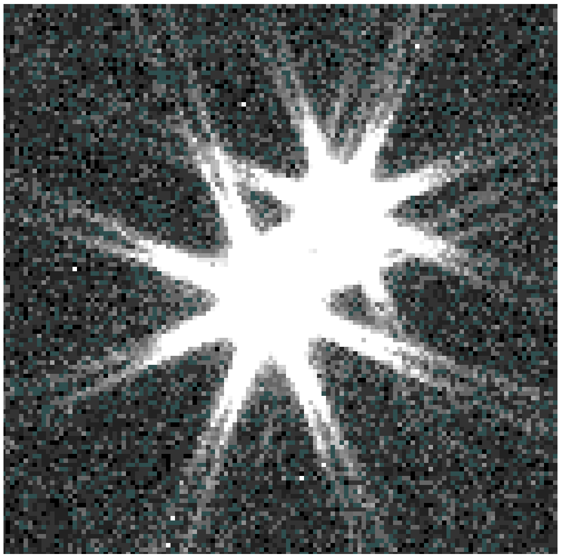
Guide to CTIOPI Parallax Reductions

by

Wei-Chun Jao, Adric R. Riedel, John P. Subasavage, Jr.,
Charlie T. Finch, & John C. Lurie



Last Updated: April 26, 2013

Contents

Chapter 1

Photometry

This chapter outlines the basic steps necessary to obtain VRI photometry for a pi star on the CTIOPI program. It is assumed that the photometry data have been reduced in that the standard stars have been fit and the transformation equations have been solved. Thus, all that is necessary is to "tag" the reference stars and pi star and apply the transformation equations to obtain absolute photometry. If this is your first time reducing CTIOPI data, you will need to install CTIOPI's IRAF package bundle called **redpi** and your Linux account must be a member of the recons group (see the system administrator or a RECONS executive member for help with this). Before starting a photometry reduction, you should be familiar with the *CTIOPI Photometry Reduction User Guide* by Wei-Chun Jao. For the reduction to run properly, you need to edit IRAF task parameters to match those given in § 3.3 of the photometry guide and in the figures at the end of that guide.

1.1 Photometry Reductions

In this section, we will walk through the steps necessary to obtain photometry for your parallax field, which will be needed later to determine corrections for differential color refraction (DCR) and absolute parallax. **This chapter may be skipped if you wish to use existing photometry from a previous reduction**, or if no photometry is available for the star for which a parallax is being determined (i.e., the pi star). However, neither of the aforementioned corrections can be made in that case

1. Pick out an object from the astrometry observing list that needs a final parallax reduction (or a preliminary reduction if that is your intention). The list is on the RECONS protected webpage.

   To check the photometry for the pi star, search for it on the 'CTIOPI photometry catalog' page located on the protected site under the Photometry heading. Find a night with good photometry for your pi star (refer to the data quality codes at the end of the catalog) and change directory to where the photometry is located. All CTIOPI photometry frames are located in `/nfs/recons3/phot.0.9m`.

2. Open a terminal window and type `recons` to log onto the RECONS machine. This will speed up the reductions because the images are located on this machine. Open an xgterm window with a scroll bar by typing `xgterm -sb &` (the & runs xgterm in the background and keeps the terminal window active for further use). In the xgterm window, change directory into your IRAF directory and type `cl` to start an IRAF session. In this IRAF window, type `!ds9 &` to open a graphics window that will be used to view and evaluate each frame.

3. From the IRAF window, change directory into the photometry science directory for your star (i.e., type `cd /nfs/recons3/phot.0.9m/YYYY.MMDD/science`) and make a directory that is named after the pi star by typing `mkdir` *pistarname*. We will use this directory for the photometry reductions of the reference stars and pi star(s).

4. Copy the pi star's photometry FITS files (found by looking at the photometry catalog webpage), `science.cfg` and `standard.coeff` into the pi star's new directory.

    (a) `science.cfg` **is not in the photometry directory**

    i. This will occur if the reduction of the science data for the entire night utilized Wei-Chun Jao's perl **evalfit** code rather than the IRAF task **evalfit**. Copy `science.cfg` from `/nfs/recons3/phot.0.9m/default`.

    (b) **files are named slightly differently (i.e.,** `scienceri.cfg`**)**

    i. In some cases, the science data (and the standards for that night) were reduced using a different color term to fit the transformation equations. This occurs when one filter frame for the pi star is problematic (i.e., missing, saturated, etc.) yet we want to obtain values for the two other filters. Emacs both `science`*XX*`.cfg` and `standard`*XX*`.coeff` and scroll down to the bottom of both where the transformation equations are. Verify that the same color [i.e., (mr-mi)] was used in both files. If not, these files are incompatible yet will run through the reduction but give you spurious values. Seek help from a RECONS certified photometry expert. If the same color was used, these files are compatible and thus, you can proceed with the reduction by copying the files into the *pistarname* directory.

5. In the pi star's new directory, make a list of all the photometry files by typing `ls *.fits > listfile`. We need to evaluate each of the potential reference stars as well as the pi star in each frame to check for saturation (peak counts $> \sim$ 65,000), cosmic rays within a $7''$ radius, and any aperture corrections that might be needed. This is done by typing `displayexam listfile` (IRAF). In the ds9 window, the following keystrokes are helpful: q- quits, e- contour plot, r- radial curve.

6. Select reference stars to tag in an order that is most convenient for tagging multiple frames. The pi star(s) *must* be the last star(s) tagged. The tagging sequence here will be the same as when the parallax frames are tagged and centroided. We recommend clicking on each star to circle it, so you know which ones they are.

