Abstract

**V**ariable **S**earch **T**oolkit (VaST) is a software package designed to find variable objects in a series of sky images. It can be run from a script or interactively using its graphical interface. VaST relies on source list matching as opposed to image subtraction. **SEXTRACTOR** is used to generate source lists and perform aperture or PSF-fitting photometry (with **PSFEx**). Variability indices that characterize scatter and smoothness of a lightcurve are computed for all objects. Candidate variables are identified as objects having high variability index values compared to other objects of similar brightness. The two distinguishing features of VaST are its ability to perform accurate aperture photometry of images obtained with non-linear detectors and handle complex image distortions. The software has been successfully applied to images obtained with telescopes ranging from 0.08 to 2.5 m in diameter equipped with a variety of detectors including CCD, CMOS, MIC and photographic plates. About 1800 variable stars have been discovered with VaST. It is used as a transient detection engine in the New Milky Way (NMW) nova patrol. The code is written in C and can be easily compiled on the majority of UNIX-like systems. VaST is free software available at [http://scan.sai.msu.ru/vast/](http://scan.sai.msu.ru/vast/).

**Keywords:** methods: data analysis, techniques: photometric, stars: variables

1. Introduction

Variable stars are important tracers of stellar evolution (e.g. Sherwood and Paul 1975), fundamental stellar parameters (Torres et al. 2010), 3D structure of our Galaxy (Pietrukowicz et al. 2015, Dekany et al. 2015, Gran et al. 2016) and beyond (Lah et al. 2005, Subramanian and Subramaniam 2012, Hoffmann and Macri 2015, Jacyzyn-Dobrzeniecka et al. 2017) as well as various astrophysical processes related to accretion (Osaki 1996, Mukai 2017), ejection (Russell et al. 2016) and strong magnetic fields (Cropper et al. 1989). It is believed that only a few per cent of the variable stars easily accessible to ground-based photometry are currently known (Samus and Antipin 2013). The reason is that contemporary CCDs are very sensitive and small and hence image only a small field of view to a high limiting magnitude.

The next generation surveys Gaia (Gaia Collaboration et al. 2016, Clementini et al. 2016), VVV (Minniti et al. 2010), Pan-STARRS (Chambers et al. 2016), LSST (Ridgway et al. 2014), NGTS (Wheatley et al. 2017) and TESS (Ricker et al. 2015) employing large mosaic cameras (or multiple small cameras in case of NGTS and TESS) are expected to greatly increase the number of known variable stars. Still, these surveys have their limitations in terms of observing cadence, sky coverage, accessible magnitude range and survey lifetime. All this leaves room for variability searches with other instruments and different observing strategies. Photometric measurements needed to detect stellar variability are relatively easy to perform (compared to spectroscopy and polarimetry of objects with the same brightness), so even small-aperture telescopes are useful for finding and studying variable stars.

Large time-domain surveys employ custom-built pipelines (e.g. Bertin et al. 2002, Laher et al. 2014, Kessler et al. 2015, Mauro et al. 2015) to perform photometric data reduction and variable object detection. These pipelines are fine-tuned for the particular equipment and observing strategies employed by these surveys and, while often sharing many common pieces of code, require intervention of a software engineer to adopt them to another telescope or camera. Developing a purpose-built pipeline for a small observing project is often impractical. Instead, one would like to have data reduction software applicable to a variety of telescopes and cameras.

The problem of extracting variability information from surveys, in practice, has not been completely solved. New variable stars are still being identified in the NSVS survey data (e.g. Khruslov 2013, Sergeev et al. 2014) while its observations were completed in 1999–2000 (Wozniak et al. 2004). A number of recent ground-based exoplanet transit surveys, despite having sufficient photometric accuracy and sky coverage for detecting the majority of bright variable stars, have so far provided only limited information on individual objects (e.g. Rodriguez et al. 2013, Norton et al. 2016) or specific classes of objects (e.g. Devor et al. 2008, Norton et al. 2011, McQuillan et al. 2012, Holdsworth et al. 2014, Labadie-Bartz et al. 2016). Better variability detection algorithms, open data-sharing policies and interfaces to published time-series that allow non-trivial searches

*This code is registered at the ASCL with the code entry ascl:1704.005.
in the whole database rather than providing access to a limited number of objects at a time are needed to fully exploit the information hidden in the data. A software to perform variability search in a set of lightcurves imported from a survey archive and visualize the search results may be useful for information extraction and debugging fully-automated search procedures.

Photographic plates used to be the primary type of light detectors in astronomy in the 20th century. Direct images of the sky recorded on glass plates contain information about the positions and brightness that celestial objects had decades ago. This information may be useful on its own or as the first-epoch for comparison with modern CCD measurements. The plates are stored in archives in observatories around the world. Many observatories are digitizing their collections in an effort to preserve the information stored on the plates and make it more accessible. At the time of writing, only the DASCH (Grindlay et al. 2012) and APPLAUSE (Groot et al. 2014) archives provide source catalogs and photometry derived from the plates while other archives provide only images. Performing photometry on digitized photographic images is a non-trivial task (Bacher et al. 2005, Tang et al. 2013, Wertz et al. 2016) that cannot be done well with conventional photometry software developed for CCD images. The conventional software relies on the assumption that an image sensor responds linearly to the number of incoming photons. This assumption is violated for photographic emulsion as well as for some types of contemporary light detectors including microchannel plate intensified CCDs (MCCs) used in space-based UV-sensitive telescopes Swift/UVOT (Poole et al. 2008, Breeveld et al. 2010, see also Brown et al. 2014), XMM/OM (Mason et al. 2001) and fast ground-based cameras (Karpov et al. 2012). There is a need for a user-level software capable of performing photometry on images obtained with non-linear detectors.

A number of software packages for detection of variable objects have been developed recently. Some of them feature a graphical user interface (GUI) and are aimed at processing small datasets, while others have command-line interfaces and are meant as a complete data processing pipeline or as building blocks for constructing one. LEMON is a pipeline based on SExtractor (Bertin and Arnouts 1996) and PyRAF for automated time-series reduction and analysis (Terón and Fernández 2011). It takes a series of FITS images as an input and constructs lightcurves of all objects. PI (Mommer) is an automated Python pipeline based on SExtractor and SCAMP (Bertin 2008). PP produces calibrated photometry from imaging data with minimal human interaction. While originally designed for asteroid work, the code is be applicable for stellar photometry. LEMON and PP are very similar to VAST in spirit, while they differ in technical implementation and user interface. C-MUNIPACK/MUNIW ST offers a complete solution for reducing observations of variable stars obtained with a CCD or DSLR camera. It runs on Windows and Linux and provides an intuitive GUI. Fotorad is a Windows CCD photometry package providing capabilities similar to Muniwi. ASTROK I (Burdanov et al. 2014, 2016) corrects for atmospheric transparency variations by selecting an optimal set of comparison stars for each object in the field. IRAF11 and PHOT/APPHTO task is meant to be used to generate input photometric data for Astrokit. Variable star candidates are selected in Astrokit with Robust Median Statistics (Rose and Hintz 2007). FITSH12 is a collection of tasks for advanced image manipulation (including stacking) and lightcurve extraction using aperture, image subtraction, analytic profile modeling and PSF-fitting photometry. VARTOOLS13 (Hartman and Bakos 2016) implements C a collection of advanced lightcurve analysis methods providing a command-line interface to them. HOTPANTS14 (Becker 2015) is designed to photometrically align one input image with another, after they have been astrometrically aligned with software like WCSREM.15 SWARP (Bertin et al. 2002) or MONTAGE (Jacob et al. 2010). HOTPANTS is an implementation of the Alard (2000) algorithm for image subtraction. The program is intended as a part of a transient detection/photometry pipeline. ISIS16 (Alard and Lupton 1998) is a complete package to process CCD images using the image subtraction method. It finds variable objects in the subtracted images and builds their light curves from a series of CCD images. DIAPI17 (Rozyczka et al. 2017) is able to identify variable stars via image subtraction, implemented in C. TRAF18 (Swinbank et al. 2015) is a PYTHON and SQL based pipeline for detecting transient and variable sources in a stream of astronomical images. It primarily targets LOFAR radio astronomy data, but is also applicable to a range of other instruments (including optical ones).

Most of the above packages were not available at the time VAST development was started (Sokolovsky and Lebedev 2005). Many of them cannot construct lightcurves without finding a plate solution with respect to an external star catalog. None of the above software addresses the issue of photometry with non-linear imaging detectors. VAST provides a combination of features not yet offered by other software, including the ability to process thousands of images with tens of thousands of stars and interactively display the results in a GUI.

VAST is designed as a user-friendly software implementing
the full cycle of photometric reduction from calibrating images to producing lightcurves of all objects within a field of view and detecting variable ones. VaST is capable of handling images obtained with non-linear detectors such as photographic plates and MICs. The software may be applied to images obtained with telescopes of any size with minimal configuration. VaST can be used interactively to inspect a set of images of one field in a PGPLOT-based GUI. As soon as optimal source extraction and lightcurve post-processing parameters have been identified in the interactive mode, VaST may proceed non-interactively and produce lightcurves for subsequent variability searches with VARTOOLS or custom-built scripts. VaST lightcurve visualization tools and built-in variability statistics routines may also be used to inspect and process lightcurves obtained with other software.

The outline of the paper is as follows. Section 2 presents the overall design of the program centered around the idea of distinguishing variable stars from non-variable ones using “variability indices”. Section 3 describes the steps performed by VaST when processing a set of images. Section 4 presents a specialized transient detection mode that does not rely on variability indices. In Section 5 we present some general remarks on the VaST development procedures. Section 6 gives an account of several notable examples of applying VaST to real data processing. We present a summary in Section 7. Appendix A describes common use cases that can be solved with VaST. Appendix B presents the list of the command-line arguments to be used with the main program of the toolkit. Appendix C describes the log files that summarize image processing results.

