H_α Filter Analysis and Calibration using PSF photometry on h and χ Persei

Andrew Hernandez

A senior thesis submitted to the faculty of
Brigham Young University
in partial fulfillment of the requirements for the degree of

Bachelor of Science

Eric Hintz, Advisor

Department of Physics and Astronomy

Brigham Young University

Copyright © 2018 Andrew Hernandez

All Rights Reserved

ABSTRACT

H_α Filter Analysis and Calibration using PSF photometry on h and χ Persei

Andrew Hernandez
Department of Physics and Astronomy, BYU
Bachelor of Science

Magnitudes of stars are measured from the energy we measure (apparent) and the total energy output over the entire surface of the star (absolute). In this work we find apparent magnitudes for specified wavelengths of the double cluster h and χ Persei. Apparent magnitudes are attained using point spread function photometry, which is utilized in order to help separate closely spaced stars existing in the clusters. Color-color diagrams are shown detailing the physical properties of h and χ Persei. These are done using the H_α index, described herein, with the H_α magnitude. From these diagrams we can recreate a Hertzsprung-Russel diagram with the addition of Be type stars being easily identifiable. A second, unknown group is discussed. An analysis of the H_α index with the H_β index shows how H_α emission continues beyond H_β emission for emission type objects. Using this H_α H_β index plot emission objects are easily identifiable.

Keywords: stars: clusters, filter calibration

ACKNOWLEDGMENTS

Dr. Eric Hintz for his patience and guidance throughout my long undergraduate career,

My astronomy professors for their help and support,

and my Wife for her love and encouragement.

Contents

List of Figures

Chapter 1

Introduction

1.1 Young Clusters

In our search for understanding of the universe around us and the elements, stellar study is of great importance. Though our own star, the Sun, serves as a source for much knowledge, there are other types of stars that provide different information. Massive stars burn hotter and brighter, creating heavier elements, while less massive more dim stars, similar to our own star teach us about our our origins. Young stellar clusters are groups of stars that formed from the same material and at the same time but cover a large range of stellar masses and therefore stellar spectral types.

Every star is categorized by its spectral type. Spectral type is a characterization system based on the spectrum of light a star emits and is directly related to the mass and temperature of a star. Starting from more massive, hotter stars and going to the less massive, cooler stars the spectral type nomenclature is O, B, A, F, G, K, M type stars. For reference, our own sun is a G type star and O type stars can reach 10 - 100 times the mass of our own star and burn 10 times hotter.

Young stellar clusters serve as robust objects of study as they contain stars from all spectral types. These young clusters have formed fairly recently on the cosmic timescale, having ages of 10^7

Figure 1.1 From left to right: *h* and χ Persei, also known as NGC 869 and 884 respectively

to 10^8 years. This "short" lifetime allows for the larger O and B spectral type stars, which normally have shorter lives, to be observed. Here smaller stars have already had sufficient time to begin formation and are thus prevalent as well. Due to this complete sample of stars, we have chosen to study the young double cluster *h* and χ Persei.

The double cluster *h* and χ Persei (see Fig 1.1) are young stellar clusters with ages $\sim 13-14$ million years (Myrs) (Boyer et al. 2012). The two have both been shown to contain stars ranging from spectral type O to M. This being the case, the pair have been a subject of major study for many years. Analysis of both clusters has been done with a wide variety of photometric and spectroscopic data. Currie et al. (2010) have done photometric imaging in the V and I filters where they define cluster properties, membership, and the intrinsic colors and temperatures of stars. Boyer et al. (2012) did a spectroscopic analysis of the clusters. They found more stellar parameters from these clusters such as rotational velocities, effective temperatures, and surface gravity. Each group confirmed that, though there exist a wide range of spectral type stars, *h* and χ Persei are predominately populated with B spectral type stars.

Of the B type stars therein, there have also been found many Be stars. Be stars are B type stars with prominent emission spectra. A description of emission spectra for stars will be explained in

Section 1.2. Keller et al. (2001) were able to show that 30% of the brightest B type stars from these clusters are Be stars. In 2008 and 2010, spectroscopic studies were done on the B and Be type stars in h and χ Persei with the Spitzer Telescope by Currie et al. (2008) and Currie et al. (2010). The rotational velocities, effective temperatures, and surface gravity were again calulated. This is crucial because certain Be stars emit due to various reasons.