NOTE ABOUT CHOOSING A GOOD REFERENCE FIELD
When picking a reference field, it is recommended to choose between 8 to 12 reference stars (less only if absolutely necessary) that have at least 1,000 peak counts and closely surround the pi star. It has been shown (by co-author C. Finch) that reference stars on the outer edges of the field are not as good as those near the center of the field. That said, some fields are very sparse such that it is not possible to select 8 reference stars. In these cases, select as many reference stars with ∼1,000 peak counts or more as you can. It may be beneficial to also include reference stars with as little as a few hundred peak counts, if they are close to the pi star. We recommend reviewing the astrometry frames before finalizing the reference field to ensure that the stars are on the chip and sufficiently exposed in most of the frames. Once a reference field is chosen, a digital pi reduction setup file will be created as discussed in § 2.1 step 5

7. If any cosmic rays are found within a 7″ radius of any of the reference stars or the pi star, use IRAF task **cosmicray** to remove them. Also, perform aperture corrections if needed.

   (a) **Cosmic ray correction** (see *CTIOPI Photometry User Guide* § 3.1)

      i. Load **crutil** in IRAF

      ii. Type `cosmicray` and input the file that needs correction. You can output to the same FITS file.

   (b) **Aperture correction** (see *CTIOPI Photometry User Guide* § 4.2)

8. For stars that do not need aperture corrections, we need to tag them using the standard 7″ radius aperture. This is done by running **apercorr** (IRAF) and selecting `option 3: tag stars with selected aperture`. When presented with parameters, scroll down to "photpar=/photometry parameters", type `:e` and make sure the only value in the aperture radius is 7. Also ensure that the parameters fwhmpsf and cbox are set to the correct values (see the Photometry Guide for details). Then type `Ctrl-D` until the mouse snaps to DS9, and start marking stars with `spacebar` (`Ctrl-Z` moves to the next frame).

9. Once all the reference stars and pi star are marked, you need to generate a file that lists which frame corresponds to which filter. This is done with **mkimsets** (IRAF), where you select the pi star's name to contain exactly 7 characters (required for the parallax pipeline, i.e., `lhs1749` or `gj0234x`, where the `'x'` is added as a place holder). Set output file to be named *pistarname*`.imsets`. Make sure the top two blank lines of the file are removed.

10. Next, a file that contains each star as well as relevant information for it (i.e., airmass, instrumental magnitude, etc.) needs to be generated. This is done with **mknobsfile** (IRAF) and setting the output file to be named *pistarname*`.obs`. Evaluate the output file and make sure everything looks good (i.e., no INDEF lines).

11. We are now ready to apply the transformation equations with the standard star fits to the reference stars and pi star. This is done by running **evalfit** (IRAF) and setting

the output file to be named *pistarname*`.phot`. Edit this file by deleting all of the text lines above the results. Also, delete the error columns leaving exactly two spaces between the photometry values. Leave a space at the beginning of each line for all stars numbered less than 10, even if there are less than ten lines. This way, when there are 10 or more stars total, the first column will appear to be right justified. Also, there should be two spaces in between each column.

Example:

```
 LTT1234-9  14.218  14.083  13.921
LTT1234-10  12.832  11.368  10.028
```

12. Move the file *pistarname*`.phot` to the current `pi.`*YYYY.MMDD* reduction subdirectory for the pi star, so that it can be used in the parallax reduction discussed in the next section. The parallax data is on `/nfs/recons4/CTIOPI/regions/`*RA-RA*`/`*pistarname*, where *RA-RA* is the right ascension range into which the pi stars are grouped. You may have to create the subdirectory now.

Chapter 2

Parallax Reductions

This chapter addresses the individual steps necessary to reduce a parallax. It does not go in detail about the underlying geometry and physics that are built into the pipeline. It is recommended that you read Wei-Chun Jao's Ph.D. thesis (2004) entitled *Discovery and Characterization of the Highest Proper Motion Stars* for a more comprehensive treatment of ground-based astrometric reductions.

2.1 Before you Begin (housekeeping)

Before we begin reductions, we need to prepare a directory for the reductions and go over a few good housekeeping tips.

1. First we need to change directory into that where the parallax frames are stored for the pi star we are reducing. Open a Linux terminal and navigate to the directory of the pi star. This directory is at `/nfs/recons4/CTIOPI/regions/`*RA-RA*`/`*pistarname*, where *RA-RA* is the right ascension range into which the pi stars are grouped. Inside this directory you should find all frames taken of that star; and if a previous reduction was done at least one folder called `pi.`*YYYY.MMDD* corresponding to the year and month the reduction was done, a `badframes.dir` folder, and a `badframes.`*pistarname* file.

2. If this is the star's first reduction, you should have already created a new directory called `pi.`*YYYY.MMDD* corresponding to the current day when you did the reference star photometry. If the star has previously been reduced, you can run **folder.processor** (Linux) from within the star-level folder. It will ask you for the name of the folder to copy things from, and the name of the new folder (generally use today's date), and copy over useful information like photometry, the star.settings file, taglists, and listfile.all to the new folder. Unless you plan on only tagging and centroiding new frames, you should run **rm -f *.list.*** in the new directory to delete the old **autotag** output.

