Baltic Astronomy, vol.00, 000–000, 2002.

TRIPP: AN APERTURE PHOTOMETRY PACKAGE FOR THE REDUCTION OF CCD TIME SERIES IMAGES

S.L. Schuh[1], S. Dreizler[1], J.L. Deetjen[1], E. Göhler[1]

[1] *Institut für Astronomie und Astrophysik, Universität Tübingen, Sand 1, D-72076 Tübingen, Germany*

Received September 20, 2002

Abstract. TRIPP is an aperture photometry program designed with the purpose of extracting light curves from large sets of similar CCD frames typically obtained during time-resolved photometric monitoring campaigns such as, for example, WET runs. We describe its properties and usage with an emphasis on where the functionality of TRIPP may differ from similar programs, and try to outline both its strengths as well as the non-trivial issues where difficulties may arise.

Key words: techniques: image processing – techniques: photometric

1. INTRODUCTION

TRIPP is an aperture photometry program for CCD frames written in IDL. A single parameter file (called "log file" hereafter) and an extraction mask guide the data processing. IDL is fast, allowing to go through the following steps in considerably less than typical cycle time: bias and flatfield correction, locating a specified source on all frames, extracting stellar and sky fluxes for several sources, then applying the background correction and calculating a relative light curve. A typical reduction procedure is outlined in Sect. 7. Before that, Sect. 2, 3 and 4 detail the technical prerequisites and describe a few general principles of the software. Sect. 5 briefly considers the bias and flatfield correction procedures, while Sect. 6 gives some particular information about the aperture photometry. Data formats resulting from a typical run of the program are described in Sect. 8. A comparison with other programs (Sect. 9), a list of special features (Sect. 10) and of ideas for the future (Sect. 11) follow.

The overall concept is to mention features in the chronological order in which they will appear to somebody who is actually setting up and using the program.

2. TECHNICAL REQUIREMENTS

2.1. Interactive Data Language

The Time Resoved Imaging Photometry Package (TRIPP) is a set of procedures written in the Interactive Data Language (IDL). Running TRIPP therefore requires a full IDL installation (version 5.0 or higher), including a licence that has to be purchased. Astronomical institutes often provide the IDL software, and many mathematical and astronomical libraries have been developed in the IDL programming langage. The most important one is probably the so-called astrolib, compiled by the Goddard Space Flight Center (NASA) and freely available from their website. It includes the IDL-DAOPHOT package, an adapted version of the Fortran DAOPHOT program. Parts of this package as well as other components of the astrolib are used in TRIPP; the availability of these libraries therefore needs to be ensured in order to run TRIPP. Finally, TRIPP also uses procedures from the aitlib, a compilation of programs developed at the Institut für Astronomie und Astrophysik Tübingen. These are also available from the web, but since they are not usually part of a standard installation, we provide all routines that are used packaged with every distribution of TRIPP.

2.2. Operating systems

In principle, IDL programs can be written independently of an operating system (OS). However, TRIPP works with directory structures and occasionally a few OS-level commands that assume a UNIX-type OS. While it should not in principle be impossible to adapt TRIPP for other systems, the current distribution has only been developed and tested for UNIX-type operating systems, and there are no plans to implement the necessary changes for extended portability in the near future.

2.3. Format of input data

The project of developing TRIPP (based on and as a successor to the previously existing CCD package by Geckeler, 1998) was directly triggered by the need to reduce CCD data obtained as a contribution to a WET campaign. It has therefore been set up to handle large numbers of image files in FITS format[1] that are identical to each other with respect to their instrumental setup such as filter and binning used, and similar to each other with respect to their windowing[2] (if any) and approximate location of the relevant sources on the frames. Another important requirement is that all frames in a dataset need to have a common prefix and a four-digit number that identifies subsequent frames, followed by the .fits extension. Since most data acquisition systems produce such a format or at least one that is very similar, it is usually unproblematic to meet these specifications, and if not to either adapt the corresponding routines of TRIPP to a slightly different naming scheme or, which is often quicker, to rename the input data files via a simple external shell script.

The need to make use of some of the information contained in FITS headers, which is standardised only to a certain degree, is harder to deal with, and is treated in some more detail in Sect. 4.

