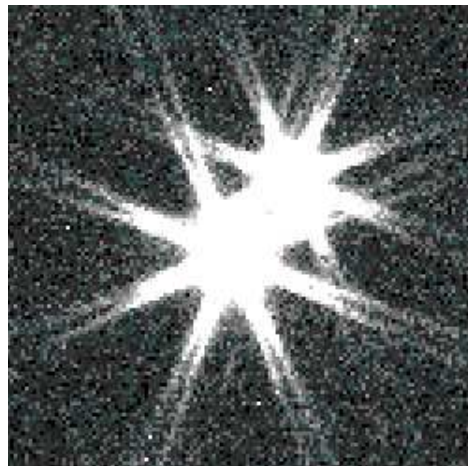


# GUIDE TO CTIOPI PARALLAX REDUCTION

by

ADRIC R. RIEDEL & JOHN P. SUBASAVAGE, JR. & CHARLIE T. FINCH



## TABLE OF CONTENTS

<b>1</b>	<b>PHOTOMETRY</b> . . . . .	<b>1</b>
1.1	Photometry Reductions . . . . .	1
<b>2</b>	<b>PARALLAX REDUCTIONS</b> . . . . .	<b>5</b>
2.1	Before you Begin (housekeeping) . . . . .	5
2.2	Parallax Reductions . . . . .	7
2.3	Solve the parallax . . . . .	11
2.4	Variability Analysis . . . . .	13
2.5	Cataloging Results . . . . .	15
2.6	Cheat sheet . . . . .	16

# 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 recon group (see the system administrator or a RECONS executive member for help with this). For a complete treatment of CTIOPI photometry reductions, please refer to the *CTIOPI Photometry Reduction User Guide* by Wei-Chun Jao (2003).

### 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 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 these cases.

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 website (<http://www.RECONS.org>).

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 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 `c1` 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 either by typing `imhead *.fits` or 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 `scienceXX.cfg` and `standardXX.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 `files *.fits > listfile`. We need to evaluate each of the 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`. In the ds9 window, the following keystrokes are helpful: q- quits, n- next image, 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 printed. 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  $\sim 1,000$  peak counts or more as you can. Once a reference field is chosen, a digital pi reduction setup file will be created as discussed in 2.1

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**
    - 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*)
8. For stars that do not need aperture corrections, we need to tag them using the standard  $7''$  radius aperture. This is done by typing **apercorr** and selecting option 3: **tag stars with selected aperture**
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 by typing **mkimsets**, 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*.
10. Next, a file that contains each star as well as relevant information for it (i.e., air-mass, instrumental magnitude, etc.) needs to be generated. This is done by typing **mknobsfile** 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 typing **evalfit** 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. 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` 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. 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. Create a new folder called `pi.YYYY.MMDD` corresponding to the current day, or copy the previous reduction's folder to make a new one: `cp pi.yyyy.mmdd pi.YYYY.MMDD`. If you copy the folder, remove all intermediate pipeline files: `rm -rf coord.* *.coord *.coord* coordbak/ var.* *.var varbak/` and sometimes `*.sex.*`.
3. From within the subdirectory, run `star.processor`. 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; these can usually be found on the CTIOPI master photometry page. If `star.processor` has already been run, those values will already be filled in.

`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/CTI0PI/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 digital setup field: Close any open ds9 windows, and run **setfield** in the IRAF terminal. Select a decent frame by looking at at least 3 frames from different nights, and then evaluate possible reference stars. Follow the instructions and tag the expected reference stars, 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.

#### **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  $\sim 1,000$  peak counts or more as you can.

Number the stars in the order that is most convenient for tagging multiple frames but the PI star *must* be the last star tagged. This same sequence will be used for tagging parallax frames.

6. **emacs listfile.all.** 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.

Check the RECONS website in the Astrometry section under ‘Final Results’. Open the page ‘Observing Nights with Issues’ and check to see if any of your star’s data are from the nights listed as “Not Fixable” (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 exist, 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 (i.e., distortions,  $\text{FWHM} > 6.0$  pixels, bad columns on your stars) by typing `displayexam listfile` (or, if you are updating a parallax, `displayexam listfile.newframes`) in the IRAF window. Mark an ‘**X**’ next to the bad frames in **listfile.all** and describe why each was bad as in the above example. Save **listfile.all** when you are done.
8. If you hit the ‘T’ key while examining stars, you will have created extra `.fits` files which must be removed before the next step.