1.2 B Emission Stars (Be)

Emission spectra are caused by energy release in electron energy transitions (see Fig 1.2). This energy, as per the Bohr hydrogen model, is quantized and comes in discrete energy levels, shown in Figure 1.2. Stars consist mostly of hydrogen so as temperatures fluctuate electron energy transitions occur frequently. Surface temperatures of A type stars are ideal for electron energy transitions and will be discussed in Section 4.3. Though B spectral type stars are hotter and more energetic, B stars with emission (Be stars) are those whose physical properties also lend to an abundance of electron energy transition in the hydrogen Balmer series. The hydrogen Balmer series includes all energy transitions of the electron in a hydrogen atom to/from the $n = 2$ energy level. These transitions give photons in the visible range of light. The first two transitions in the Balmer series are the H_α ($n = 3 \leftrightarrow n = 2$) and H_β ($n = 4 \leftrightarrow n = 2$) energy transitions. The abundance of these transitions in the stars of h and χ Persei is the main study of this work.

There are different types of Be stars due to the mechanism that causes the emission spectra. A few of these are Luminous Blue Variables (LBV), High Mass X-Ray Binary (HMXB) companionships, and there are rapid rotating Be stars. LBV's are extremely large blue stars, spectral type B, who are so large that the surface gravity is not sufficient to retain the outer layers of the star. These stars then shed off the outer layer as part of the star but the material remains gravitationally bound. This outer layer then becomes a source for emission for the LBV. As the material becomes detached, the

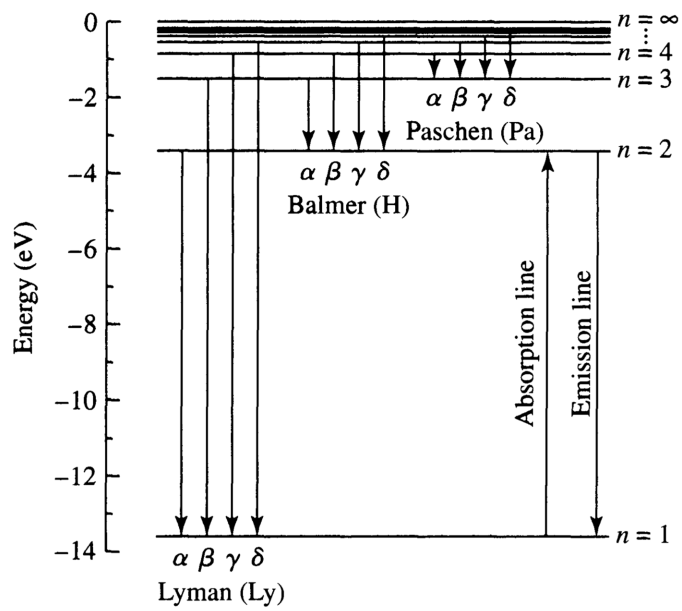


Figure 1.2 Shown are the possible energy transitions for electrons in the hydrogen atom. The Balmer series are the primary study of this work, particularly the H_α and H_β transitions, as they are abundant in Be stars.

star becomes unstable and begins to pulse. The change in mass and temperature can cause the star to fluctuate and their spectra are very unique with strong emission in the hydrogen Balmer emission series, particularly the H_α wavelength.

HMXB companionships compose of two objects, a high mass, compact object and the large companion. The high mass companion can be either a neutron star or a black hole, a small, but very massive object. This massive companion accretes/steals matter from the large companion, which can be either an O or B type star. The infalling matter results in X-Ray emission and is usually the marker for the companionship. For h and χ Persei the large companion is typically a B type star. The large companion then has an emission spectra created from the mass it loses to the massive companion. Again this emission is in the Hydrogen Balmer series, exceptionally strong at the H_α wavelength.

The third kind of Be star is a rapid rotating star. The emission spectra here is caused by an oblate star, with material at the equator being the emission source. The cause of this rapid rotation is still under study, but Boyer et al. (2012) notes that it could be that the star was "formed as a rapid rotator, spun up by mass transfer in a close binary system, or spun up during the main sequence evolution". They also note that there might be "other ... processes, such as non-radial pulsations [that give] additional angular momentum necessary for [the] material to leave the stellar surface." In the end, these mechanisms cause the mass at the equator of the star to begin to leave the star and this material is the emission source, emitting in the same H_α wavelength.

1.3 Telescope Filter Observations

Given the large spectral type stellar sample and Be population it is useful to use this double cluster for telescope filter calibration for the H_α optical wavelength. Telescope filter calibration is the process of calculating absolute magnitudes (true brightness) for different spectral type stars for the wavelength of light covered by each filter. Joner & Hintz (2015) have defined an H_α photometric system that serves as a companion to the H_β calibration established by Crawford & Mander (1966). Joner & Hintz (2015) work is done for 75 stars from three different stellar clusters: 8 from NGC 752, 12 from the Coma star cluster, and 24 from the Hyades. A future goal of this work is to add over 500 stars from h and χ Persei to the Joner & Hintz (2015) set.