3. From within the new subdirectory, run **star.processor** (Linux). It will ask you for the full name of the system (which should be a name in our astrometry catalog), and

a short 7-character name for the star. It will also ask for 2MASS coordinates and date of observation, and proper motion and position angle. If **star.processor** has already been run, those values will already be filled in. If not, these can usually be found on the CTIOPI master photometry page.

star.processor accomplishes the following steps:

(a) Syncs all *.fits files from the main directory.

(b) Copies a few standard template files necessary for the reduction from `../../../defaults/copythese.dir/* .`.

(c) Modifies the standard template files `default.sex` and `default.param` to replace the word 'default' with *pistarname* in their names, and everywhere therein.

(d) Creates a link to the **de405** ephemerides we will use to calculate the exact position of the Earth in its orbit for every frame of data

(e) Pulls entries for that star from the latest astrometry catalog in `/nfs/morgan/users/thenry/public_html/CTIOPI/protected.dir/` to `listfile.catalog` based on the full name you entered.

(f) Creates `listfile` (a list of all frames in the directory), `listfile.newframes` (a list of all new frames added to the directory since star.processor was last run), and creates/appends to `listfile.all`, which is used to hold notes about individual frames.

(g) Generates a file called `star.settings` which will store important information for use in later stages of the pipeline. Note that **star.processor** completely rewrites the file into the format subsequent programs are expecting, so if you drastically modify `star.settings` (not recommended) you may have to run **star.processor** to fix it.

4. **star.processor** will spit out a few statistics on frame counts which should add up properly, and print out any mismatches between the astrometry catalog and the current directory. Missing files can be found by using the **locate** command in the Linux terminal, and then copied into the current subdirectory. Extra files should only exist if you are a dedicated scientist and couldn't wait for the catalog to be updated before reducing your star (congratulations). In this case, you will have to add those new frames to `listfile.all` manually.

5. If you are making/using a new setup field: In the IRAF terminal run **setfield**. Select a decent frame by looking at at least 3 frames from different nights to find a good one that is not shifted. You will then be asked to examine the frames (with **imexam**, `e` and `r` to get contour and radial plots of stars, `q` to quit) to find the best reference stars. See § 1.1 step 6 for details about choosing a good reference field.
   Once that is done, come up with an order that is most convenient for tagging multiple frames, but the pi star *must* be the last star tagged. This sequence *must* be the same one used for photometry in Chapter 1, and will be used for tagging parallax frames in § 2.2. Follow the instructions and tag the expected reference stars (`spacebar`-tags, `ctrl`

`Z`-quits), then make sure the labels are appropriately placed. When the program exits, you will have a file called *pistarname*.setup#.ps, which must be printed and added to the parallax packet at the end.

6. **emacs listfile.all** (Linux). You should see any notes made during previous reductions, and **star.processor** should have automatically marked photometry frames in the wrong filter as bad, based on the astrometry catalog entry.

   Run **badnight** (Linux) to check if any of your star's data are from the nights listed on the 'Observing Nights with Issues' page under the Astrometry section of the RECONS protected website. If so, open the webpage and make sure your data are in the same filter as the problematic filter, i.e., 20050301 data is bad at V but good in R and I). If any of the issues are listed as "Not Fixable", mark those frames with an `X` in `listfile.all` and state why the frames have been discarded.

   Example:

   ```
   20000618.09.099.o.fits   Ref #1 off frame...okay!

   20040205.09.100.o.fits X Filters in Wrong Slots...BAD!

   20010128.09.061.o.fits
   20010128.09.062.o.fits X Wrong Filter...BAD!
   ```

7. Evaluate each frame and discard those with problems (e.g., saturations, distortions, FWHM > 6.0 pixels, or bad columns on your stars) by typing `displayexam listfile` (IRAF) (or, if you are updating a parallax, `displayexam listfile.newframes`). Generally you only need to look at the pi star, unless it is not the brightest star in your setup field. Mark an `'X'` next to the bad frames in `listfile.all` and describe why each was bad as in the above example. In the DS9 window, press `e` for a contour plot of the area under the mouse, `r` to get a radial plot of the area under the mouse, and `q` to advance to the next image. Save `listfile.all` when you are done.

8. The last thing we need to do is file the frames and create a list of frames to tag called `taglist`[1] To do this, first run **taglist.processor** (Linux). It will accomplish the following steps:

   (a) Crops out just the lines of `listfile.all` you marked with an `'X'` and appends them to a file called `badframes.`*pistarname* in the main directory

   (b) Moves the bad frames from both the subdirectory and the main directory to a folder called `badframes.dir` off the main directory.

   (c) Creates a new `listfile` that only has good frames with `ls *.fits > listfile`

   (d) Generates a file called `taglist` that lists one good frame from each night so as to tag the reference and pi stars (we assume the telescope pointing has not

[1] If you hit the `t` key while examining stars, you will have created extra `.fits` files which must be removed before this step.

changed during each night's acquisition of data for one pi star – usually 5 frames in succession). If a night has already been tagged (check for `*.list.*` files), it will not be added to `taglist`, so you will need to use the same reference star setup as the previous reduction.

(e) Writes the number of frames in the reduction to `star.settings`. From here on out, all frames will be referred to by the same numbers–anything that removes a frame from here on out will leave a gap in the numbering.

2.2 Parallax Reductions

This section will cover the steps necessary to complete a parallax reduction. The following procedures mention only a few of the most common problems that arise during a reduction. Undoubtedly, you will encounter additional problems and may need to seek the help of one of the RECONS astrometry experts. The majority of problems can be avoided by paying attention to detail. Avoid any distractions when performing a parallax reduction (at least for the first few dozen reductions).