2. VaST design and the challenge of variability detection

VaST is designed to accomplish three main tasks:

1. Construct lightcurves of all objects visible at the input series of images of a given star field.
2. Compute variability indices to quantify “how variable” each lightcurve appears to be.
3. Visualize the variability indices, lightcurves and images to let the user decide which objects are actually variable.

In addition, VaST includes tools to calibrate the magnitude zero-point of the produced lightcurves, check if a given object is a cataloged variable, apply the heliocentric correction to a lightcurve, etc. The software modules implementing each of the above tasks are united by a common internal lightcurve file format. They can be run together or independently of each other. All data processing is done within the VaST working directory which contains all the necessary binaries and scripts, so the modules can locate each other without the need to set environment variables. This also allows for an easy “unpack and compile” installation procedure described in Appendix A.1.

The input to VaST may be either a set of images (for a practical example see Appendix A.2) or a set of lightcurves constructed using other software or imported from an archive.
whose measurements were severely corrupted by some rare instrumental effect like an object’s image falling on a dead pixel. If the instrumental effect is not rare, e.g. if bad pixels are abundant and affect measurements of many stars, the affected stars will not stand out as “more variable” compared to the other stars in the dataset. It is ultimately up to a human expert to investigate lightcurves and images of the candidate variables and judge if the measurements of a particular object are trustworthy or have obvious problems (bad pixels, blending, etc) making them unreliable.

There is a number of ways to characterize a “degree of variability” in a lightcurve. We call the variability indices all the values that:

- Quantify the scatter of brightness measurements: \( \sigma \), MAD, IQR, etc. (Ferreira Lopes and Cross 2017, Sokolovsky et al. 2017).
- Quantify the smoothness of the lightcurve like the J index (Welch and Stetson 1993) or \( 1/\eta \) (von Neumann 1941).
- Sensitive to both smoothness and scatter (Stetson’s J index), maybe also taking into account the shape of the measured brightness distribution (e.g. L index Stetson 1996, Ferreira Lopes and Cross 2016).
- Characterize the strength of a periodic signal in the lightcurve (e.g. Fruth et al. 2012, Sokolovsky et al. 2016).

In the absence of detailed information about measurement errors and instrumental effects one should consider the above parameters computed for a given lightcurve in the context of other lightcurves in this dataset to decide if a given variability index value corresponds to a variable object or not. The indices are not equally sensitive to all types of variability. Some of them are more susceptible to individual outliers (corrupted measurements) than the others. Scatter-based indices are generally sensitive to variability of any kind while the indices that characterize smoothness are more sensitive than scatter-based indices to objects that vary slowly compared to the typical observing cadence. That comes at the cost of these indices being insensitive to variability on timescales shorter than the typical observing cadence. Table 1 presents the list of variability indices computed by VaST. We refer to Sokolovsky et al. (2017) for a detailed discussion and comparison of these indices.

In the following Section we present a detailed description of all the processing steps from reading input images to constructing lightcurves, computing and visualizing the variability indices. Appendix A gives practical examples of using the code.

3. Data processing flow

The primary input for VaST is a series of FITS images. The images are expected to be taken with the same telescope-camera-filter combination and cover the same area of the sky. The images may be shifted, rotated and reflected with respect to each other, but should overlap by at least 40 per cent with the sky area covered by the first image. The first image supplied to VaST serves as the astrometric and photometric reference. If the input images vary in quality, it is up to the user to select a good reference image and put it first on the command line (Appendix A.3). Below we describe the main processing steps performed to extract lightcurves from a set of images.

3.1. Reading metadata from the FITS header

Information about the observing time, image dimensions and CCD gain information (that is needed to accurately compute contribution of the Poisson/photon noise to the total photometric error), is extracted from FITS header of each input image. The observing time assigned to each image is the middle of the exposure. The exposure start time is derived from one or two (if date and time are given separately) keywords from the following list: DATE-OBS, TIME-OBS, EXPPSTART, UT-START, START. The exposure time in seconds is extracted from EXPOSURE or EXPTIME. If none of the two keywords are present in the header, the difference between the observation start and its midpoint is assumed to be negligible (the exposure time is set to

---

**Table 1: Variability indices computed by VaST.**

| Index Reference |
|------------------|
| Weighted std. deviation – \( \sigma \) | Kolesnikova et al. (2008) |
| Clipped \( \sigma \) \( \rightarrow \) \( \sigma_{\text{clip}} \) | Kolesnikova et al. (2008) |
| Median abs. deviation – MAD | Zhang et al. (2016) |
| Interquartile range – IQR | Sokolovsky et al. (2017) |
| Reduced \( \chi^2 \) statistic – \( \chi^2_{\text{red}} \) | de Diego (2010) |
| Robust median stat. – RoMS | Rose and Hintz (2007) |
| Norm. excess variance – \( \sigma^2_{\text{NXS}} \) | Nandra et al. (1997) |
| Norm. peak-to-peak amp. – \( \nu \) | Sokolovsky et al. (2009) |

---

**Indices quantifying lightcurve smoothness**

- Autocorrelation – \( l_1 \)
- Welch–Stetson index – \( I_{WS} \)
- Flux-independent index – \( I_0 \)
- Stetson’s J index
- Time-weighted Stetson’s \( J_{\text{time}} \)
- Clipped Stetson’s \( J_{\text{clip}} \)
- Stetson’s L index
- Time-weighted Stetson’s \( L_{\text{time}} \)
- Clipped Stetson’s \( L_{\text{clip}} \)

**Indices quantifying magnitude distribution shape**

- Stetson’s K index
- Kurtosis
- Skewness

---

<http://fits.gsfc.nasa.gov/>
0). The derived time is converted to Julian date (JD).

If no keyword describing the exposure start time is found, but the JD keyword is present in the header, the keyword is assumed to correspond to the middle of exposure. This simplified convention for recording the observation time in the FITS header has been introduced within the program of digitizing photographic plates of Sternberg Astronomical Institute’s collection (Kolesnikova et al. 2008, 2010, Sokolovsky et al. 2014a). This is different from the definition of the JD keyword adopted by the widely-used CCD camera control software MaxIm DL. However, MaxIm DL will always write the DATE-OB keyword which, if present, is used by VAST instead of JD. If VAST fails to recognize the format in which the observing time is specified in the FITS header of your images, please send an example image to the authors.

VAST supports the following two time systems: Coordinated Universal Time (UTC) and Terrestrial Time (TT). The Earth rotation period is a few milliseconds longer than 86400 SI sec and is changing irregularly when measured at this level of accuracy. To keep UTC timescale in sync with the Earth rotation, a leap second is introduced every few years. The UTC time, being the basis of civil time, is readily available in practice through GPS receivers, NTP servers and radio broadcast, so the time of most images is recorded in UTC. The observing time from some instruments, notably Swift/UVOT, is recorded in TT. The advantage of TT is that this timescale is continuous, not interrupted by leap seconds. At each moment of time, the two timescales are related as

\[ TT = UTC + (TAI – UTC) + 32.184 \text{ sec} \]

where (TAI – UTC) is the difference between the International Atomic Time (TAI) and UTC for a given date.\(^2\) As of spring 2017, the difference between TT and UTC is 69.184 sec.

The program tries to determine the time system in which the observing time is expressed in the FITS header by searching for the TIMESYS key or parsing the comment field for the keyword used to get the time. If neither UTC nor TT are explicitly mentioned in the header, the input time is assumed to be UTC. The observing time in the lightcurves created by VAST is by default expressed in terrestrial time – JD(TT). The conversion from UTC to TT may be disabled (Appendix B).

No heliocentric correction is applied to the lightcurves by default, as by design of the program, it is possible to produce lightcurves without knowing the absolute celestial coordinates of the imaged objects. If the coordinates are known, the correction may be applied after constructing a lightcurve as described in Appendix A.16 See Eastman et al. (2010) for a detailed discussion of accurate timing problems in the context of optical photometric observations.

3.2. Source extraction and photometry

VAST relies on SExtractor for detecting sources, measuring their positions (in pixel coordinates) and brightness. The source extraction parameters are set in the SExtractor configuration file default.sex located in the VAST directory. Depending on the command-line settings (Appendix B), more than one run of SExtractor may be needed for each image:

option I) If all images are to be measured with a circular aperture of the fixed diameter, only a single SExtractor run is performed for each image.

option II) If the circular aperture diameter is allowed to vary between images to account for seeing changes (e.g. Collins et al. 2017), a preliminary SExtractor run is performed to determine the typical source size. For each source the maximum spatial r.m.s. dispersion of its profile is computed (SExtractor output parameter \(A_{IMAGE}\)). The aperture diameter for each image is computed as \(C \times \text{median}(A_{IMAGE})\), where the median is computed over all the detected sources passing selection criteria and the default value of \(C = 6.0\).

option III) If the program is running in the PSF-fitting photometry mode, multiple SExtractor passes over the input image are required. First, a preliminary run is performed to determine seeing, as in the option II) above. Then SExtractor is run with conservative detection limits to extract bright sources and save their small images (SExtractor output parameter VIGNET). These small images are used by PSFEx\(^2\) (Bertin 2011) to construct a model of spatially-variable point spread function (PSF). The size of these small images as well as limits on size of objects accepted for PSF reconstruction are determined based on seeing. The PSF reconstruction step is followed by PSF-fitting with SExtractor. Objects that provide a bad fit (non-stellar, heavily blended, corrupted or overexposed objects) are discarded from the output catalog. Finally, an aperture photometry run is conducted with SExtractor (again, same as in option II). This step is performed for technical reasons: a catalog that includes all the objects, even the overexposed ones, is needed at later processing steps to perform absolute astrometric calibration discussed in Section 3.3.