The absolute H_α magnitudes of a large sample of stars can then be compared with work done by Joner & Hintz (2015) to further reduce errors. By comparing the data from h and χ Persei with their work, we are able to see if there is an offset in our magnitude calculation. This offset can then be corrected, this process is called zero-pointing. A complete H_α magnitude set set can then be applied to any stellar object. This H_α filter calibration would then be useful in indicating the spectral type of a star as well as the different types of Be stars. We show that the H_α emission is related with H_β emission in section 4.2. This relation can be exploited to further correlate our work with the filter calibration set established by Crawford & Mander (1966).

In order to calculate magnitudes of stars, astronomers use a technique called photometry. There are two fundamental types of photometry, aperture and point spread function fitting. The former is a simple method that will be discussed in Section 3.1. Unfortunately, aperture photometry cannot differentiate between stars in clusters. The stars in young clusters are too close for this method to work. Point spread function fitting (PSF) photometry, however, can separate the stars in the densely packed stellar regions of clusters in order to calculate magnitudes with high precision. PSF photometry fits a Gaussian curve to the photon count of light that is contained in a given telescope image. This will be explained in more detail in Section 3.1. PSF photometry allows for all the light from each stars to be accurately measured, no matter any nearby stars in the image. This is especially powerful for *h* and χ Persei being tightly packed clusters.

The version of PSF photometry that that BYU uses is free to the public, thanks to the National Science Foundation (NSF) and the Space Telescope Science Institute (STScI). It was written by Peter Stetson at the Dominion Astrophysical Observatory (DAO) and modified to work with IRAF by the STScI. IRAF is a command line program from the NSF and STScI that is very capable of handling the large amounts of data that telescopes produce. The PSF photometry process therein is titled DAOPHOT and will thus be called when referring to this specific version of PSF photometry. DAOPHOT is a package that comes inside the free astronomical data analysis software IRAF.

In order to calculate accurate magnitudes for the range of stellar types in each cluster, we utilize PSF photometry with DAOPHOT. As we will show in Section 4.1, this allows for the least error in magnitude calculation. For *h* and χ Persei this method of PSF photometry is crucial as it allows for separation of stars in the dense regions and accurate magnitude calculation despite the wide range in brightness. Accurate magnitudes for the clusters lets us do a comparison with the Joner & Hintz (2015) data in order to fully calibrate absolute H_α and H_β magnitudes.

The format for this work will be a discussion of the observations, a manual for DAOPHOT, *h* and

χ Persei magnitude analysis for the H_α and H_β wavelengths, and conclusions. In the observations chapter we detail the type of images taken. The wavelengths for the optical filters as well as the telescope optics are detailed. The DAOPHOT manual will discuss the proper use of the package for cluster work. There will be some sense of generality for the reader, however, applications for h and χ Persei are detailed as well, as these specific clusters are very dense stellar regions. An analysis of the H_α and H_β magnitudes follows. This discussion will center around work previously done by Joner & Hintz (2015) as well as noting new discoveries. A conclusion of the analysis will follow detailing any further work to be done.

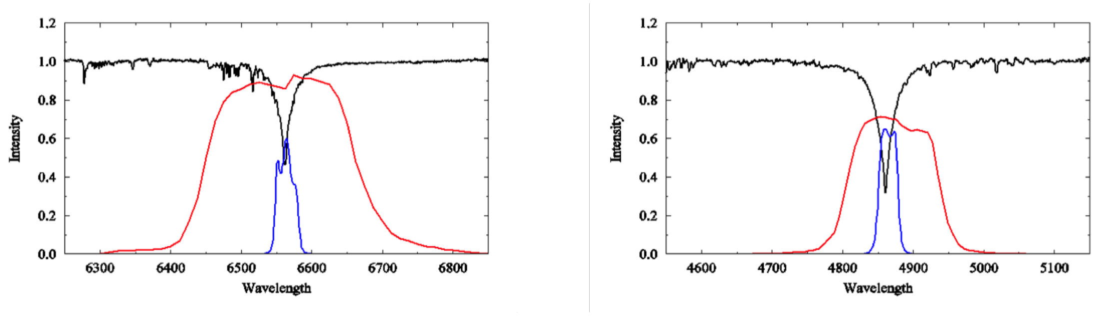
Chapter 2

Observations

Data were obtained over a 6 year span starting in 2010 going through 2016 from the West Mountain Observatory, owned by Brigham Young University and operated Dr. Michael Joner. Observations were made using the 0.91 m telescope. The telescope has a f/5.5 system giving a plate scale of 41"/mm.