1. The first thing we need to do is determine precisely where the pi star and the reference stars are. There are two supported ways to do this.

   (a) Autotagger: To use the autotagger, run **autotag** (IRAF). You will be asked for the name of a frame (any frame where all the reference stars are visible will do). Then, in the DS9 window, tag (with `spacebar`) the reference stars in a convenient sequence making sure to always tag the pi star last. This must be the same order you used in Section 1.1, or the one in the previous reduction's parallax packet if you want to reuse previously-reduced photometry. Check to make sure you tagged all the reference stars by typing `emacs autotag1.list`. When you are done, hit `ctrl-Z` to exit. The autotagger will then find the coordinates of every frame's center, and calculate the expected positions of the reference stars, and (taking proper motion into account) the pi star. At the end of this process, you should have one *.list.* file per frame.

      i. If autotag fails with the error message "Cannot open image (fliptempw.fits)", it may mean the plate scale is missing from the FITS header. Edit the parameters of **hedit** (IRAF) and set *add* to `yes`. Then type `hedit XXXX.fits` where `XXXX` is the frame with the missing plate scale. For *fields to be edited:* enter `XPIXSIZE` and for *value expression:* enter `0.401`. Press `enter` to confirm and then repeat these steps for `YPIXSIZE`. Finally, delete all files named *autotag.** except *autotag.sex* and *autotag.param* and re-run **autotag**.

      ii. Autotag may also fail if imwcs[2] cannot identify the field, in which case you will HAVE to use the manual tagging method. If you've already tried **autotag** and failed, re-run **taglist.processor** (Linux) first, to get an appropriate set of frames to manually retag.

[2] imwcs, http://tdc-www.harvard.edu/wcstools/imwcs/ is used by **autotag** (IRAF) and **wcsangle** (IRAF) to find frame centers

(b) Manual method: To do this, run **taglist** by typing `taglist` in the IRAF window (`spacebar`- tags, `ctrl z`- moves on to next frame). Tag the reference stars in a convenient sequence making sure to always tag the pi star last. This must be the same order you used in Section 1.1, or the one in the previous reduction's parallax packet if you want to reuse the previous reduction's photometry. Similarly, tag all of the frames using the same sequence (problems that are very difficult to identify will result if a reference star is tagged out of order). At the end of this process, you should have one `*.list.*` file per night.

When done, check to make sure you did not omit any reference stars by typing `wc -l *.list.*` in the linux window. The numbers in the first column should all be identical and equal to the total number of stars tagged per frames (reference stars + pi star).

2. Run **sss.var** (Linux). Run options 1-3 in sequential order (i.e., 1 first, 2 second, etc.).

   (a) **option 1: Sort *.list.***

   (b) **option 2: generate *.sex.* files for the first time and fix** [3]

   (c) **option 3: Run the SExtractor** program; choose **option a: run for the first time**.

3. Check that the **SExtractor**[4] program centroided on all of the stars. The final output of **option 3a** should be a list of 10 + the number of stars in each file (from `wc -l *.coord*`). All of the numbers in the first column should be the same. If not, refer to the troubleshooting steps below.

   (a) **Too many stars were centroided**

      i. Extra "stars" are usually detected because of cosmic rays and are often easily identifiable by comparing the FWHM column (column 7). Delete the line(s) in the `coord` file for that frame (i.e., *pistarname*`.coord.28`) that do(es) not correspond to a stellar object.

      ii. If there is a star in the vicinity of any of the reference stars or pi star (closer than 6 pixels), **SExtractor** will centroid it as well. They are usually identified as being very faint and several pixels off where you tagged. Delete the line(s) corresponding to this star(s) in the appropriate `coord` file.

   (b) **Not enough stars were centroided**

      i. Occasionally, a single frame in a night is shifted when compared to the remainder of the frames from that night because the pointing was adjusted during the sequence. This frame needs to be tagged again. Run the program **sex.ed** (Linux) to edit the .sex files, and type the numbers of frames that need to be retagged. **sex.ed** (Linux) will print the IRAF command you need

[3] If there are already a lot of properly tagged stars and/or a lot of extra listfiles from pointing shifts, you may want to run **option 5: add more *.sex.file** in place of **option 2**

[4] **SExtractor**, http://www.astromatic.net/software/sextractor is used to identify and accurately centroid all our frames

to run. Tagging is the same procedure (spacebar- tags, ctrl Z- moves on to next frame). When you hit *enter* in the Linux terminal, the `ASSOC_NAME` entry in the `.sex` file will be changed to point to the new taglist file you created (ex. *pistarname*.`list.20030405.09.075`)
Then run **sss.var option 1: sort *.list.*** and **option 3 a: Run the SExtractor program, for the first time** to rerun the SExtractor. DO NOT RUN **OPTION 2**.

A. If you are using manual tagging and the new suggested name already exists, you can simply hit `enter` when asked, and the `.sex` file's `ASSOC_NAME` entry will be changed appropriately. (when using the autotagger, existing lists are deleted)

B. If two frames from a particular night are bad, you can usually use one retagged file for both of them. Emacs the other `*.sex.*` files and manually change the `ASSOC_NAME` entries.

ii. One of the reference stars was off the edge of the chip in a frame (or sequence of frames from a given night). Add a dummy line at the bottom of the appropriate `coord` file by running **sss.var option 6: add placeholder line**, which will place a `9` everywhere in the line except for the last column, which corresponds to the sequential number of the reference star that was off the chip.

iii. If you are still having trouble, it may be helpful to expand the centroiding search radius. Open `shortname.sex`, where `shortname` is the shortened star name listed in `star.settings`. At the bottom of the file, change `ASSOC_RADIUS` to `10`. However, this is likely to result in the centroiding of some bogus "extra stars", which must be removed as descibed in the previous subsection.