The input images are expected to be well-focused. VAST may be used to process strongly-defocused images with “donut-shaped stars” given a careful selection of the processing parameters including source detection and deblending thresholds (Appendix A.8) and manually specified aperture size and star matching radius (Section 3.3 Appendix B).

The list of sources detected by SExtractor may be filtered to remove saturated, blended and extended sources as their brightness measurements are less accurate and such sources may appear as spurious candidate variables later in the analysis. Saturated sources are removed based on the source detection flag set by SExtractor. The filtering of blended and extended sources may be accomplished using SExtractor flags (if SExtractor deblending was successful) and by identifying outliers in the magnitude-size plot (Figure 1). The sources that were not properly deblended will appear larger than the other sources of similar brightness. Rejection of blended sources is done using individual images rather than using information about source size and detection flags collected from the full series of images because a generally “good” source may appear blended...
the algorithm differ in how the lists of triangles are constructed from the input star lists, how the triangles are matched to find the initial guess of the coordinate frame transformation and how this initial transformation is further refined.

Our cross-matching algorithm goes through the following steps for the lists of sources (assumed to be mostly stars) derived from each image:

1. Sort the detected stars in magnitude.
2. Select $N$ brightest stars having acceptable SE\textsc{extractor} flags as the reference. The value of $N$ is discussed below.
3. Construct a set of triangles from the selected $N$ stars using the following two algorithms together:
   
   (a) For each reference star find the two nearest reference stars to form a triangle.
   
   (b) For each reference star having the index $n$ in the magnitude-sorted list, form triangles from stars having indices: $(n, n+1, n+2), (n, n+1, n+3), (n, n+1, n+4), (n, n+1, n+5), (n, n+2, n+3), (n, n+2, n+4), (n, n+2, n+5), (n, n+3, n+4), (n, n+3, n+5)$ and $(n, n+4, n+5)$.

In practice, a combination of these two algorithms allows one to obtain overlapping lists of triangles for virtually any pair of reference star lists if these lists overlap.

4. Similar triangles are identified in the lists by comparing ratios of the two smaller sides to the larger side of each triangle in the first list (corresponding to the reference image) to these ratios for triangles in the second list (corresponding to the current image).

5. Accept only the similar triangles that have approximately the same scale to improve stability of the algorithm.

6. For each pair of similar triangles construct a linear transformation between the pixel coordinates at the current frame and pixel coordinates at the reference frame. The transformations are applied to the $N$ reference stars.

7. The transformation that allowed \textsc{vaST} to match the highest number of reference stars is now applied to all the detected stars on the frame. The residuals along the two axes $(\text{d}X, \text{d}Y)$ between the predicted and measured positions of each matched star are recorded.

8. The residuals $\text{d}X$ and $\text{d}Y$ are approximated as linear function of coordinates $(X,Y)$ on the reference frame by least-square fitting planes into the 3D datasets $(X,Y,\text{d}X)$ and $(X,Y,\text{d}Y)$. The derived least-square coefficients are used to correct the coordinates of all sources computed from the initial linear transformation determined from a pair of matched triangles. This correction is necessary for large images matched using a pair of small triangles (likely constructed by the algorithm (a) above). Since the positions of stars forming the triangle are measured with a limited accuracy, the initial linear transformation will be inaccurate for stars far away from the triangle.

9. Use the corrected positions to match stars to the ones detected on the reference image. A custom spatial indexing scheme is used to avoid the computationally expansive step of calculating distances from each star in the current

![Figure 1: Semi-major axis length (SE\textsc{extractor} output parameter $A_{\text{IMAGE}}$) as a function of instrumental magnitude for sources detected on a CCD frame. Crosses mark sources rejected as being too large for their magnitude (blended or extended).](image-url)
image list to each star in the reference list (see the discussion by e.g. 
Riccio et al. 2017). The reference list used in this last step is complemented by stars detected on previously matched images even if they were not detected (or rejected because of unacceptable flag values) on the reference image. Only the stars originally detected on the reference image are used in the reference list to construct triangles.

The main input parameter of this algorithm is the number of reference stars $N$. VaST is trying to match images probing various values of $N$ in the range 100–3000. The lower value is sufficient to match most narrow-field images while the upper value is set by the requirement to perform the match in a reasonably short time. The secondary input parameter is the matching radius by the requirement to perform the match in a reasonably short time. The above image matching algorithm assumes a linear transformation between images. This assumption may be approximately correct even for highly-distorted wide-field images as long as the relative shifts between the distorted images are small. For a successful match, the coordinate transformation error at image edges needs to be smaller than the matching radius and typical distance between stars (to avoid confusion). The requirements for star matching are less strict than they would be for image stacking (that demands sub-pixel coordinates transformation accuracy). In practice, the matching algorithm works well for sky images obtained with lenses having a focal length of 100 mm or longer.

### 3.4. Cross-calibration of instrumental magnitudes

VaST uses a large number of matched stars to reconstruct the scale offset as a function of magnitude. This allows VaST to calibrate magnitude scales of images obtained with non-linear image detectors listed in Section [1]. The only requirement is that the current and reference image have a sufficient number of common stars (≥ 100). The dependency of the magnitudes in the instrumental scale of the reference image, $m_{ref}$, on the instrumental magnitudes of the current image, $m_{inst}$ (the calibration curve), is reconstructed using all the stars matched between the two images and approximated by one of the three functions selected by the user:

1. linear function
   $$m_{ref} = a_1 m_{inst} + a_0$$  
   (1)

2. parabola
   $$m_{ref} = a_2 m_{inst}^2 + a_1 m_{inst} + a_0$$  
   (2)

3. “photocurve” function proposed by Bacher et al. 2005:
   $$m_{ref} = a_0 \log_{10}(10^{a_1 m_{inst}} - 1) + a_3$$  
   (3)
   or the inverse function of it
   $$m_{ref} = \frac{1}{a_1} \log_{10}(10^{\frac{a_0}{a_1}} - 1) + a_2$$  
   (4)

depending on which of the two functions provides a better fit to the data.

Here $a_0, a_1, a_2, a_3$ are the free parameters of the fit. Fitting is performed using the linear least-squares for the first two functions while the Levenberg-Marquardt algorithm is used to perform non-linear least-squares fitting of the photocurve. The data points are weighted according to the inverse square of their estimated photometric errors. An iterative clipping procedure is applied to discard variable, poorly measured or misidentified objects. An example magnitude scale calibration between two photographic images is presented on the left panel of Figure [2].

VaST can attempt to minimize the effect of the difference in extinction across the image (e.g. Irwin et al. 2007) by least-squares fitting a plane in the 3D dataset $(X, Y, dm)$, where $(X, Y)$ are the image coordinates and $dm$ is the residual difference in magnitudes measured for a given object on the reference and the current image after applying the calibration curve. The best-fit plane is then subtracted from the magnitudes measured on the current image, thus correcting for the linear (in magnitude) term of the extinction, regardless of image orientation. This correction is similar to the coordinates correction performed after applying the initial transformation derived from matched triangles (Section [3.3]). However, performing this correction may do more harm than good if the plane cannot be fitted with sufficient accuracy. The correction may be turned on or off by the user (Appendix B). By default, the correction is applied to images having >10000 detected sources. It is assumed that this large number of sources will be sufficient to accurately fit the plane. Note that this plane-fitting correction does not correct for the extinction difference, but rather for the difference in extinction difference between the reference and the current image. Photometric calibration described in Section [3.8] assumes
one zero-point for the whole field, resulting in offsets between zero-points of lightcurves of objects visible in the upper and lower (with respect to the horizon) parts of the reference image. At this stage VaST does not take into account the color term in extinction correction (differential color extinction) as the colors of the observed sources are, in general, unknown. After constructing lightcurves you may fit for the differential color extinction together with other systematic effects affecting multiple sources in the field by applying the SysRem algorithm as described in Appendix A.11.

3.5. Output lightcurves, statistics and the graphical interface

The lightcurves of all the objects are saved in individual ASCII files named outNNNNN.dat where NNNNN is the source number. Each line of the lightcurve file contains:

- The middle-of-exposure JD in TT or UTC (Section 3.1).
- Magnitude in the instrumental system corresponding to the reference image.
- Magnitude error estimated by SExtractor. The estimation includes contributions from the background noise over the aperture area and photon noise. The photon noise is estimated correctly only if the GAIN parameter is set correctly from the FITS header (Section 3.1) or by the user in the default.sex file.
- Pixel coordinates of the object on the current image reported by SExtractor (parameters XWIN_IMAGE and YWIN_IMAGE).
- Aperture size in pixels used for this image.
- Path to the image file from which the above measurements of the object’s parameters were obtained.

For each object VaST computes a number of variability indices characterizing the scatter of magnitude measurements and/or smoothness of the lightcurve (Sokolovsky et al. 2017). The indices are constructed so that variable stars tend to have larger values of the indices compared to the majority of non-variable stars, but the index values (and their scatter) expected for non-variable stars are strong functions of magnitude, so the simple cut in index values is usually not sufficient for efficient selection of variable stars. The index values for each object are stored in vast_lightcurve_statistics.log its format being detailed in the accompanying file vast_lightcurve_statistics_format.log. The index values may be utilized by the user as an input for automated selection of candidate variables using external software implementing complex cuts in index values or machine learning.

The simplest way of using VaST is to visualize the variability indices vs. magnitude plots using its GUI (Figure 5). By clicking on an object in these plots a user may visualize its lightcurve and clicking on a point in the lightcurve plot – display an image corresponding to this point. This way it is possible to select objects displaying high variability index values and make sure the apparent variability is not caused by brightness measurement problems resulting from blending with nearby objects or CCD cosmetic defects (the problems that readily appear on visual inspection of the object’s images). With one keystroke a user may launch the star identification script described in Section 3.7 or...
send its lightcurve to the online period search tool.\footnote{http://scan.sai.msu.ru/lk/}

The VAST GUI is based on the PGPLOT library\footnote{http://www.astro.caltech.edu/~tjp/pgplot/} which is well suited for displaying and editing data and image plots. It is exceptionally easy to use for a developer. The downside is that the resulting interface is counterintuitive to a contemporary user as it has no buttons, just the clickable plots. Whenever a user has a choice between multiple actions, instead of clicking a button to execute the desired action, a user has to press a key on a keyboard. For each GUI window, a list of possible keyboard keys is printed in a terminal window.