2.4. Disk space and memory

Obviously, the computer to be used for the data reduction needs to provide sufficient disk space to store the raw data, and sufficient memory to work with it. Note however that TRIPP will again double the number of image files since formally, a bias and flatfield correction is always being done (see Sect. 5.), and original data are never overwritten. The format used for images is FITS format throughout, so that any additional overhead introduced by conversion to other formats is suppressed. Several output files will be added.

3. GENERAL CHARACTERISTICS

3.1. Adaptability

For regular IDL users, the borders between using existing programs and writing own procedures are often floating, since IDL is an interpreter with a command line interface. This allows any user to adapt TRIPP to her own needs, and to maybe contribute additional solutions that will be of interest for the main version. This process is in fact what has made TRIPP evolve from its first preliminary version (Schuh et al. 2000) to the much more elaborate package that it is today.

[1] http://heasarc.gsfc.nasa.gov/docs/heasarc/fits.html

[2] selection of a subarea of a CCD chip

At the same time however, TRIPP has been designed with the idea in mind that it should be, first of all, easy to use, even for first-time IDL users. After all, many aperture photometry packages are on the market, and they give good results – but some only after considerable effort has been put into getting them to run properly for the data at hand. TRIPP tries to automate as many steps as possible, while attempting to offer as many possibilities for a direct visual inspection of the usefulness of specific parameters as necessary.

3.2. Log files and recommended directory structure

For an immediate and comprehensible access to the parameters used in the overall reduction process of a data set, TRIPP works through the concept of so-called "log files" to share information. It is recommended to have a special directory where the log files for a collection of data sets will be stored, which will ideally then also be the working directory for IDL, and, depending on the set-up of the IDL installation, it may additionally have to provide the storage location for the TRIPP routines themselves, e.g. during a temporary installation at observatories. A permanent set-up should of course allow for a separate storage of the code in libraries.

The log file will collect most of the information necessary to run the data reduction, and can be complemented or corrected during the course of the processing as needed. It is however necessary to fill in default values of the appropriate data type right at the start (see Sect. 4.1). Essential entries required for the log file are the path for the directory where the original data are stored, and the path for the directory where the reduced data and data products should be placed. It has often proven to be a useful solution to maintain, on the same hierarchic level, the directories `Logs/`, `data/`, and `data_reduced/`, where `data/` could also be a link pointing to wherever the orginal data is physically stored, and `data_reduced/` is empty at the beginning but must exist.

The log files themselves are created or can be edited by invoking one of `tripp_write_zero_log`, `tripp_write_flat_log` or `tripp_write_image_log`, respectively, which in turn bring up a graphical interface where the required information can be conveniently filled in. Non-existing log files are filled with default values (see Sect. 4.1) which can then be edited; in every subsequent call with an existing file as argument the edited entries are restored and can be further altered. The log file constitutes the main argument in the calls to all following routines throughout the reduction process, and is the only argument that is necessarily required since it already transports most of the information that will be needed along the way.

3.3. Modes

Depending on the amount of interaction that the user thinks is necessary to ensure an adequate processing of the data, four different modes can be distinguished for several of the individual steps. The default mode may be called "automatic"; this simply means that a procedure is called with the log file name as the only argument. In this mode, procedures will ask for user interaction only when absolutely necessary (these are marked with an asterisk in Table 2). Often, some kind of action is required either just at the start of a procedure that will then run on its own for a longer period of time, or during a shorter procedure that basically is finished as soon as the user has provided what was requested.

To increase the user's influence on a specific process, several procedures offer an "interactive" mode which is switched on by setting the `/mouse` keyword. It increases the possibility to control what is happening and can become especially important for the positioning process (where the location of the source on a frame is determined, see Sect. 6.1).

Two procedures that run for a longer period of time have a simultaneous graphical output to allow a direct quality control, which in case of unsatisfactory results may prompt the user to re-start the process with the `/mouse` keyword switched on. When everything runs smoothely however the graphical output produces an overhead and causes a substantial increase in computation time that one might want to get rid of. This is what the "number crunching" mode is for, switched on by the `/silent` keyword. It supresses both excess graphical output as well as some of the text output.

Finally, in the other direction, when even in the "interactive" mode the desired results cannot successfully be produced, the "debug" mode may be switched on for the most delicate procedure using `/debug` to help search for possible causes. It adds further graphical and text output that might be helpful in the diagnostics.

To check what modes are available for which routines, please consult Table 2.

