaia Collaboration. Gaia Early Data Release 3. Summary of the contents and survey properties. *Astronomy & Astrophysics* **649**:A1, 2021. [Erratum: *Astronomy & Astrophysics* **650**:C3, 2021]. https://doi.org/10.1051/0004-6361/202039657

[42] J. L. Tonry, L. Denneau, H. Flewelling, et al. The ATLAS All-Sky Stellar Reference Catalog. *The Astrophysical Journal* **867**(2):105, 2018. https://doi.org/10.3847/1538-4357/aae386

[43] Gaia Collaboration. Gaia Data Release 3. The Galaxy in your preferred colours: Synthetic photometry from Gaia low-resolution spectra. *Astronomy & Astrophysics* **674**:A33, 2023. https://doi.org/10.1051/0004-6361/202243709

[44] S. Karpov. stdpipe/stdpipe/catalogs.py at master – karpov-svstdpipe, 2024. [2025-02-26]. https://github.com/karpov-sv/stdpipe/blob/master/stdpipe/catalogs.py

[45] E. Pancino, P. M. Marrese, S. Marinoni, et al. The Gaia EDR3 view of Johnson-Kron-Cousins standard stars: The curated Landolt and Stetson collections. *Astronomy & Astrophysics* **664**:A109, 2022. https://doi.org/10.1051/0004-6361/202243939

[46] T. Boch, P. Fernique, F. Bonnarel, et al. HiPS2FITS: Fast generation of FITS cutouts from HiPS image datasets. In R. Pizzo, E. R. Deul, J. D. Mol, et al. (eds.), *Astronomical Data Analysis Software and Systems XXIX*, vol. 527 of *Astronomical Society of the Pacific Conference Series*, p. 121. 2020.

[47] F.-X. Pineau, T. Boch, S. Derrière, A. Schaaff. The CDS cross-match service: Key figures, internals and future plans. In P. Ballester, J. Ibsen, M. Solar, K. Shortridge (eds.), *Astronomical Data Analysis Software and Systems XXVII*, vol. 522 of *Astronomical Society of the Pacific Conference Series*, p. 125. 2020.

[48] J. Berthier, F. Vachier, W. Thuillot, et al. SkyBoT: A new VO service to identify Solar System objects. In C. Gabriel, C. Arviset, D. Ponz, S. Enrique (eds.), *Astronomical Data Analysis Software and Systems XV*, vol. 351 of *Astronomical Society of the Pacific Conference Series*, p. 367. 2006.

[49] C. Alard, R. H. Lupton. A method for optimal image subtraction. *The Astrophysical Journal* **503**(1):325, 1998. https://doi.org/10.1086/305984

[50] B. Zackay, E. O. Ofek, A. Gal-Yam. Proper image subtraction – Optimal transient detection, photometry, and hypothesis testing. *The Astrophysical Journal* **830**(1):27, 2016. https://doi.org/10.3847/0004-637X/830/1/27

[51] A. Dey, D. J. Schlegel, D. Lang, et al. Overview of the DESI legacy imaging surveys. *The Astronomical Journal* **157**(5):168, 2019. https://doi.org/10.3847/1538-3881/ab089d

[52] F. Bonnarel, P. Fernique, O. Bienaymé, et al. The ALADIN interactive sky atlas. A reference tool for identification of astronomical sources. *Astronomy and Astrophysics Supplement Series* **143**(1):33–40, 2000. https://doi.org/10.1051/aas:2000331

[53] D. A. Duev, A. Mahabal, F. J. Masci, et al. Real-bogus classification for the Zwicky Transient Facility using deep learning. *Monthly Notices of the Royal Astronomical Society* **489**(3):3582–3590, 2019. https://doi.org/10.1093/mnras/stz2357

[54] K. Makhlouf, D. Turpin, D. Corre, et al. O'TRAIN: A robust and flexible 'real or bogus' classifier for the study of the optical transient sky. *Astronomy & Astrophysics* **664**:A81, 2022. https://doi.org/10.1051/0004-6361/202142952

[55] S. J. Brennan, M. Fraser. The Automated Photometry of Transients pipeline (AUTOPHOT). *Astronomy & Astrophysics* **667**:A62, 2022. https://doi.org/10.1051/0004-6361/202243067

[56] M. Mommert. PHOTOMETRYPIPELINE: An automated pipeline for calibrated photometry. *Astronomy and Computing* **18**:47–53, 2017. https://doi.org/10.1016/j.ascom.2016.11.002

[57] K. V. Sokolovsky, A. A. Lebedev. VaST: A variability search toolkit. *Astronomy and Computing* **22**:28–47, 2018. https://doi.org/10.1016/j.ascom.2017.12.001

[58] K. A. Collins, J. F. Kielkopf, K. G. Stassun, F. V. Hessman. AstroImageJ: Image processing and photometric extraction for ultra-precise astronomical light curves. *The Astronomical Journal* **153**(2):77, 2017. https://doi.org/10.3847/1538-3881/153/2/77

[59] R. R. Laher, V. Gorjian, L. M. Rebull, et al. Aperture photometry tool. *Publications of the Astronomical Society of the Pacific* **124**(917):737, 2012. https://doi.org/10.1086/666883

[60] Pleiades Astrophoto. Pixinsight – pleiades astrophoto, 2025. [2025-02-26]. https://pixinsight.com

[61] Gxccd.com. SIPS (Scientific Image Processing System), 2025. [2025-02-26]. https://www.gxccd.com/cat?id=146&lang=405

[62] Aavso.org. VPhot. [2025-02-26]. https://apps.aavso.org/vphot/

[63] A. Möller, J. Peloton, E. E. O. Ishida, et al. FINK, a new generation of broker for the LSST community. *Monthly Notices of the Royal Astronomical Society* **501**(3):3272–3288, 2021. https://doi.org/10.1093/mnras/staa3602