### 3.6. Processing time

VAST running time is mostly limited by the image processing time with SExtractor. On a 2.20 GHz quad core Intel Core i7 laptop it takes about 2 minutes to process 784 images each containing about 250 stars (the dataset described in Appendix D) in the aperture photometry mode. Processing the same images in the PSF-fitting mode takes about 50 minutes using the same hardware.

### 3.7. Astrometric calibration

It is possible to construct lightcurves in the instrumental magnitude scale and search for variability with VAST without finding the transformation between the reference image and celestial coordinates. If the field center is known to the user, detected variable stars can be identified by visually comparing the displayed image with a star atlas like Aladin\footnote{http://aladin.u-strasbg.fr/} (Bonnarel et al. 2000), see also Appendix A.6. However, this approach is practical only for narrow-field image sets containing only few variable objects.
For larger fields it is more practical to find a plate solution in
an automated way (see the usage example in Appendix A.7).
The script will use the Astrometry.net software to plate-solve
the reference image and identify the star with USNO-B1.0
(Monet et al. 2003) for positional reference and with databases
listing variable stars: GCVS (Samus et al. 2009), VSX
(Watson 2006) and SIMBAD (Wenger et al. 2000). The cata-
logs are accessed through Vizier using vizquery or directly
through a web interface using CURL. The Astrometry.net soft-
ware may be run by the script locally (if installed in the system)
or on a remote server. In the latter case source extraction
is done locally and only the list of detected stars (not the full im-
age) is uploaded to the server to save bandwidth. The FITS
header containing the WCS information created by Astrome-
try.net is merged into the original FITS image and the approx-
imate equatorial coordinates of all sources are measured with
the additional run of SExtractor.

In practice, the accuracy of the coordinates derived this way
may not be sufficient for unambiguous object identification for
the following reasons. While finding a blind plate solution
Astrometry.net may fit for image distortions and store them in
the FITS header following the SIP convention (Shupe et al.
2005). This convention is not understood by SExtractor
which is using the PV convention to represent geometric dis-
tortions (Shupe et al. 2012). As a workaround, VaST employs
xy2sky routine from WCSTools (Mink 1997, 2014) to convert
SExtractor-derived pixel coordinates to celestial coordinates
rather than rely on SExtractor to do this conversion.

The following two problems are specific for large-format
photographic plates digitized with a flatbed scanner. If the scan-
er’s image detector (scanning ruler) is smaller than the plate
width, the detector has to make multiple passes along the plate
and the image strips resulting from each pass have to be stitched
together (Figure 4). Such a stitch results in a discontinuity in
image to celestial coordinates conversion (Figure 5 right panel)
that cannot be adequately represented with a low-order poly-
nomial description of distortions. The other problem is the char-
acteristic “hacksaw” distortions pattern (Vicente et al. 2007)
resulting from non-uniform mechanical movement of the scan-
ning ruler (see Figure 5 left panel). While the star matching
algorithm proposed by Heyl (2013) is capable of dealing with
shearing, we are not aware of an algorithm that could accom-
modate a shift or a discontinuity in coordinates transformation.

To minimize the negative effects of the above, VaST relies
on the assumptions that image distortions are similar for ob-
jects that are close to each other and the distortions are suffi-
ciently small to allow for correct identification for the majority
of objects. After attempting to match all the detected objects
with the UCAC4 catalog (Zacharias et al. 2013, accessed from
VizieR) using the uncorrected coordinates, for each detected
object VaST computes the mean difference between the mea-
sured and the catalog positions of matched objects within a cer-
tain radius of the current object. This difference is used as a
local astrometric correction. A range of local correction radii
is tested for each object and the one resulting in the smallest
scatter of the measured-to-catalog distances is used to compute
the final correction. The corrected positions of all the detected
objects are used to match them with UCAC4 again in an at-
tempt to find new matches. This procedure is repeated itera-
tively until the new iteration does not result increased number
of matched stars or the maximum number of iterations is reached.
The matching radius is set based on the image field of view.
Figure 5 illustrates how the complex distortions introduced by
a flatbed scanner can be corrected by applying the described
procedure. The systematic effects are removed at the expense
of slightly increased random errors in astrometry.

![Figure 5: Differences between the cataloged and measured positions of stars on a digitized photographic plate. Distortion patterns commonly introduced by flatbed scanners are illustrated: the stitch between two passes of the scanning ruler (left; red bars indicate approximate boundaries of the stitch area, cf. Figure 3) and hacksaw distortions introduced by non-uniform motion of the ruler (right). These distortions are mitigated by the local astrometric correction.](image-url)
4. Searching for transients with V\textsc{extractor}

Searching for transients with \textsc{Vextractor} formed by SE amplitude variable star, a new object appearing well above the detection limit results in an obvious change in an image of a nearby star. This can be done by interactively specifying a comparison star with a known magnitude as described in Appendix A.6. If the reference image has a sufficiently large field of view to be blindly solved with \textsc{Astrometry.net}, the magnitude scale can be calibrated using APASS (Henden et al. 2016; Henden and Munari 2014) magnitudes as described in Appendix A.7. The following filters are supported: B, V, R, I, r and i. The R and I magnitudes are not present in APASS, so they are computed from r and i magnitudes following Jester et al. (2005):

\[ R = V - 1.09(r - i) - 0.22 \]  
\[ I = R - 1.00(r - i) + 0.21 \]

The candidate transients are selected as objects that were not detected on the second-epoch images - it would be incorrectly categorized as a flare if this object is masked by the host galaxy or it should be successfully deblended from the host galaxy or it should be sufficiently bright to cause a detectable change in the measured brightness of the galaxy+supernova compared to the measured brightness of the galaxy alone. However, the image subtraction may be difficult to implement if the PSF and its variations across the images are difficult to reconstruct. In such cases transient detection based on comparison of the lists of sources detected on first- and second-epoch images may still be preferable.

The transient detection was implemented in \textsc{Vast} for processing the New Milky Way (NMW\footnote{http://scan.sai.msu.ru/nmw/}) nova patrol (Sokolovsky et al. 2014b) images. Unlike the main lightcurve-based variability search mode that is fairly generic, \textsc{Vast}'s transient detection mode is applicable only to wide-field relatively shallow images (as it relies on Tycho-2 catalog for magnitude calibration; Høg et al. 2000) and tied to the specific observing strategy. For each transient survey field, exactly four images are required as the input: tow first-epoch (reference) images and two second-epoch images. The two images at each epoch should be obtained with a sufficiently large shift (20 pixels or more) to suppress spurious detections due to image artifacts. Two first-epoch images are needed to reduce the probability of an object visible on the reference image not being detected by \textsc{SExtractor} due to blending or an image artifact. If this object is detected on the second-epoch images - it would be incorrectly identified as new.

The candidate transients are selected as objects that were not detected at any of the reference images, or where at least 1 mag. fainter compared to the second-epoch images. The second criterion is needed to identify flaring objects and new objects that are blended with previously-visible ones.

\textsc{Vast} generates an HTML report containing a list of candidate transients and opens it in a web browser. For each candidate transient the HTML report displays the following information (Figure 7):
Cutouts from the first and second epoch images centered on the transient candidate.

Photometry and astrometry of the candidate.

Results of VSX search for known variable stars around the transient’s position. A local copy of the catalog is used.

Results of astcheck search for known asteroids around the transient’s position; astcheck relies on a local copy of asteroid orbit data.

Links to search the transient’s position in SIMBAD and VizieR databases as well as NSVS, ASAS (Pojmanski 1997, 2002), and CSS (Drake et al. 2005) photometry archives. A link to the WISE (Wright et al. 2010) image atlas is also provided. It is especially useful for distinguishing unknown red variables (bright in the infrared light) from nova/cataclysmic variable candidates. The object’s position may also be checked in MPChecker (that has the latest information on asteroids and comets) and the NMW image archive.

Appendix A.15 provides an example of running VaST in the transient detection mode.

5. Remarks on development process

Some of the best practices for scientific computing were reviewed by Wilson et al. (2012). Here we highlight a few procedures that were found especially useful for VaST development.

The core functionality of VaST is implemented in C. Valgrind (Nethercote and Seward 2007) tools are used for profiling (Nethercote et al. 2006) and detecting memory errors and leaks (Seward and Nethercote 2005). Parallel processing is implemented using OpenMP.

Defensive programming style is adopted whenever possible. VaST continues execution after failing to process an individual image. With this we are trying to avoid the situation when hours of computation are lost because of a bad image mixed into a generally good input dataset (e.g. Reid et al. 2001). An attempt is made to print a meaningful error message in cases when the execution of the program cannot continue: most such situations are caused by incorrect input on the command-line or from image/lightcurve files and can be corrected by the user.

Automated testing was not implemented from the start of the project, but it quickly appeared that adding new functionality to the code broke some of the rarely-used functionality added earlier. The solution was found in system testing implemented in a form of a BASH script. The script runs VaST in a non-interactive mode on various sets of test data and checks if the output is consistent with the expectations. After VaST pro-
cessed a set of test images, the script checks if all middle-of-
exposure JDs were computed correctly, if all the images were
successfully matched to the reference image, if the reference
image was plate-solved and if the object having the highest
lightcurve scatter can be identified with a known variable star
or if a known flaring or moving object is among the list of
detected transients. While the tests check high-level functionality
of the software, once a problem is discovered it is often easy to
pin-point a recently modified piece of code that caused it. Bug
reports from VaST users provide a steady source of non-trivial
test cases.