4. NECESSARY ADJUSTMENTS

4.1. Defaults for log files

The first and also one of the more time consuming steps in any data processing with TRIPP is filling out the log file, be it to produce the mean bias or flatfield images, or for the reduction and extraction itself. For the individual entries that need to be considered, the reader is referred to the user's manual. It is noted here that the default entries that appear when a new log file is first invoked can be changed by editing the source code of `tripp_log_type`. This can help in making the editing more efficient.

4.2. Bias and flatfield cutouts

Although standardised as a format, FITS files generally use many different header keywords, often to describe the same information, and not all formats are equally informative. Different formats therefore have to be made understandable to TRIPP using specialised interfaces for each of them (see Sect. 4.3). The rule is to collect the necessary header information in a single place, but one of currently two exceptions to this rule is the critical step of reconstructing the location of a windowed (science) frame with respect to a fully readout calibration frame in `tripp_reduction`. It does not matter to TRIPP whether the calibration frames are of larger dimension than the science frames or not. But since specific FITS header keywords are used to determine this, the procedure will not work when those keywords are different from what is expected. Whenever the science and calibration frames are of the same size, a quick remedy to this problem is to simply delete the part of the code in the local copy that attemps to do the – unneccessary – determination of offsets and use trivial default values instead. Otherwise, an individual solution has to be found.

A test run of `tripp_reduction` will very quickly show if any of these measures has to be taken.

4.3. Time stamps

The very first entry in the log file is an identifier of the telescope/instrument the data has been taken with. This identifier is used as a code for the FITS file format to be expected (as explained in Sect. 4.2). The main information to be extracted from a certain format in the interface is the timing information; therefore, the procedure `tripp_get_gjd` (gjd: geocentric julian date) has grown to be the module where all available interfaces are compiled. Table 1 lists the tested observatory/telescope/instrument configurations available at the time of writing and the identifier they can be addressed with. For a first test run, it can also be convenient to simply use one of the identifiers JD, MJD or HJD if the respective keyword is present in the FITS file header with sufficient accuracy. For a final exact determination of the time stamps, one of the identifiers has to be carefully selected from the list by looking at how the information in the header is processed to eventually yield the middle of the exposure in reduced julian date, or a new identifier and the corresponding prescription to calculate the time has to be introduced by amending `tripp_get_gjd`. Detailed knowledge about the meaning of the timing entries in the FITS header is necessary; it can vary from case to case and should already have been checked at the telescope.

Additionally, a constant offset in time as caused by chip wiping procedures preceding the exposure, or systematic shutter delays, can always be added individually to each data set via the log file.

4.4. Graphical output

The exact appearance on the screen of graphical output produced by TRIPP can vary for different systems. Sometimes, the dimensions of graphical user interfaces or plot windows are not optimal for a given system and must then be adjusted to allow convenient working. The authors will of course help in identifying the pieces of code that need to be changed.

Table 1. Currently implemented identifiers

Identifier	Observatory / Telescope / Instrument
WENDELSTEIN	(Wendelstein Observatory, Germany)
SA_CCD	(SAAO, South Africa)
SAAO	(SAAO, South Africa)
CA1.2	(Calar Alto, Spain, 1.2m, before 2000)
CAN1.2	(Calar Alto, Spain, 1.2m, after 2000)
CA2.2	(Calar Alto, Spain, 2.2m with CAFOS, after 2000)
BUSCA	(Calar Alto, Spain, 2.2m with BUSCA)
EFOSC	(ESO La Silla, Chile; decomissioned)
ST7_E	(IAAT, Tübingen, Germany, 40cm)
MSSSO12	(Mount Stromlo/Siding Springs Observatory, 1.2m)
GUNMA	(GUNMA Observatory, Japan)
SARA	(SARA Observatory, USA)
Bohyunsan	(Bohyunsan Observatory, Korea)
WISE	(WISE Observatory, Isreal)
MJD	(read MJD keyword)
HJD	(read HJD keyword)
JD	(read JD keyword)

5. BIAS AND FLATFIELD CORRECTIONS

5.1. Median bias

It depends on the instrumental properties whether a bias correction is sufficient or whether a dark correction should be done instead. Although the naming of the routines in the following suggests otherwise, either can be used.

To produce a median bias (or zero, or dark) frame, a log file needs to be written via `tripp_write_zero_log`, followed by a run of `tripp_zero`. The latter starts with an ample visual inspection of each calibration exposure to let the user decide for each file if it is of good enough quality to be included in the calculation. A pre-selection is therefore not necessary.