4. Sort all of the `coord` file entries into one single file by running **sss.var option 4: sort the output files**. (Linux)

5. Now we need relevant timing information from the files' headers. This is accomplished by running **headmet** (IRAF). Ignore the error messages regarding humidity, temperature, and pressure. It will produce a file called *pistarname*.`hinfo`, and will run fixpix on all images. Sometimes coordinates need to be edited so that everything lines up. Certain "nights with issues" such as 20010223 have the wrong date in their headers; this can (and must) be corrected manually here by editing *pistarname*.`hinfo`.

6. Print a copy of `/nfs/recons4/CTIOPI/regions/defaults/table.pi.pdf` to fill out. This will be your data sheet for this reduction.

7. Evaluate whether image qualities are sufficient to be included in the reduction. This is done with **parallax.processor** (Linux).

   You will want to set up for the 0.9m, and only answer `yes` to DCR correction if you have a .phot file for this reference setup as outlined in Chapter 1. Including faint reference stars is not recommended unless the field has few bright stars and many faint ones. Finally, answer `yes` to both `coord` (for parallax) and `var` (for relative variability) files.

You will then be asked to throw out frames with Hour Angles greater than 120 minutes and frames with fewer than 5 reference stars. For consistent results, answer `yes` to all. The system will also automatically throw out frames with no pi star or PSFs that are more than 20% elliptical. (60% if you included faint reference stars).

Finally, it will print a list of frames of sufficient quality to be used as trail plates (i.e., HA < 4 min, all reference stars are included) out of the *pistarname*`.pifonly` file. Select the best trail plate (lowest HA, low FWHM) and write its number on the data sheet. Cross out any bad frames on the data sheet, and fill in the rest of the information from the last output of **parallax.processor**. Also circle the trail plate number, and mark a line to the right of the number corresponding to the total number of frames in the listfile. A sample completed data sheet is provided at the end of this guide.

The program **parallax.processor** runs **genpif**, which gathers all the information about the reference stars' positions, observation times, and positions of the Earth into something to which we can fit a parallax. It reads the outputs from **genpif**, and removes frames from the reductions. It also automatically modifies `ctio.model` and `varia.model` to use the trail plate and pi star you have selected.

8. Adjust for the alignment of the CCD on the telescope vs. true north-south for the trail plate, and the plate scaling.

   This is done by running **wcsangle** (IRAF). A file named `wcs.cat` will pop up in an emacs window. Exit this window and a cursor will appear over the DS9 window. The stars from the file will be marked. If a bad column is marked, then **emacs** `wcs.cat` and delete the corresponding line(s). Then type '`q`' in the DS9 window to quit. Two options are available for comparison; but only 2MASS (**option 2**) is working as of 2009.0806. The output file `angle.out` will contain the plate scale and the rotation angle. Both of these should be written on the data sheet

9. Apply the rotation angle to the trail plate by running **pltrotate** (Linux). It will apply the rotation automatically, and runs even if **wcsangle** failed to find any rotation.

2.3 Solve the parallax

You are now ready to solve for the parallax and proper motion with **GaussFit**, using the **gauss.uty** (Linux) task.

1. To start, prepare the **gaussfit** environment by running **gauss.uty** (Linux) and then choosing **option 1: GaussFit utility — generate setPar, env and etc**.

2. If you are reducing both members of a binary star system, now is the time to choose which one to solve for. Remove the other one by running **gauss.uty, option 2: Remove high residual stars for all the plates** and giving its number.

3. Run **gauss.uty, option 3: Run gaussfit**. It will ask if you want to change the pi star.

   If you have a single star the answer is `no`, so just hit enter.

   If you are reducing a binary star, this is your chance to change the pi star to solve for and/or the full name of the star (ex. LHS3001AB -> LHS3001A). The full name will be used in the output of **piplot.pro**.

   GaussFit will then solve for the parallax and proper motion. It should take less than 10 iterations to converge.

4. In another terminal window, change directory into the pi star's reduction directory (ie, *RA-RA*/*pistarname*/ `pi.`*YYYY.MMDD*. Start IDL by running **idl70** (Linux), then plot the results with **piplot** (IDL). **piplot** will generate a parallax printout, nightly mean plot, and a complete line for this reduction to include on the final results page.

   piplot will report the mean exposure time for the frames used in the reduction, and statistics about the reference and pi star residuals. The output should look someting like this:

   ```
   Mean exposure time:  58 seconds
   Number of Frames:  100
   Stars in this reduct.  (REF+PI): 10

   Reference Star Residuals
   REF  ResX   ResY   Mean
   1  1.000  0.700  0.850
   2  0.800  1.000  0.900
   3  1.000  1.000  1.000
   4  0.900  0.800  0.850
   5  1.000  1.000  1.000
   6  1.400  1.600  1.500 <<
   7  1.000  0.900  0.950
   8
   9  0.900  1.000  0.950
   10  1.000  1.000  1.000
   11  1.000  1.000  1.000
   ```

   The middle two columns in the table are how often each star has the highest x and y residual, versus if you simply take the number of frames divided by the number of stars. That is, a value greater than one means the star has the highest residual more often than average. In the above example, there are 100 frames and 11 reference stars, one of which has been dropped (denoted by the blank line). Each of the remaining stars should have the highest residual in 10 frames. In this case, star 6 has the highest x residual in 14 frames and the highest y residual in 16 frames. The last column is simply the average of the first two. The program also flags any averages greater than 1.5.