VaST is developed and tested under Gentoo Linux that, in
the authors’ view, provides a developer-friendly environment.
A set of portability tests is performed prior to release. The tests
ensure that VaST compiles and runs well on the latest Ubuntu
and Scientific Linux, as well as the old Scientific Linux 5.6 rec-
commended for building portable applications by the BOINC
project (Korpela 2012). The testing is also performed on the
latest stable release of FreeBSD. All these operating systems
are run as virtual machines through VirtualBox. A number of
bizarre differences in behavior between versions of gcc, make,
BASH, awk and sort commands supplied with different Linux
distributions (not to mention differences between their version
supplied with Linux and FreeBSD) were encountered, confirming
the need for the portability testing. We emphasize that the
operating system versions mentioned above are not the ones
strictly required for running VaST; rather they should be represen-
tative examples of the current UNIX-like systems diversity.
The goal of our portability testing is to have a reasonable ex-
pectation that VaST will compile and run on any contemporary
Linux or FreeBSD system.

To make VaST installation as easy as running the command
make, the package includes copies of some non-standard lib-
raries it relies on that are automatically built from source code
before compiling VaST source files. VaST also comes with its own copy of SExtractor however it is configured with
--disable-model-fitting to avoid dependency on the AT-
LAS library and therefore cannot perform PSF-fitting. To en-
able PSF-fitting photometry one has to install ATLAS, PSFEx
and SExtractor system-wide. VaST will use the system instal-
lation of SExtractor if it finds one.

The main obstacle in VaST adaptation by the users appears to
be the complexity of its command-line interface combined with
the lack of clear documentation. While software with overly
complex user interfaces and non-trivial installation procedures
(think AIPS, ESO-MIDAS, IRAF) are often tolerated in astronomy
for historical reasons, many variable star observers expect
software in this field to have an intuitively-understandable GUI.
The problem is not confined to the interface of the program,
but also includes the operating system interface. Familiarity
of potential users with POSIX-like command-line interface is
increasing thanks to the rise in popularity of Mac OS. VaST
documentation should be improved, in particular video tutori-
als seem to be a good way to introduce the software to potential
users. We are also considering a radical re-design of the user
interface to make it fully web-based.

6. VaST applications

The development of VaST was mostly driven by the au-
thor’s data processing needs. These evolved from a pig-
gyback variability search in targeted CCD observations of
known variable objects (Antipin et al. 2005, Sokolovsky et al.
2011) to variability search using digitized photographic plates
(Kolesnikova et al. 2008, Sokolovsky et al. 2014c). Swift/U-
VOT data analysis (Sokolovsky 2009, Gupta et al. 2012), visu-
alization of CoRoT lightcurves (Sokolovsky et al. 2010), pho-
tometry of individual active galactic nuclei (Schinzel et al.
2011, Sokolovsky et al. 2014d) and wide-field search for bright
optical transients (Sokolovsky et al. 2014b) that resulted in the
discovery of Nova Sagittarii 2012 #1 (ν=V5589 Sgr; Korostiv et
al. 2012, Sokolovsky et al. 2013). VaST was appli-
cd for variability search in CCD data by others including
Maiaess et al. (2013), Lapuhkin et al. (2014), Kryachko et
al. (2015), Pashchenko et al. (2017) used VaST to compute mul-
tiple variability indexes needed to test a machine learning-based
approach to variable star detection. From the list of VaST-
related publications maintained at the code’s homepage, it is
estimated that VaST contributed to the discovery of at least
1800 variable stars.

7. Summary

VaST takes a series of sky images as an input and produces
lightcurves for all imaged objects. A set of variability indices
is computed that characterize scatter and smoothness of the
lightcurves and allow the user to distinguish variable objects
from non-variable ones. The user interface aids in visual in-
spection of candidate variables by visualizing variability index–
magnitude plots, lightcurves and images associated with each
lightcurve point. Thanks to parallel processing and smart mem-
ory management, the code is able to process thousands of im-
ages while running on modest hardware. VaST is free software,
distributed under the terms of the GNU General Public License
(GPL). We hope that the description of the code provided here
will be useful both to VaST users and those who aim to develop
the next generation variability search software.

Acknowledgments

We thank the anonymous referees, Dr. Alceste Bonanos,
George Kakeletris, Maria Mogilen, Dean Roberts, Dr. Niko-
lay Samus, Olga Sokolovskaya, Dr. Alexandra Zubareva and

---

\[\text{https://boinc.berkeley.edu/trac/wiki/VaSTCompatibility}^{39}\]

\[\text{\texttt{gawk}} \text{ which is the default awk in Ubuntu} \text{ will not understand}
\text{\texttt{"%lf"}} \text{ in the printf formatting string (accepting only \texttt{"%f"}):}
\text{echo 1.23 | awk \{'printf \"%lf\n\",$1\'}
\text{while it is perfectly fine for \texttt{awk} which is the default in most Linux distribu-
tions as well as BSD \texttt{awk}. This kind of differences are hard to anticipate, so}
\text{they need to be tested for.}

\[\text{\texttt{http://scan.sai.msu.ru/vast/\#public}}^{40}\]

\[\text{\texttt{https://www.gnu.org/licenses/gpl.html}}^{41}\]
Dmitry Litvinov for critically reading the manuscript. Thanks to Mark Vinogradov for the suggestion on improving the text. We thank Dmitry Nasonov who designed the VaST homepage. Sergei Nazarov (Moscow) who coined the name for the code and Sergei Nazarov (CrAO) who created great instructions on variability search (in Russian), Stanislav Korotkiy for starting the NMW nova patrol project, Dr. Alexei Alakoz, Dr. Panagiotis Gavras and Dr. Jean-Baptiste Marquette for testing the code on Mac OS. KS thanks Dr. Sergei Antipin and Dr. Vladimir Amirkhanyan for illuminating discussions of photometric data analysis, Daria Kolesnikova and Andrey Samokhvalov for many suggestions that helped to improve the output and interface of the program, Dr. Richard White for the illuminating discussion of flag images and weight maps handling by SEXTRACTOR. This work is partly based on observations made with the 2.3 m Aristarchos telescope, Helmos Observatory, Greece, which is operated by the Institute for Astronomy, Astrophysics, Space Applications and Remote Sensing of the National Observatory of Athens, Greece. This research has made use of the SIMBAD database, operated at CDS, Strasbourg, France. This research has made use of the VizieR catalog access tool, CDS, Strasbourg, France. The original description of the VizieR service is presented by Ochsenbein et al. (2000). This research has made use of the International Variable Star Index (VSX) database, operated at AAVSO, Cambridge, Massachusetts, USA. This research has made use of NASA’s Astrophysics Data System.

Appendix A. Use cases

This section provides installation instructions and practical examples of how common variability-search problems can be solved with VaST.

Appendix A.1. Compiling VaST

To run VaST you will need a computer running Linux, FreeBSD or Mac OS X operating system. If you have a different system – run one of the supported systems in a virtual machine. Make sure gcc compiler with C++ and Fortran support as well as header files needed to compile X.Org GUI applications are installed.

In a terminal window, download and unpack the archive containing VaST source code, change to the unpacked directory and compile the code:

```bash
wget \
|protect\vrule width0pt\protect\hhref{http://scan.sai.msu.ru/vast/vast-1.0rc78.tar.bz2}\n\n|tar \n\xrightarrow{rxf} vast-1.0rc78.tar.bz2\n\ncd vast-1.0rc78\n\nmake
```

The script may complain about missing external programs or header files. Install any missing components in your system and run `make` again. At this point you should have a working version of VaST ready to perform aperture photometry, as described in the following examples. All processing is done from within the VaST directory (vast-1.0rc78 in the above example) – no system-wide installation is needed.

Note that in this example f_72-058r.fit appears on the list of input images twice: the first time it is specified explicitly and the second time – when it satisfies the condition specified with the wildcard character *. VaST will recognize this as again the reference image and will not try to measure it twice.

Appendix A.2. Variability search in a series of CCD images

Suppose we have a series of dark-subtracted and flat-field-corrected images of a particular star field in the directory `./sample_data`. To construct lightcurves and perform variability search in these images, form the VaST directory run

```bash
/vast ./sample_data/*
```

This will create an interactive window displaying a variable index–magnitude plot and allowing to inspect individual lightcurves and images as described in Section 5.5.

Appendix A.3. Setting a reference image of your choice

VaST is using the first image specified on the command-line as the reference image. This may be a bad choice if the first image in a series is of poor quality. To specify a different image, just put it first on the command-line. In the example above, one may set `f_72-058r.fit` as the reference image

```bash
/vast ./sample_data/f_72-058r.fit ./sample_data/*
```

If one runs `./find_candidates` without a command-line argument, it will re-compute the variability indices instead of re-displaying the old computations. This is useful if the lightcurves were altered, e.g. after calibrating the magnitude scale (Appendix A.6) or applying SystRM algorithm (Appendix A.11).

All the lightcurves and log files produced after measuring a set of images may be saved to a directory `MY_FIELD_NAME`

```bash
util/save.sh MY_FIELD_NAME
```

This can be used to load these images back into the Vast working directory:

```bash
util/load.sh MY_FIELD_NAME
```

Appendix A.5. View a lightcurve of an individual star

The user may view a lightcurve of an object by selecting it on the reference image:

```bash
/select_star_on_reference_image
```

or if the object’s number (12345) is known from the previous VaST run, the lightcurve can be plotted with

```bash
/lc out12345.dat
```
You may zoom in to a section of the lightcurve by pressing ‘Z’ on the keyboard (press ‘Z’ twice to zoom out). You may send the entire lightcurve to the online period search tool by pressing ‘L’. If the lightcurve is generated by VaST and not imported from external software, you may click on each lightcurve point to view the image corresponding to this point. You may try to identify the object with a known variable by pressing ‘U’ (see Section 5.7). As with the other VaST GUI applications, look at the terminal to see the list of possible keyboard commands.