5.2. Median flat

To produce a median bias-corrected flatfield exposure, a median bias frame needs to exist already. In all other respects, the duo of `tripp_write_flat_log` and `tripp_flat` operates in a very similar way to that described in Sect. 5.1. Additionaly the possibility of automatical removal of stars from the flat internally is given by using parts of the DAOPHOT Package.

5.3. Using or suppressing the bias and flatfield corrections

Median bias and flatfield frames may or may not be produced using TRIPP; it does not matter to the reduction and extraction part where these calibration frames came from since FITS format is used throughout.

Furthermore, bias and flatfield corrections may or may not be done using TRIPP; the extraction can also be run on pre-reduced (i.e. bias and flatfield corrected) frames. In this case, the entries in the log file for "Zero correction" and "Flat correction" can simply be set to "no", then a dummy bias equal to zero everywhere and a dummy flat equal to unity everywhere will be used instead.

For a quick-look reduction or test run of TRIPP, it can also be desirable to take advantage of these options and simply ignore the bias and flatfield corrections at first.

6. THE APERTURE PHOTOMETRY

6.1. Positioning process

A prerequisite for the aperture photometry is finding the exact location of a source on all frames. TRIPP currently uses a combination of three methods to achieve this: a 2-D cross-correlation, a search for the centroid of the star, and, if these don't agree within a certain error limit, a determination by the user via mouse click (only in the interactive `/mouse` mode, otherwise only a warning will appear). It is important to note the following limitation: the source chosen for the positioning process will become the object for which a relative light curve will eventually be calculated.

Both the calibration of the science frames for bias and flatfield effects as well as the process of locating the selected source on each frame are done in `tripp_reduction`. In addition to the full set of "mode" keywords, several other keywords and parameters are also accepted, the most important one being `csize` which can be used to enlarge the area used for the 2-D cross-correlation. The 2-D cross-correlation is not done on the full frame by default to save computation time; but it may become necessary to incease the correlation area by setting `csize` to a different value and trade speed for a higher stability of the process.

Another complication may also trigger such a decision: The area accessible for an interactive determination of the source position via mouse click is limited to the area also used for the cross-correlation. If the source moves a significant distance away from its original position and drifts outside the selected correlation area, both the automatic determination by cross-correlation as well as the determination by hand will fail. It can then become necessary to make use of the full chip size.

On the other hand, this implies that TRIPP can not only follow a drifting source over large distances across the chip, but (in the interactive `/mouse` mode) can also routinely handle sudden source displacements. Rotation of the field can not be accounted for.

6.2. Source apertures

The apertures used for the sources are circular, and the extraction of source fluxes is done for a set of different radii. A circular aperture is approximated by dividing a full pixel in many small subpixels, and determining which of those lie within the given radius: the aperture is "softened". The total number of different aperture sizes, the minimum and maximum radius value to be used, and the best guess for the final extraction radius are all entries requested for the log file, and will be visualised during the definition of the extraction mask.

6.3. Sky measurements

For reasons of speed, simpler apertures are used for the sky measurements. Six squares are positioned around each source, and each square consists of integral pixels only. As for the source apertures, the side length of the squares and their distance from the source center (for a first, automated placement of the fields which can be interactively altered later) are also log file entries. Their position can be set extraction mask definition process.

The sky flux is extracted for each field separately as the median value, then their sum, appropriately scaled with the extraction area, is used as the final measurement associated with one source. From a statistical point of view, this may not be ideal, but gives better results than other permutations of the ordering of these processes.

7. A TYPICAL REDUCTION PROCEDURE

7.1. Overview in Table 2.

The necessary procedure calls are listed in chronological order in Table 2. It additionally gives the available keywords for the modes, as well as a schematic primer to other optional keywords or parameters that may be of interest ("`/special`"). A more detailed description can be found in the user's manual or in the documentation (both available on the web, see References).

In addition to the procedures whose results are relied upon by their successors, there also exist procedures that are optional (and marked as such in Table 2). Their purpose is to allow a quality control of intermediate steps in the data processing. They are characterised by the common prefix `tripp_show`. Furthermore, there is a similar group of routines characterised by the common prefix `tripp_write` that generate data products. In Table 2 the extensions of the different data formats are listed.

7.2. Log file, reduction and positioning

Once the median bias and flatfield calibration frames are available, if necesssary at all, the main part of the reduction starts with the creation of a log file using `tripp_write_image_log`.