5. Review the `piPar` and `muPar` files. They contain the raw calculated parallaxes and proper motions (in PMra/PMdec) for all the stars in the reduction. In particular, if any of the reference stars is nearby according to the `piPar` file, or has a high proper motion according to `muPar`, it should be discarded from the reduction as bad and (probably) reduced on its own.

6. Review the parallax statistics printout itself by typing `gv YYYYMMDD.`*pistarname*`.stats.plot.ps`, and filp it to landscape.

7. If you have a *pistarname*`.phot` file, do a reduction to absolute now:

   (a) Create a photometry file with only the reference stars in it (i.e., no pi star(s)) by typing `cp` *pistarname*`.phot` *pistarname*`.refphot`. Edit the *pistarname*`.refphot` file and remove the line(s) corresponding to the pi star(s). Also, if any reference stars were bad and were removed from the reduction, delete those lines as well.

   (b) Calculate photometric distance estimates for the reference stars used in the parallax reduction by running **refdist** (Linux) The average photometric parallax estimate should be less than 3 mas (i.e., 0.00300); if not, check to see if a single reference star (usually a red giant) has a large parallax estimate skewing the average by running **emacs *.refdistance**. If any of the stars are predicted to be nearby, it is recommended to try reducing THAT star as the pi star (throwing out the actual pi star) to see if it's actually a nearby star; see below for instructions. If it turns out to be nearby, remove its line from *pistarname*`.refphot` and rerun **refdist** (Linux). If it is NOT, we will adopt a generic correction prior to publication.

   Check the output of the `piPar` file (actual trig reduction) to ensure the reference star is not actually nearby. If it is nearby (i.e., within ≈100 pc), you should make a note of it and come back later to specifically reduce that star for parallax.)

   (c) If you had to throw out any stars because of the previous step, you will need to re-reduce the parallax with the star removed (see directions below). If you did not have to throw out anything, just rerun **piplot** to redo the statistics plot with the correction to absolute.

8. If you are doing a binary system, run **gauss.uty, option 4: Re-copy files for another pi reduction** and go back to the beginning of this section. This time you will throw out the *other* pi star. Remember to change the name appropriately when running **gauss.uty, option 3**. You will not need to redo the reduction to absolute, as long as you're using the same reference stars for both components.

   (a) **If a bad star is found**

      i. Delete all coord files and replace by running **gauss.uty**, **option 4: Re-copy files for another pi reduction** (Linux) or type `rm -f coord.*`, then `cp coordbak/* .`

ii. Re-run **gauss.uty, option 1: GaussFit utility — generate setPar, env and etc**, then **option 2: Remove high residual stars for all the plates** and input the number of the bad reference star. This will have to be done after each time you run option 1.

iii. Cross out the reference star's number on the datasheet.

iv. You are now ready to resume the parallax reduction at step 3. If you already did a reduction to absolute, you must edit *pistarname*.`refphot` to remove this bad star, and rerun **refdist**.

v. Should something go very badly wrong with the datafiles, remove all coord and var files with **rm -rf coordbak coord.* *.coord varbak var.* *.var**, and go back to run **parallax.processor** (§ 2.2 step 7)

(b) **If a bad frame is found**

i. Any significant outliers that skew parallax results (or `nV` filter frames) can be semi-permanently removed from the reduction with **gauss.uty, option 5: 'Permanently' toss bad frames (until parallax.processor is rerun)**. It will ask you for a list of frame numbers to remove, then recopy files for a new reduction and then remove the specified frames. You will then need to rerun **gauss.uty option 1**, and then **option 3**.

9. Print out the results page(s) (i.e., `YYYYMMDD.`*pistarname*`.stats.plot.ps`) by typing `lpr` *filenames*

2.4 Variability Analysis

In this section, we perform a variability analysis of all the parallax frames. This is done by using instrumental magnitudes scaled to a common zero point and then comparing the reference field values to the pi star in search for variability. This method can be problematic (especially if any of the reference stars are faint or if the pi star has a close optical or physical companion and the seeing varies) but, in general, we can determine if a star is variable on the ≈1% level or larger.

1. Generate an environment file for **gaussfit** by running **var.uty** (Linux) and selecting **option 1: GaussFit utility – generate setPar, varenv and etc**

2. Calculate the variability with **var.uty, option 3: Run gaussfit**

3. Evaluate the output by using **varplot** (IDL) to plot each star and interactively select stars that may be variable. In the IDL environment used in § 2.3 step 4, type `colorset` and then `varplot`. A plot will appear with the stars shown and 1σ and 2σ dotted lines. Look for stars that are above the 1σ line. Right-click on the star to obtain its ID. Note: if stars to the right of the plot (i.e., faint) are just above the 1σ line, it's usually due to their faintness rather than intrinsic variability such that they do not need to be removed.

4. Right click on the pi star, then right click again on the plot. IDL will ask you if you want to print the result to a ps file. Say “yes”, and then “no” to changing the plotting range. Open the ps file you just created using gv, and right click on the number “1” next to the scroll bar so it turns red. Then select “Print Marked”. Finally, remove the ps file.

5. Delete var files and replace – run **var.uty, option 4: Re-copy files for another variability analysis** and then run **var.uty option 1**.