Appendix A.6. Manual single-star magnitude calibration

After generating lightcurves from a set of images (Appendix A.2, Appendix A.10 or Appendix A.12) run

util/magnitude_calibration.sh

The script will display an image marking the stars that pass all the quality cuts. Identify a reference star of your choice by clicking on it and enter its magnitude in the terminal. The APASS catalog visualized through Aladin is a good place to find reference stars (Section 3.3). An example of such single-star zero-point calibration is discussed in Appendix D.

Appendix A.7. Automated magnitude scale calibration

If the field of view is large enough to be automatically plate-solved, the magnitude calibration can be performed by automatically matching the detected stars to the APASS catalog:

util/magnitude_calibration.sh V

where the command-line argument specifies the observing band (see Section 3.3). An interactive plot displaying the APASS magnitude as a function of the instrumental magnitude (Figure 2) will be displayed. A fitting function may be chosen among the ones described in Section 3.4 by pressing ‘P’. Weighting may be turned on or off by pressing ‘W’. Outlier points may be interactively removed by pressing ‘C’ and drawing a rectangle with a mouse around the outliers. After a satisfactory fit is found, right-click to apply the calibration.

Appendix A.8. Fine-tuning source extraction parameters

To check how well star detection is performed on an image:

./sextract_single_image ../sample_data/f_72-001r.fit

where ../sample_data/f_72-001r.fit is the path to the image file. This will display the image marking all the detected objects with green circles. Normally, one would like to have all objects clearly visible on the image to be detected while a minimal number of obvious artifacts like hot pixels or noise peaks to be mistaken for real astronomical sources. The default source extraction parameters work well for most CCD images while the images acquired with DSLR cameras (typically equipped with CMOS detectors) and digitized photographic images typically require fine-tuning of the extraction parameters.

The extraction parameters are set in the default.sex file. The most relevant are DETECT_MINAREA and DETECT_THRESH. A detailed description of the parameters may be found in SExtractor documentation. The VaST directory includes a few example files named default.sex.

A click on a detected source will print its derived properties (including pixel position and instrumental magnitude) in the terminal. Inspection of object’s properties is useful to find out why a particular known object of interest did not pass the selection criteria and its lightcurve was not generated. It is useful to check SExtractor flags typically assigned to the detected objects. By default, VaST only accepts sources that have flag values 0 or 1. If the star field is very crowded or the instrument’s PSF has a funny shape, the majority of objects may be marked with flags 2 or 3. In that case you may either try to find more optimal extraction parameters by changing the detection threshold as described above as well as deblending parameters DEBLEND_NTHRESH and DEBLEND_MINCONT or run VaST with -x3 command-line argument in order not to filter out blended stars.

Appendix A.9. Selecting best aperture size for each source

VaST may use SExtractor to measure brightness of each source in multiple circular apertures. For each source all apertures have the same center and their sizes are 10 per cent smaller, equal, 10, 20 and 30 per cent larger than the size of the reference aperture selected automatically for each image based on seeing (as described in Sec. 3.2) or set to a fixed size by the user for the whole series of images (Appendix B). After processing all the images, VaST may select for each source the aperture that resulted in the smallest lightcurve scatter (quantified by MAD). The following command will activate the best aperture selection:

./vast --selectbestaperture ../sample_data/*

In practice, selecting aperture size individually for each object was found to result only in a minor improvement of photometric accuracy compared to the use of a properly-selected single-size aperture for all sources on a frame. This was previously reported by [Deeg and Doyle 2001], see also [Mighell 1999].

Appendix A.10. PSF-fitting photometry with PSFEx

In order to enable PSF-fitting you will need to install SExtractor (not disabling PSF-fitting support, see Section 5) and PSFEx (along with the libraries they depend on) system-wide. Then you may run

./vast -P ../sample_data/*

where ‘-P’ tells VaST to use PSF-fitting.

While the simple aperture photometry works well out-of-the-box for most CCD images, PSF-fitting photometry will require fine-tuning of PSF extraction parameters for each new telescope-camera combination. The PSF extraction parameters

https://www.astromatic.net/pubsvn/software/sextractor/trunk/doc/sextractor.pdf
are set in default.psfex and described in the PSFEx documentation. The two most relevant ones are PSFVAR, DEGREES and PSFVAR_JSWAP. A user should experiment with different values of these parameters and select the ones that minimize the lightcurve scatter for the bright stars in the field (that may be visualized with ./find_candidates, see Appendix A.2) for final processing. This may be a time-consuming process as each PSF-fitting photometry run needed to construct lightcurves and estimate their scatter is taken considerably longer than an aperture photometry run on the same images. Fortunately, it is sufficient to select the best PSF extraction parameters once for a given instrument.

The source catalogs generated from the input images are cleaned from detections that resulted in bad PSF-fit quality. This reduces the number of false detections due to cosmic ray hits and hot pixels as well as objects that could not be properly deblended and hence have their photometry corrupted.

For faint stars PSF-fitting photometry results in considerably smaller lightcurve scatter compared to the fixed-aperture photometry. However, the accuracy of PSF-fitting photometry of bright stars is actually lower than that of aperture photometry. The reason is that for the bright stars the dominating source of errors are the residual uncertainties in reconstructing spatial and temporal variations of the PSF rather than background noise (as is the case of faint stars). The quality of PSF-fitting lightcurves can often be considerably improved by applying a few iterations of the SysRem procedure discussed in Appendix A.11.

Appendix A.11. Improving photometric accuracy with SysRem

The SysRem algorithm proposed by Tamuz et al. (2005) attempts to remove linear effects that, to a various degree, affect many lightcurves in a set. The original paper explains the algorithm using differential extinction as an example, but actually the algorithm is applicable to systematic effects of any physical origin, as long as many stars are affected by it.

After constructing lightcurves in the usual way (Appendix A.2) run
util/sysrem2
then use
./find_candidates
to re-compute the variability indices and inspect the results. Repeat the procedure until there is no further improvement in lightcurve scatter. Each util/sysrem2 removes no more than one systematic effect (while a real physical effect may be modeled as multiple linear effects removed by multiple SysRem iterations). Typically 3–6 SysRem iterations are sufficient to considerably improve quality of CCD lightcurves. One should avoid unnecessary SysRem iterations, as after all the detectable systematic effects are cleaned from the dataset, individual variable objects may start to dominate the SysRem solution and their real variability may be erroneously removed (Roberts et al. 2013, e.g.).

Appendix A.12. Variability search with photographic plates

When dealing with digitized photographic images, they typically first have to be converted from TIFF (the format commonly produced by scanner software) to FITS. The conversion can be performed with the tiff2fits tool:

```
/tiff2fits -i input.tif output.fits
```

where the -i argument indicates that a positive (white stars on black sky) should be produced. Information about the observing time should be added to the FITS header:

```
util/modhead output.fits JD 2442303.54
```

After producing a series of images they may be processed with VaST (note that a default.sex with customized detection parameters is needed)

```
cp default.sex.beta_Cas_photoplates default.sex
/vast -o -j ..//photographic.fits_images/*
```

Here default.sex.beta_Cas_photoplates is the example SEXTRACTOR parameters file for photographic plates, -o enables the photcurve magnitude calibration function described in Section 3.3. -j enables correction for the linear magnitude trend across the image (Section 3.4).

Appendix A.13. Identification of a variable object

If the image field of view is sufficiently large to be automatically plate-solved with ASTROMETRY.NET you may run the automatic star identification by pressing 'U' key in the lightcurve inspection window or typing in a free terminal

```
util/identify.sh out12345.dat
```

where out12345.dat is the file containing VaST-format lightcurve of the object you are interested in. This will run a series of scripts that will plate-solve the first image at which the object is detected, determine its equatorial coordinates and attempt to match it external catalogs as described in Section 3.7.

The ability to plate-solve narrow-field images is limited by the index files available to ASTROMETRY.NET code and processing time. At the time of writing, the plate-solve servers communicating with VaST are able to solve only images with the field of view larger than about 30°. If a narrow field images need to be solved, it is recommended to install ASTROMETRY.NET code locally on the computer running VaST and supply the code with only with index files corresponding to the field of view of the images. VaST will automatically detect a local ASTROMETRY.NET installation if its binaries are found in PATH.

Appendix A.14. Importing lightcurves from other software

It is possible to load lightcurves produced by other software into VaST. The lightcurves should be in three-column ("JD mag err") ASCII files. If such lightcurve files are placed in the directory ../lcfiles run

```
util/convert/three_column_ascii2vast.sh ../lcfiles/* ./find_candidates
```

http://psfex.readthedocs.io

ftp://scan.sai.msu.ru/pub/software/tiff2fits/
Appendix A.15. Transient detection

An overview of transient search with VaST is presented in Section 4. You'll need 1ibpng header files installed in your system for this mode to work. If they were not present the time you installed VaST, you will need to re-compile it by running

```
vast -x7 -uf
```

The following parameters are recommended:

```
/vast -x7 -uf
../transient_detection_test_Ceres/reference_images/* 
../transient_detection_test_Ceres/second_epoch_images/*
util/transients/search_for_transients_single_field.sh
```

here -x7 tells VaST to accept all detections with SExtractor flags 7 or lower (a transient may well be blended with background stars or saturated), -u (or --UTC) indicates that the UTC to TT time system conversion should not be performed (Section 3.1), -f tells the program not to start ./find_candidates. Instead of staring the interactive display, VaST will start a web browser displaying an HTML page presenting search results (Figure 7). If the Astrometry.NET code is installed locally (Section 3.7), it is possible to run the transient search and perform a basic search for known variables (VSX) and asteroids (astcheck) without having internet connection, but for a full investigation of transient candidates it is highly recommended to use also the online services. To keep the number of false candidates low, you will likely need to experiment with source extraction parameters as described in Appendix A.8.