Table 2. Procedure calls with important keywords, and markers for optional procedures - see text for more.

tripp_write_zero_log	,logfile	*		opt.
tripp_zero	,logfile	*	[,/special]	opt.
tripp_write_flat_log	,logfile	*		opt.
tripp_flat	,logfile	*	[,/special]	opt.
tripp_write_image_log	,logfile	*		
tripp_reduction	,logfile	[,/debug ,/mouse	,/silent ,/special]	
tripp_show_pos	,logfile	*		opt.
tripp_define_mask	,logfile	*	[,/special]	
tripp_extract_flux	,logfile		[,/silent ,/special]	
tripp_show_raw	,logfile		[,/special]	opt.
tripp_write_wetstandard	,logfile	*		.wet
tripp_calc_relflux	,logfile		[,/special]	
tripp_show_relflux	,logfile	[,/mouse]		opt.
tripp_write_relflux	,logfile		[,/special]	.dat
tripp_write_final	,logfile	[,/mouse	,/special]	.fin
tripp_write_error	,logfile			.err
tripp_show_final	,logfile		[,/special]	opt.

Next, `tripp_reduction` applies the calibration corrections and performs the positioning process. The quality of the positioning can be verified with `tripp_show_pos`.

7.3. Mask, extraction and raw data

An extraction mask is constructed with `tripp_define_mask`. The total number of sources and the dimensions of apertures have to be defined in the log file as touched upon in Sect. 6.2 and 6.3. If the dimensions on the mask do not look satisfactory, the log file entries should be edited and the process of mask definition repeated. To make this iterative process efficient, the mask is first generated in a half-automated manner, then further changes can be applied. The mask is then used by `tripp_extract_flux` to do the actual aperture photometry. The raw extracted data can be visualised with `tripp_show_raw`, and finally stored in ASCII format with `tripp_write_wetstandard` (see also Sect. 8.1).

7.4. Relative light curves

A light curve of the selected source with respect to a defined sample of reference stars is computed by `tripp_calc_relflux`. The total number of reference stars to be used also has to be specified in the log file, as well as their names in the mask (this is what the second graphical interface in the log file definition is for). Their maximum number is restricted to 10. The relative light curve as well as its periodogram can be displayed with `tripp_show_relflux` (for several extraction radii). Then the light curve (for the unique, selected extraction radius given in the log file which can be edited before continuing) can be written into an ASCII data file with `tripp_write_relflux`.

Beyond that, `tripp_write_final` offers several possibilities to delete points from the overall data and to correct for trends using low-order polynomials. The procedure `tripp_write_error` calculates the errors for these cleaned data points and also writes them into an ASCII file, again together with the final light curve. The final light curve and its peridogram are plotted in `tripp_show_final`.

8. DATA FORMATS

8.1. Light curves and errors

Although the dedicated procedure `tripp_write_wetstandard` exists, it does not yet comply to WET specifications at the time of writing. An update is in preparation; the resulting data file should then contain essentially the raw extracted data, preceded by an informational header.

As described in Sect. 7.4, relative light curves, containing the time in reduced julian date and the flux in relative intensity units, are generated by `tripp_write_relflux` and `tripp_write_final`. The data files are simple ASCII x,y tables containing no header, which makes them easily suitable for further processing by many other programs.

Finally, `tripp_write_error` adds a third column to this format which contains errors for the relative fluxes, estimated from the standard deviation of the reference stars.

8.2. Informational Plots

Many of the informational plots created by the group of "show" routines are output in the postscript file format (extension .ps). In addition to the representation of the raw data, the unmodified relative data and its periodogram, and the optimised relative data with its periodogram, a further product worth mentioning is an image of the extraction mask to identify the reference stars. It is generated early on as a by-product of the mask definition.

8.3. Overall data flow

TRIPP routines are generally quite verbose about which files they use as input and what output they produce. To visualise this process, Fig. 1 schematically displays an overview of the data flow.

9. COMPARISON WITH OTHER PROGRAMS

The performance of TRIPP has been tested on the "standard data set" proposed by O'Donoghue et al. (2000); the results of a comparison with an extraction by O'Donoghue et al. using the IRAF aperture photometry task `apphot` in the `digiphot` package have been described in Schuh et al. (2000). The outcome of that test was satisfactory, but it is very clear that a more systematic and extensive series of tests would be desirable here.