## 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. To do this, first run **taglist.processor**. It will accomplish the following steps:
  - (a) Crops out just the lines of `listfile.all` you marked with an 'X' and append 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) Copies all remaining files back to the main directory (anything that was previously missing that you **located** will thus be saved for future reductions)
  - (d) Creates a new `listfile` that only has good frames with `ls *.fits > listfile`
  - (e) 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 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.
  - (f) Writes the number of frames in the reduction to `star.settings`.
2. Run the IRAF process **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 1.1, or the one in the previous reduction's parallax packet. 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).

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).

3. Now we need relevant timing information from the files' headers. This is accomplished by running **headmet** in IRAF. 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.

4. Run **sss.var** in the linux window by typing **sss.var**. 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** <sup>1</sup>

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

5. Check that the **SExtractor** 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., **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** in the terminal window to edit the **.sex** files, and type the numbers of frames that need to be retagged. **sex.ed** will print the IRAF command you need 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*)

---

<sup>1</sup>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**

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 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.
  - B. If two frames from a particular night are bad, you can usually use one re-tagged 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.
6. Sort all of the `coord` file entries into one single file by running **sss.var option 4: sort the output files**.
  7. Print a copy of `/nfs/recons4/CTIOPI/regions/defaults/table.pi.ps` to fill out. This will be your data sheet for this reduction.
  8. Evaluate whether image qualities are sufficient to be included in the reduction. This is done by typing **parallax.processor** in the terminal window.

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 § 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 both. The system will also automatically throw out frames with no pi star or PSFs that are more than 20% elliptical.

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 `pisturname.pifonly` file. Select the best trail plate 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 copy any relevant notes from the previous reduction's datasheet; we will only be keeping this current one.

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.

9. Adjust for the alignment of the CCD on the telescope vs. true north-south for the trail plate.

This is done by running **wcsangle** in the IRAF window (i.e., type **wcsangle**). A file will pop up, exit and place the cursor over the ds9 window. The stars from the file will be marked. If no bad columns are marked, type 'q' to quit. Two options are available for comparison; GSC (**option 1**) and 2MASS (**option 2**). As of 2009.0806, 2MASS is the only working option. The output file **angle.out** will contain the plate scale and the rotation angle. Both of these should be written on the data sheet (write the plate scale just above "Trail Plate Rotation Angle:").

10. Apply the rotation angle to the trail plate by running **pltrotate** in the Linux terminal. 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**.

1. To start, prepare the **gaussfit** environment by running **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 do now. Remove the other one by running **option 2: Remove high residual stars for all the plates.**
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**, or just hit enter (N is the default).

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 **parallax.stats.pro** and **nightlyplot.pro**

Gaussfit will then solve for the parallax and proper motion.

4. In another terminal window, change directory into the PI star's reduction directory (ie, *RA-RA/pistarname/ pi.YYYY.MMDD*). Run IDL to plot the results by typing `idl170`, then `.r parallax.stats.pro` (run this twice to compile **angleerr**), then `piplot`. **piplot** will generate a parallax printout and a complete line for this reduction.
5. Review the output from **gaussfit** to ensure no reference stars were problematic by typing `ls *.stats.*`. Emacs the file *pistarname.stats.residual.out* and look for any star number that shows up as the highest residual on an inordinate number of frames (this is fairly subjective but if there is really a problem star, it is usually pretty obvious).
6. Review the parallax statistics printout itself by typing `gv YYYYMMDD.pistarname.stats.plot.ps`. Flip it with `flip landscape` on the top bar.
7. If the parallax looks good, generate a nightly means plot for the star. In the same IDL window run `.r nightlyplot.pro` and then `nightlyplot`.
8. 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** in the linux window. 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. If so, it is best to remove that reference star from the *pistarname.refphot* file, and rerun **refdist**.  
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  $\approx 100$  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.
9. 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: Recopy files for another pi reduction** 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 8)

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

- i. Any significant outliers that skew parallax results can be semi-permanently removed from the reduction with **gauss.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.