Appendix A.16. Heliocentric correction

As the Earth orbits around the Solar system barycenter, it gets closer or further from distant celestial objects that are not at the ecliptic poles. The time light from a distant object needs to travel this extra distance may be as high as 499 seconds for an object at the ecliptic plane observed from the closest and furthest points of the Earth orbit. This extra time should be taken into account for accurate timing analysis.

VaST has a tool (based on NOVAS library; Kaplan et al. 1989) that applies heliocentric (center of the Sun) correction to a lightcurve given the equatorial J2000 coordinates of the observed object:

```
# If the input lightcurve is in TT
util/hjd_input_in_TT out123.dat 12:34:56.7 +12:34:56
# If the input lightcurve is in UTC
util/hjd_input_in_UTC out123.dat 12:34:56.7 +12:34:56
# The output JDs are always expressed in TT
```

If timing accuracy of better than 8 sec is needed, one should consider using an external software (e.g. VARTOOLS compiled with SPICE support; Acton 1996) to compute barycentric correction instead of heliocentric. The reason why only the simple heliocentric correction is implemented in VaST instead of barycentric correction is that the heliocentric correction can be computed using a few inline constants in the code, while the computation of barycentric correction requires a complex Solar system model.

Appendix A.17. Re-formatting a lightcurve for publication

VaST out*.dat lightcurve files include columns with information about object pixel coordinates, measurement aperture and local path to the image files that are not needed for a lightcurve attached to a publication of VSX database submission. VaST includes a tool that removes the unnecessary columns and sorts the input lightcurve in JD:

```
util/cute_ic out00268.dat > lc_00268.txt
```

Appendix A.18. Split multi-extension FITS image

VaST cannot properly handle multi-extension images – FITS files containing multiple images obtained at different epochs of with different CCD chips. Multi-extension images are commonly found in the HST and Swift/UVOT archives. These images should be split into multiple FITS files each containing only one image before they can be processed with VaST. A multi-extension file can be split using the tool

```
util/split_multiextension_fits multiextension_image.fits
```

Appendix B. Command line options

```
./vast is the main program that constructs lightcurves from a set of input images. It accepts a number of command-line options listed below.

-h or --help print help message
-9 or --ds9 use DS9 instead of pgfv to view FITS images
-f or --nofind do not run ./find_candidates after constructing lightcurves
-p or --poly use linear instead of polynomial magnitude calibration (useful for good quality CCD images)
-o or --photocurve use formulas (1) and (3) from Bacher et al. (2005, MNRAS, 362, 542) for magnitude calibration. Useful for photographic data
-P or --PSF PSF photometry mode with SExtractor and PSFEx
-r or --norotation assume the input images are not rotated by more than 3 deg. w.r.t. the first (reference) image
-e or --failsafe FAILSAFE mode. Only stars detected on the reference frame will be processed
-u or --UTC always assume UTC time system, do not perform conversion to TT
-k or --nojkeyword ignore "JD" keyword in FITS image header. Time of observation will be taken from the usual keywords instead
-a5.0 or --aperture=5.0 use fixed aperture (e.g. 5 pixels) in diameter
-b200 or --matchstarnumber=200 use 200 (e.g. 200) reference stars for image matching
```

Appendix A.8. Split multi-extension FITS image

VaST cannot properly handle multi-extension images – FITS files containing multiple images obtained at different epochs of with different CCD chips. Multi-extension images are commonly found in the HST and Swift/UVOT archives. These images should be split into multiple FITS files each containing only one image before they can be processed with VaST. A multi-extension file can be split using the tool

```
util/split_multiextension_fits multiextension_image.fits
```

Appendix B. Command line options

```
./vast is the main program that constructs lightcurves from a set of input images. It accepts a number of command-line options listed below.

-h or --help print help message
-9 or --ds9 use DS9 instead of pgfv to view FITS images
-f or --nofind do not run ./find_candidates after constructing lightcurves
-p or --poly use linear instead of polynomial magnitude calibration (useful for good quality CCD images)
-o or --photocurve use formulas (1) and (3) from Bacher et al. (2005, MNRAS, 362, 542) for magnitude calibration. Useful for photographic data
-P or --PSF PSF photometry mode with SExtractor and PSFEx
-r or --norotation assume the input images are not rotated by more than 3 deg. w.r.t. the first (reference) image
-e or --failsafe FAILSAFE mode. Only stars detected on the reference frame will be processed
-u or --UTC always assume UTC time system, do not perform conversion to TT
-k or --nojkeyword ignore "JD" keyword in FITS image header. Time of observation will be taken from the usual keywords instead
-a5.0 or --aperture=5.0 use fixed aperture (e.g. 5 pixels) in diameter
-b200 or --matchstarnumber=200 use 200 (e.g. 200) reference stars for image matching
```
Appendix C. Log files

After processing an image series, apart from the lightcurve files VAST will create a number of log files:

- **vast_summary.log** file summarizes the processing results. It indicates how many images were successfully processed, which image was used as the reference (Appendix A.3) what system time is used for JDs in lightcurve files (Section 3.1).

- **vast_image_details.log** contains information about processing of individual images including the start time and middle of exposure JD derived from the FITS header, aperture size used for this image, number of detected sources and how many of them are matched and if the overall image matching and magnitude calibration were successful, image rotation angle with respect to the reference image.

- **vast_command_line.log** stores all the command-line arguments specified for the latest VAST run.

- **vast_lightcurve_statistics.log** is the table with raw variability index values computed for all lightcurves that have a sufficient number of points (Section 3.3).

- **vast_lightcurve_statistics_normalized.log** is the table with variability index values normalized by their scatter estimated for each object’s magnitude.

**vast_lightcurve_statistics_format.log** describes the format of the variability index tables.

Appendix D. Comparison with AstroImageJ and Munin

In order to check the quality of photometry produced by VAST in aperture and PSF-fitting modes we compare it with two other aperture photometry packages: AstroImageJ (Collins et al. 2017) and Munin (Section 3.1). These two packages were selected for comparison as they are free and provide a friendly GUI for lightcurve construction. As the test dataset we used a series of R-band images of a candidate cataclysmic variable Gaia16bnz obtained with the 2.3 m Aristarchos telescope (Goudis et al. 2010, Bonanos and Boumis 2016) using a VersArray 2048 × 2048 e2v back-illuminated CCD, cooled by liquid nitrogen. The images cover the sky area of 5.5 × 5.5 arcminutes. The images were bias-subtracted and flat-fielded in VAST which was also used to weight the observing time of each image in the commonly accepted format using the DATE–OBS and EXPTIME keywords in the FITS header (Section 3.1).

As the target is one of the few brightest stars on the frame (which may cause problems for VAST when running with the default parameters), the following VAST command line options (Appendix B) were used to run the test in the aperture photometry mode:

```bash
./vast -a10 -up --noerrorsrescale --magzsizefilter \--notremovebadimages ../Helmos_Gaia16bnz_fixed_date/*
```

This requires VAST to use a single aperture 10 pixels in diameter for all images, do not perform UTC to TT time conversion (Section 3.1), use a linear function (4) for frame-to-frame magnitude calibration (Section 3.4), enable the magnitude-size rescaling (Section 3.2) and disable the bad image filter (as no similar filtering is offered by the other tools). For the PSF-fitting photometry we used the following command line:

```bash
./vast -P -a10 -up --noerrorsrescale --magzsizefilter \--notremovebadimages ../Helmos_Gaia16bnz_fixed_date/*
```

For AstroImageJ and Munin we set the measurement aperture diameter to 10 pixels and the sky annulus inner and outer diameters to 15 and 35 pixels. We used a single comparison star URAT1 697-107802 (Zacharias et al. 2015) assuming \( R = 13.648 \) – computed from APASS colors using (5). The same comparison star was used to set the zero-point of VAST photometry (Appendix A.6). The lightcurves obtained with the different methods agree within 0.005 mag (or) and 0.003 mag (MAD). A section of the resulting lightcurves is presented in Figure D.3

The errorbars reported by AstroImageJ and Munin are a factor of three larger than the ones reported by VAST because they take into account different sources of errors (resulting from the difference in magnitude calibration technique). The VAST errorbars are derived from the combination of the background and photon noise corresponding to the target (and scaled with

---

47 [http://scan.sai.msu.ru/vast/Helmos_test](http://scan.sai.msu.ru/vast/Helmos_test)
48 [http://helmos.astro.noa.gr/](http://helmos.astro.noa.gr/)
the magnitude calibration function described in Section 3.4. They neglect the uncertainty in magnitude zero-point determination as it is derived from all stars matched between the current and reference frames. The errorbars reported by astroImageJ and Muniwin include the uncertainty in magnitude zero-point determination which in this case include the photon and background noise from a single comparison star that is fainter than the target. Normally, the VaST errorbars are rescaled following the procedure described by Wyrzykowski et al. 2009, Zinn et al. 2017. However, this rescaling procedure results in overestimating the errors for the brightest stars (if there are few brighter stars on the frame) and was disabled for this test.

The candidate cataclysmic variable may be identified as the object having a higher lightcurve scatter compared to other objects of similar brightness. Figure D.8 presents the lightcurve scatter versus magnitude plots computed with VaST and Muniwin. AstroImageJ has no built-in capability to visualize magnitude-scatter plots so it is not considered here. VaST and Muniwin have a different way to quantify lightcurve scatter. In VaST the default measure of scatter is the unweighted standard deviation computed over a lightcurve from which 5 per cent of brightest and faintest points were removed, but not more than 5 points from each side (see appendix in Pashchenko et al. 2017). In addition to this clipped σ VaST has other ways to characterize scatter and shape of a lightcurve (Table I). Muniwin uses a custom robust measure of lightcurve scatter as the variability detection statistic. The few objects with R > 16 showing elevated lightcurve scatter in the Muniwin plot were automatically identified as blended (Section 3.2) and excluded from the VaST analysis.