10. SPECIAL FEATURES

10.1. The "intup" image

As an additional quality control for the positioning process in `tripp_reduction`, the shifts found are used to add up all frames. Oblate, exceptionally large or double stellar images in the integrated image are a tell-tale sign for difficulties in the position finding process. A successfully constructed integrated image, on the other hand, is not only a joy to contemplate, especially while it slowly builds up, it is also useful to identify weak sources, invisible in individual frames, in the vicinity of potential reference stars. It is therefore by default used to define the extraction mask.

10.2. Online capability

To monitor incoming data during time series observations with CCDs, TRIPP has a monitor function called `tripp_monitor`. A recent, still experimental addition to that monitor is a `/quicklook` option, which runs a very rudimentary version of the reduction process in a highly automated mode. The trick here is that for the time-intensive parts, all previous results are restored ("recycled") and only the newest image is measured.

The quicklook analysis is independent of any data acquisition proccess. A log file has to be prepared exactly as for a normal run, except that the name of the last image is simply equal to the first image. Then as soon as the first image has been acquired, it can be used to define the extraction mask. Everything else is handled by

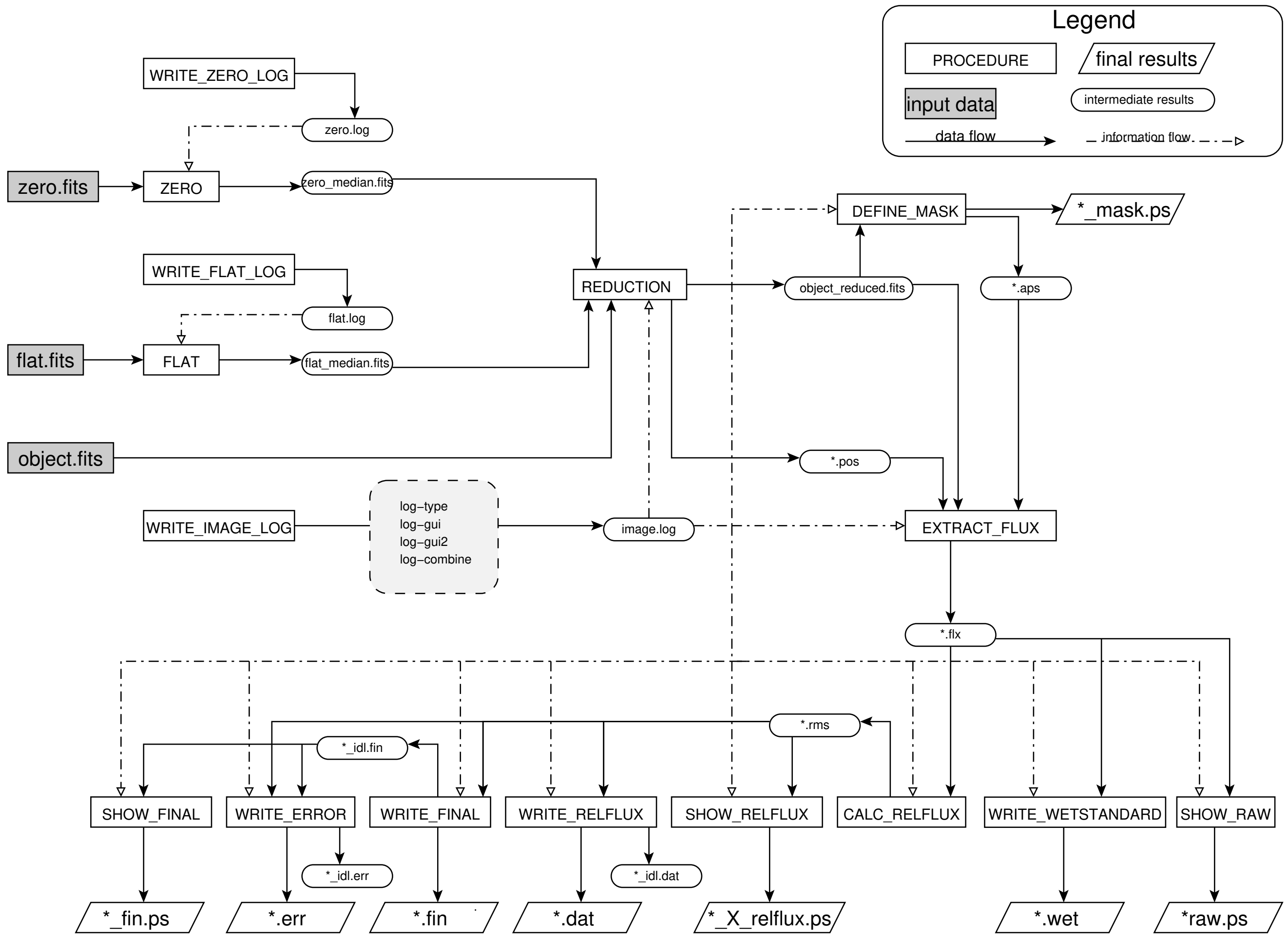

Fig. 1. Data flow in TRIPP.