10. 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  $\approx 1\%$  level or larger.

1. Generate an environment file for **gaussfit** by running **var.uty** and selecting **option 1: GaussFit utility – generate setPar, varenv and etc** in the linux window.

2. Run **gaussfit** with **option 3: Run gaussfit**
3. Evaluate the output by using the IDL program **varia.stats.plot.pro** to plot each star and interactively select stars that may be variable. In the IDL environment used in § 2.3 step 4, type `.r varia.stats.plot.pro`, then `varplot`. A plot will appear with the stars shown and  $1\sigma$  and  $2\sigma$  dotted lines. Look for stars that are above the  $1\sigma$  line. Right-click on the star to obtain its ID. Perform the steps outlined below to remove the variable star (regardless of whether it's the PI star or a reference star). Note, if stars to the right of the plot (i.e., faint) are just above the  $1\sigma$  line, it's usually due to their faintness rather than intrinsic variability such that they do not need to be removed.
  - (a) **a reference star is found to be variable.**
    - i. Delete var files and replace – run **var.uty option 4: Re-copy files for another variability analysis** or type `rm -f var.*`, then `cp varbak/* .`
    - ii. Run **var.uty option 1**, then **var.uty option 3: Remove a possible variable star from all the plates** to remove the variable star.
    - iii. Rerun **gaussfit** with **var.uty option 3**
    - iv. Rerun **varplot** as in step 3
    - v. Generate a printout of the instrumental magnitude of the PI star from all of the parallax frames. Find the PI star and left-click on the point. Then right-click anywhere in the plot area and answer ‘y’ to send figure to a plot and ‘n’ to keep the plotting range unchanged. Print the output file `YYYYMMDD.pistarname.var.plot.ps` doublesided.
  - (b) **the PI star is found to be variable.**
    - i. Follow steps 1 — 3 from step 3(a) above to remove the variable PI star.
    - ii. Change directory into varbak (i.e., `cd varbak`) and run **var.uty option 6: Generate a data file on one variable star.**
    - iii. In the window running IDL, execute the program **problemstar** – type `.run problem.star.pro`, then `problemstar`.
    - iv. Print the output file `YYYYMMDD.pistarname.var.plot.ps` from **problemstar** to include in the reduction packet.
  - (c) **If a bad frame is found**
    - i. 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. Otherwise, replace the line for this star in the appropriate place (i.e., preliminary or final sorted by sample and RA within each sample).

`parallax.stats.pro` has generated the appropriate line, written to `YYYYMMDD.pisturname.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 disk space unnecessarily. **Be careful not to delete the data in the main directory!**
3. Prepare the parallax results packet in order of (1) parallax results printout, (2) nightly means printout, (parallax results printouts from older reductions), (3) variability printout, (4) data sheet, and (5) field setup printout from front to 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.
  - (b) If this is merely an updated parallax, unstaple the packet and replace the previous data sheet with your new updated one.
  - (c) If there are no digital versions of previous setup fields, run `rerun setfield` and replace the handwritten version(s) in the parallax packet. Previous field setups go in front of the current one.
4. Staple, hole punch (opposite the side where the PI 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) star.processor
  - (b) displayexam listfile [IRAF]
3. Set up the parallax
  - (a) taglist.processor
  - (b) (setfield [IRAF])
  - (c) (fixpix [IRAF])
  - (d) taglist [IRAF]
  - (e) headmet [IRAF]
  - (f) sss.var (& sex.ed)
  - (g) parallax.processor
  - (h) fill out table.pi.ps
  - (i) wcsangle [IRAF]
  - (j) pltrotate
4. Solve the parallax
  - (a) gauss.uty
  - (b) (refdist)
  - (c) parallax.stats.pro [IDL]
  - (d) nightlyplot.pro [IDL]
5. Solve the variability
  - (a) var.uty
  - (b) varia.stats.pro [IDL]
  - (c) (problemstar.pro [IDL])
6. Cleanup
  - (a) remove fits files from subdirectory
  - (b) update final results page
  - (c) prepare and file packet