Figure D.9 also illustrates the difference between the aperture and PSF-fitting photometry with VaST. While for the brighter point-like sources both techniques produce a comparable measurement accuracy, for fainter sources PSF-fitting results in much smaller lightcurve scatter than aperture photometry. The effect is exaggerated here as VaST was forced to use a large aperture in order to maximize photometric accuracy for the bright target at the cost of the elevated background noise mostly affecting the fainter sources.

References

Acton, C.H., 1996. Ancillary data services of NASA’s Navigation and Ancillary Information Facility. Planet. Space Sci. 44, 65–70.
Alard, C., 2000. Image subtraction using a space-varying kernel. A&AS 144, 363–370.
Alard, C., Lupton, R.H., 1998. A Method for Optimal Image Subtraction. ApJ 503, 325–331. astro-ph/9712287
Antipin, S.V., Sokolovsky, K.V., Lebedev, A.A., 2005. GSC 02050-00745: A New RR Lyrae Star with Blazhko Effect. Information Bulletin on Variable Stars 5654.
Bacher, A., Kimeswenger, S., Teutsch, P., 2005. Photometry from online Digitized Sky Survey plates. MNRAS 362, 542–548. astro-ph/0507536
Becker, A., 2015. HOTPANTS: High Order Transform of PSF AND Template Subtraction. Astrophysics Source Code Library. [1504.0004]
Belardi, C., Kilic, M., Munn, J.A., Gianninas, A., Barber, S.D., Hey, A., Stetson, P.B., 2016. The DECam minute cadence survey - I. MNRAS 462, 2506–2517. [1607.08773]
Bertin, E., 2006. Automatic Astrometric and Photometric Calibration with SCAMP, in: Gabriel, C., Arviset, C., Ponz, D., Enrique, S. (Eds.), Astronomical Data Analysis Software and Systems XV, p. 112.
Bertin, E., 2011. Automated Morphometry with SExtractor and PSFEx, in: Evans, L.N., Accomazzi, A., Mink, D.J., Rots, A.H. (Eds.), Astronomical Data Analysis Software and Systems XX, p. 435.
Bertin, E., Arnouts, S., 1996. SExtractor: Software for source extraction. A&AS 117, 393–404.
Bertin, E., Mellier, Y., Radovich, M., Missonnier, G., Didelon, P., Morin, B., 2002. The TERAPIX Pipeline, in: Bohlender, D.A., Durand, D., Handley, T.H. (Eds.), Astronomical Data Analysis Software and Systems XI, p. 228.
Bessell, M.S., 2005. Standard Photometric Systems. ARA&A 43, 293–336.
Bonanos, A.Z., Boumis, P., 2016. Evidence for rapid variability in the optical light curve of the Type Ia SN 2014J. A&A 585, A19. [1507.01011]
Bonnarel, F., Fernique, P., Bienaymé, O., Egré, D., Genova, F., Louys, M., Ochsenbein, F., Wenger, M., Bartlett, J.G., 2000. The ALADIN interactive sky atlas. A reference tool for identification of astronomical sources. A&AS 143, 33–40.
Bramich, D.M., Horne, K., Alsubai, K.A., Bachelet, E., Mislinis, D., Parley, N., 2016. Difference image analysis: automatic kernel design using information criteria. MNRAS 457, 542–574. [1512.04655]
Deeg, H.J., Doyle, L.R., 2001. V APHOT - A Package for Precision Di
Dekany, I., Minniti, D., Majaess, D., Zoccali, M., Hajdu, G., Alonso-García, 
Drake, A.J., Djorgovski, S.G., Mahabal, A., Beshore, E., Larson, S., Graham, 
M.J., Williams, R., Christensen, E., Catelan, M., Boattini, A., Gibbs, A., 
Hill, R., Kowalski, R., 2009. First Results from the Catalina Real-Time 
Transient Survey. ApJ 696, 870–884. 10.1086/594372
Eastman, J., Siverd, R., Gaudi, B.S., 2010. Achieving Better Than 1 Minute 
Accuracy in the Heliocentric and Barycentric Julian Dates. PASP 122, 935. 10.1086/594372
Ferreira Lopes, C.E., Cross, N.J.G., 2016. New insights into time series analy-
sis. I. Correlated observations. A&A 586, A36. 10.1051/0004-6361/201526052
Ferreira Lopes, C.E., Cross, N.J.G., 2017. New insights into time series analy-
sis. II - Non-correlated observations. A&A 604, A121. 10.1051/0004-6361/201629147
Friedrich, S., Koenig, M., Wicenec, A., 1997. Search for Variables in the TY-
CHO Epoch Photometry Annex B, in: Bonnet, R.M., Heg, E., Bernacca, P.I., Emiliani, L., Blauw, A., Taroc, L., Kozhevnikov, J., Lindegren, L., Hassan, H., Bouffard, M., Strim, B., Heger, D., Perryman, M.A.C., Woltjer, L. (Eds.), Hipparcos - Venice'97. pp. 441–444.
Fruth, T., Kabath, P., Cabrera, J., Chini, R., Czesznuida, E., Eigmüller, F., Piko-
rs, A., Kirste, S., Lemke, R., Murphy, M., Pasternacki, T., Rauer, H., Titz-
Weider, R., 2012. Improved Variable Star Search in Large Photometric Data 
Sorts/New Variables in CoRoT Field L102 Detected by BEST II. AJ 143, 1400–1405. 10.1088/0004-6256/143/5/1400
Gaia Collaboration, Prusti, T., de Bruijne, J.H.J., Brown, A.G.A., Vallenari, A., Babusiaux, C., Bailer-Jones, C.A.L., Bastian, U., Biermann, M., Evans, D.W., et al., 2016. The Gaia mission. A&A 595, A1. 10.1051/0004-6361/201629050
Gouls, C., Hantzos, P., Boumis, P., Xilouris, E., Katsiyannis, A., Maroussis, A., 2010. The Aristarchos Telescope. in: Tsonganos, K., Hatzidimitriou, D., Matsakos, T. (Eds.), 9th International Conference of the Helenic Astronomical Society, p. 422.
Gran, F., Minniti, D., Saito, R.K., Zoccali, M., Gonzalez, O.A., Navarrete, C., Galvez, J., Thilker, D., Urata, Y., Valenti, J., Walter, F., Waters, C., Page, M.J., Poole, T.S., Roming, P., Smith, P.J., Still, M., Yerles, E., 2016. The Pan-STARRS1 Sky Catalog. A&A 573, A100. 10.1051/0004-6361/201628795
Greisen, E.W., Calabretta, M.R., 2002. Representations of world coordinates in FITS. A&A 395, 1061–1075. astro-ph/0204707
Grindlay, J., Tang, S., Los, E., Servillat, M., 2012. Opening the 100-Year Window for Time-Domain Astronomy, in: Griffin, E., Hanisch, R., Seaman, R. (Eds.), New Horizons in Time Domain Astronomy, pp. 29–34. 10.1081/ASPC.403.2117
Groote, D., Tuukvete, T., Edelmann, H., Arlt, R., Heber, U., Enke, F., 2014. Building up APPLAUSE: Digitization and web presentation of Hamburger Sternwarte plate archives, in: Astronoptale 2014, p. 53.
Groth, E.J., 1986. A pattern-matching algorithm for two-dimensional coordi-
inate lists. AJ 91, 1244–1248.
Gupta, A.C., Krichbaum, T.P., Vitia, J.P., Rani, B., Sokolovsky, K.V., Mohan, P., Mangalam, A., Marchili, N., Fuhrmann, L., Agudo, I., Bach, B., Bacher, R., Böttcher, M., Gabanyi, K.E., Gaur, H., Hawkins, K., Kimeridze, G.N., Kurtanidze, O.M., Kurtanidze, S.O., Lee, C.U., Liu, X., McBreen, B., Nesici, R., Nestoras, G., Nikolashvili, M.G., Ohlert, J.M., Palma, N., Peneva, S., Pursimo, T., Semkow, E., Strigachev, A., Webb, J.R., Wiesemeyer, H., Zen-
sus, J.A., 2012. Multiwavelength intraday variability of the BL Lac object 7E = 0716–714. AJ 425, 1357–1370. 10.1051/0004-6264/201219032
Hartman, J.D., Bakos, G.A., 2016. VARTOOLS: A program for analyzing astronomical time-series data. Astronomy and Computing 17, 1–72. 10.1016/j.ascom.2016.05.001
Henden, A., Munari, U., 2014. The APASS all-sky, multi-epoch BVgr photom-
meter survey. Contributions of the Astronomical Observatory Skalnate Pleso 43, 518–522.
Henden, A.A., Templeton, M., Terrell, D., Smith, T.C., Levine, S., Welch, D., 2016. VizieR Online Data Catalog: AAVSO Photometric All Sky Survey (APASS) DR9 (Henden++, 2016). VizieR Online Data Catalog 2336.
Holt, J.S., 2013. A fast matching algorithm for sheared stellar samples: k-d
match. MNras 433, 935–939. 10.1093/mnras/sts576
Hoffmann, S.L., Macri, L.M., 2015. Cepheid Variables in the Maser-Host Galaxy NGC 4258. AJ 149, 183. 10.1088/0004-6256/149/1/183
Hög, E., Fabricius, M., Makarov, V.V., Urban, S., Corbin, T., Wycoff, G., Basti-
an, U., Schwekkink, P., Wicenec, A., 2000. The Tycho-2 catalogue of the 2.5 million brightest stars. A&A 355, L27–L30.