`tripp_monitor`, called with the log file name as an argument, and the `/quicklook` option switched on.

The reduction should then quickly catch up with the acquisition, so that the growing light curve can be watched in real time. Not too surprisingly, however, this highly complex integration of the building blocks of TRIPP is more vulnerable to pertubations than the individual blocks just by themselves.

10.3. Windowed readout: handling of patchwork images

TRIPP is routinely used on frames acquired in windowed readout mode; however, so far it has only ever had to deal with one window. For simple lack of such data, patchwork images have not been an issue so far. But while the bias and flatfield corrections in those cases are certainly not trivial, the extraction process should in principle also be applicable to several windows pieced together, as long as they come in normal 2-D FITS format.

10.4. Frame transfer methods

Another experimental mode of TRIPP is the photometry on frames that combine several exposures regularly shifted with respect to each other. While the technical handling of such multi-exposure frames works fine, we do not have an estimate of the scientific quality obained in such a frame transfer mode (which differs from classical frame transfer between two halves of a chip).

10.5. Time series analysis tools

In addition to the aperture photometry, more TRIPP routines exist that are elementary wrappers for sophisticated time series analysis tools supplied by the "timing" library of the aitlib. Due to their limited flexibility, and less strict compatibility maintenance, these routines are provided but not advertised for in particular. The interested reader is instead referred directly to the underlying libraries at `http://astro.uni-tuebingen.de/software/idl/aitlib/timing/`.

11. IDEAS FOR THE FUTURE

For historical and compatibility reasons, many intermediate results are currently stored in the IDL SAVE format. This should be converted to a different system with higher portability.

A module to do a sophisticated barycentric correction has been created; its implementation into TRIPP is in preparation.

The positioning process could possibly be further stabilised by additionally implementing an angle/distances matrix method.

The accuracy of the photometry could perhaps be enhanced by copying the "MOMF" combination approach for aperture and PSF fitting photometry as explained in Handler (2003).

The time series analysis part could be upgraded to a more homogeneous package.

The use of TRIPP could be further simplified by supplying a common interface for all of its components. Meanwhile, we invite you to take a look at what is currently available and help evaluate it; the code can be downloaded at the TRIPP pages (see References).

ACKNOWLEDGEMENTS. The authors would like to thank (in chronological order) Ralf D. Geckeler, Thomas Rauch, Jörn Wilms, Sara Benlloch-Garcia, Katja Pottschidt, Patrick Risse and Thomas Gleissner for their contributions to TRIPP. Test users have included Iris Traulsen, Agnes Hoffmann, Thorsten Nagel, Nicolay Hammer, and Sigi Falter.

IDL is a software product by RSI (Research Systems - a Kodak company, see also `http://www.rsinc.com/idl`).

REFERENCES

Geckeler R.D. 1998, PhD thesis, Universität Tübingen

Handler G. 2003, in The 6^{th} Whole Earth Telescope Workshop (these proceedings), eds. J.-E. Solheim and E. Meištas, Baltic Astronomy, { } **editors: please complement**

O'Donoghue D., Kanaan A., Kleinman S. J., Krzesinski J., & Pritchet C. 2000, in The 5^{th} Whole Earth Telescope Workshop, eds. G. Vauclair and E. Meištas, Baltic Astronomy, 9, 375

Schuh S., Dreizler S., Deetjen J.L., Heber U., Geckeler R.D. 2000, in The 5^{th} Whole Earth Telescope Workshop, eds. G. Vauclair and E. Meištas, Baltic Astronomy, 9, 395

TRIPP documentation, `http://astro.uni-tuebingen.de/software/idl/aitlib/tripp/`

TRIPP pages and user's manual, at the time of writing maintained at `http://astro.uni-tuebingen.de/~schuh/tripp/`