6. Run **var.uty, option 2: Remove a possible variable star from all the plates** and remove all variable stars from step 3. Also remove the pi star, even if it is not variable. On the printout you made in step 4, put a box around any reference stars you plan to remove, as well as the pi star. Also note the ID number of the stars you remove, and any other relevant remarks. An example printout is provided at the end of this manual.

7. Re-solve variability with **var.uty, option 3**

8. Run **var.uty, option 6: Generate a data file on one variable star** and enter the ID of the pi star.

9. In the window running IDL, execute the program **problemstar** (IDL). Print the output file `YYYYMMDD.`*pistarname*`.var.plot.ps` from **problemstar** (IDL) to include in the reduction packet. The plotting range should be changed to -0.2 to 0.2, and the object name is the same `pistarname` you’ve been using.

10. **If a bad frame is found**: Any significant outliers that skew variability results can be semi-permanently removed from the reduction with **var.uty, option 5: ’Permanently’ toss bad frames (until parallax.processor is rerun)**. It will recopy files for a new reduction and then remove the specified frames.

2.5 Cataloging Results

With a completed parallax in hand, it is necessary to record the results to be posted on the RECONS protected page.

1. Emacs the final results file found in `/nfs/morgan/users/thenry/public_html/CTIOPI/protected.dir/` called `results.0.9.final.YYYY.MMDD`. If the pi star has been reduced previously and *not* published, update the information already posted. If the star has been previously published, add your new line to the section titled “Updated Unpublished Parallaxes for Previously Published Targets”).

   `piplot.pro` has generated the appropriate line, written to `YYYYMMDD.`*pistarname*`.parallax`. If you have done the variability analysis, fill in that column. Do not mark an `*` to the right of the absolute parallax column. This will be marked once the results are recorded on the astrometry observing list (usually performed by Todd Henry).

2. Remove all of the FITS files from the `pi.`*YYYY.MMDD* directory to avoid filling the diskspace unnecessarily. **Be careful not to delete the data in the main directory!**

3. Ensure that the reduction directory and all its contents have group read and write permissions. That way other RECONS members can access the results.

4. Prepare the parallax results packet in order of (1) parallax results printout, (2) nightly means printout, (3) data sheet, (4) parallax results printouts from older reductions, (5) variability printout (discard older variability results), and (6) field setup printouts, most recent at the back.

   (a) If a parallax for this star has already been published (will be marked on the final results page), leave that packet intact and add yours to the front. (Although the variability goes just in front of the field setup printouts)

   (b) If there are no digital versions of previous setup fields, rerun setfield and replace the handwritten version(s) in the parallax packet. Previous field setups go in front of the current one.

5. Staple, hole punch (opposite the side where the pi star name is listed on the results printout) and file the packet in the correct binder (based on pi star name) in order of pi star name.

You are DONE! Start another parallax!

2.6 Cheat sheet

1. (photometry)
2. Examine the files
   (a) (folder.processor [Linux])
   (b) star.processor [Linux]
   (c) (setfield [IRAF])
   (d) badnight [Linux]
   (e) displayexam listfile [IRAF]
   (f) taglist.processor [Linux]
3. Set up the parallax
   (a) taglist [IRAF] or autotag [IRAF]
   (b) sss.var (& sex.ed) [Linux]
   (c) headmet [IRAF]
   (d) parallax.processor [Linux]
   (e) fill out table.pi.pdf
   (f) wcsangle [IRAF]
   (g) pltrotate [Linux]
4. Solve the parallax
   (a) gauss.uty [Linux]
   (b) (refdist [Linux])
   (c) piplot.pro [IDL]
5. Solve the variability
   (a) var.uty [Linux]
   (b) varplot.pro [IDL]
   (c) problemstar.pro [IDL]
6. Cleanup
   (a) remove FITS files from subdirectory
   (b) check directory group permissions
   (c) update final results page
   (d) prepare and file packet

Object: OBJECT NAME　　Who: YOUR INITIALS　　Date: DATE OF REDUCTION

numbers in boxes below correspond to lines in "listfile"

1	2	3	4	5	~~6~~	~~7~~	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	[✓ 42]	43	44	45	46	47	48	49	50
51	52	53	54	55	56	57	58	59	60	~~61~~	62	63	64	65	66	67	68	69	70	71	72	73	74	75
76	77	78	79	80	81	82	83	84	85	86	87	88	89	90 //	91	92	93	94	95	96	97	98	99	100
101	102	103	104	105	106	107	108	109	110	111	112	113	114	115	116	117	118	119	120	121	122	123	124	125
126	127	128	129	130	131	132	133	134	135	136	137	138	139	140	141	142	143	144	145	146	147	148	149	150
151	152	153	154	155	156	157	158	159	160	161	162	163	164	165	166	167	168	169	170	171	172	173	174	175
176	177	178	179	180	181	182	183	184	185	186	187	188	189	190	191	192	193	194	195	196	197	198	199	200
201	202	203	204	205	206	207	208	209	210	211	212	213	214	215	216	217	218	219	220	221	222	223	224	225
226	227	228	229	230	231	232	233	234	235	236	237	238	239	240	241	242	243	244	245	246	247	248	249	250
251	252	253	254	255	256	257	258	259	260	261	262	263	264	265	266	267	268	269	270	271	272	273	274	275
276	277	278	279	280	281	282	283	284	285	286	287	288	289	290	291	292	293	294	295	296	297	298	299	300
301	302	303	304	305	306	307	308	309	310	311	312	313	314	315	316	317	318	319	320	321	322	323	324	325
326	327	328	329	330	331	332	333	334	335	336	337	338	339	340	341	342	343	344	345	346	347	348	349	350
351	352	353	354	355	356	357	358	359	360	361	362	363	364	365	366	367	368	369	370	371	372	373	374	375
376	377	378	379	380	381	382	383	384	385	386	387	388	389	390	391	392	393	394	395	396	397	398	399	400
401	402	403	404	405	406	407	408	409	410	411	412	413	414	415	416	417	418	419	420	421	422	423	424	425
426	427	428	429	430	431	432	433	434	435	436	437	438	439	440	441	442	443	444	445	446	447	448	449	450
451	452	453	454	455	456	457	458	459	460	461	462	463	464	465	466	467	468	469	470	471	472	473	474	475
476	477	478	479	480	481	482	483	484	485	486	487	488	489	490	491	492	493	494	495	496	497	498	499	500

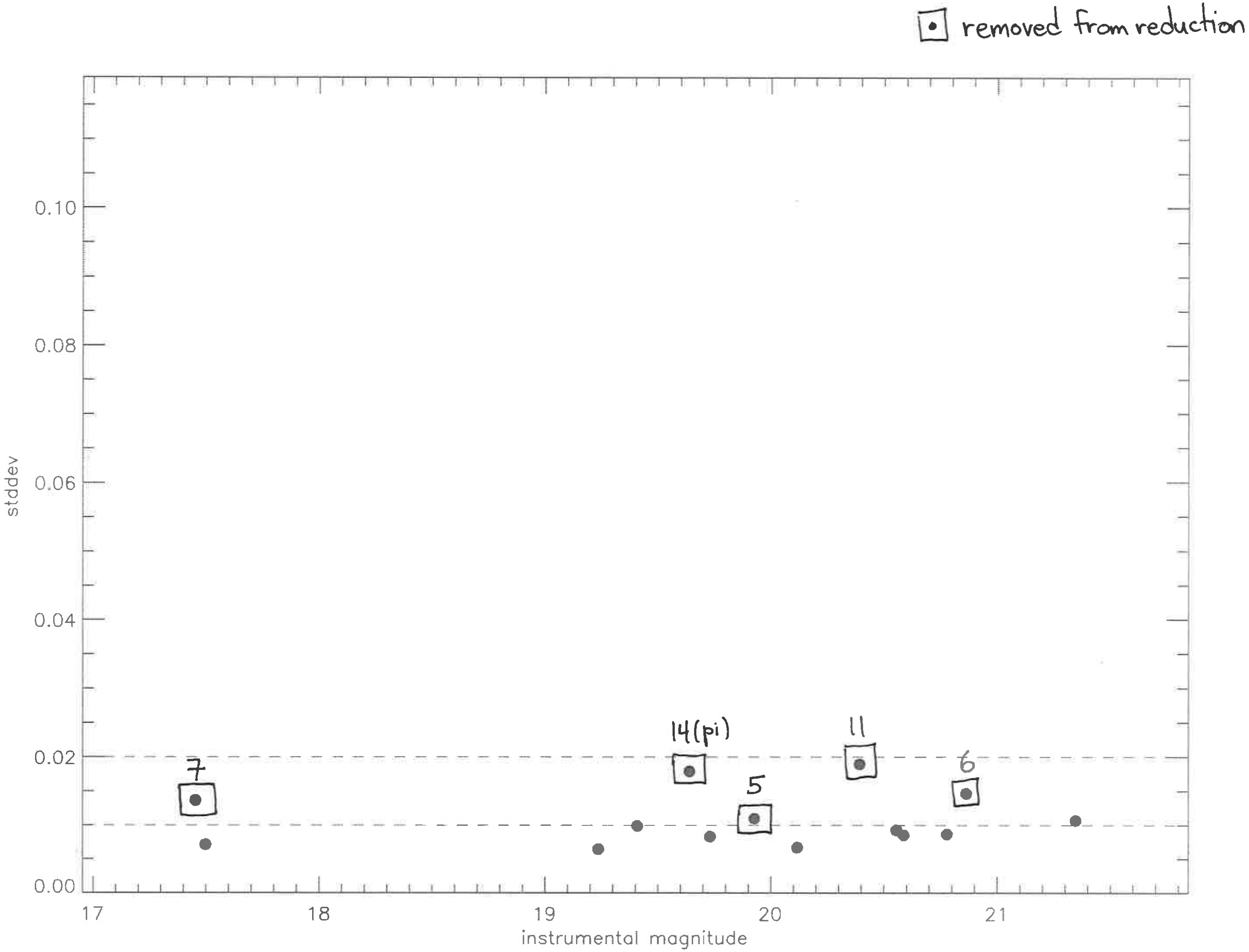
circle pi star / slash bad ref star(s): 1 2 3 ~~4~~ 5 6 7 8 9 [✓ 10] 11 12 13 14 15 16 17 18 19 20

Trail Plate ID#: 42　　File Name: 20101116.09.022.0.fits　　Rotation Angle: 0.1234 deg　Scale: 0.4XXX"/pix

frames in pi filter: 102　　# frames in listfile: 90　　# frames used for reduction: 87

average integration time: 120 sec

NOTES: Frames 6 and 7 have HA > 120 min and were removed
Frame 61 contains less than 5 ref. stars and was removed
Ref. star 4 is faint, and has high residuals. Removed

□ removed from reduction
0.10
0.08
0.06
0.04
0.02
0.00
stddev
17
18
19
20
21
instrumental magnitude
7
14(pi)
5
11
6