Observations were made using a set of 4 optical filters focused on the Hydrogen α (H_α) and β (H_β) Balmer emission lines. Spectra for the filter sets are shown in Figure 2.1. Two of these filters are an H_α narrow filter (NA) and an H_α wide filter (WA). Both are centered on a wavelength of $\lambda_\alpha = 6561$ Å. The NA has a spectral width of 3 nm and the WA a width of 21 nm. The second set are an H_β narrow (NB) and an H_β wide (WB) filters. Both H_β filters are centered on wavelengths of $\lambda_\beta = 4867$ Å. The NB has width of 3 nm while the WB has a width of 15 nm. The benefit of having each pair centered on the same wavelength is that interstellar reddening as well as any reddening due to radial velocity does not have an effect on the magnitudes calculated.

Images were taken with two different cameras or Charge-coupled Devices (CCD). Most of the data was taken using the FLI Proline 3041 UV CCD. It has a pixel size of 2048x2048 with each pixel being 15x15μm. This yields a field of view is 0.61" pixel^{-1} giving a total 23.3' x 23.3' sky per image. The read noise for this CCD is 9.5 counts per pixel and the gain is 1.28 counts per pixel.



(a) H_α spectral **(b)** H_β spectra

Figure 2.1 The Spectra for the H_α narrow and wide filters are shown in (a). The H_α narrow filter is shown in blue and the H_α wide in red. Spectra for the H_β narrow and wide filters are given in (b). Again the allowed wavelengths for the narrow filter are in blue and for the wide filter in red.

The Read Noise and Gain are essential when calculating the PSF and thus need to be added into the image header if they do not already exist.

Some data were obtained using the FLI Proline 09000 CCD. This CCD has a pixel size of 3056x3056 with each pixel being 12x12μm. The field of view is 0.49" pixel^{-1} giving a total 25.2'x25.2' sky per image. The read noise is 13.2 counts per pixel and the gain is 1.53 counts per pixel.

With two different image sizes the coordinate file had to be adjusted in order to maintain consistency of the ordering of the stars. This issue was taken care of using an IRAF command that rescaled the coordinate file. More will be said about this rescaling in Section 3.4. We chose a night with exceptional seeing that was centered on each cluster, that of 11 Dec 2011, to be the source for the main coordinate file.

Weather conditions at West mountain typically yield seeing (area of sky that has good visibility) of 0.9". There are a total of 55 nights over the 6 year span from which data were analyzed. Nights with seeing that is significantly worse were thrown out. Overall the seeing for each frame remained relatively constant and any negative seeing effects can be ignored.

In our data set, some observations were made using short exposures and other with long exposures. The benefit of the short exposure is that magnitudes for the brighter objects can be calculated without over saturating the CCD. The long exposure data obtained magnitudes for the more dim objects. This is key in calculating accurate magnitudes for the bright and dim objects in each image. DAOPHOT is capable of calculating magnitudes for dim objects that are in proximity of brighter objects that became saturated in the long exposure, however short and long exposures isolate the bright and dim objects for precise magnitude calculation.

Images were reduced using standard IRAF procedures. The method used to get magnitudes from the processed data is a package inside IRAF called DAOPHOT. The following chapter gives an overview of the DAOPHOT process as well as indicating the specific decisions made for h and χ Persei.

Chapter 3

DAOPHOT Manual

DAOPHOT is a very robust and accurate method of PSF photometry. Peter Stetson wrote it specifically to handle cluster work. As such, DAOPHOT is can differentiate stars in a crowded group and accurately calculate magnitudes for each, whether it be very bright or dim. This differentiation gives an accurate magnitude for each star in the field thus yielding smaller errors in the calculated magnitudes. We chose to use DAOPHOT for *h* and χ Persei because in each there is a wide range of magnitudes and they are both very crowded fields.

The following is a straight-forward and detailed explanation of the point spread function photometry package DAOPHOT. The instructions herein are meant to instruct on the proper use of the DAOPHOT package. I reference Peter Stetson's DAOPHOT 2 manual for helpful tips. However, be aware that the DAOPHOT 2 manual follows the version of DAOPHOT that he wrote at DAO. The version on BYU campus and that is part of the IRAF package is rather different as it was compiled by the STScI. Thus the motivation for this manual.

For sake of simplicity, the manual will go as follows. All commands will be given in CAPS throughout as well as at the beginning of each section or paragraph. What follows will be an explanation of the command. Any parameters that need to be taken into account or changed will then be discussed. My comments on what I found to be useful will then follow. In order to fully learn what each command does and how it works with the whole package, it will be useful to the reader to read the HELP file for each command (syntax: "HELP command").

3.1 Introduction to PSF photometry

Photometry is the method of counting photons in order to calculate magnitudes for an object on a given telescope image. Each image that is taken by the telescope's camera (the CCD) is not only a picture, but a set of values. For each pixel in the image, a number of "photons" is counted and that number is given in the image data. As a given particle hits a pixel in the CCD array an electron is excited from the pixel to a main computer chip from which each electron is counted. However, there are some problems that arise: every electron counted does not correlate with a photon from the source. On this small scale there are random electron counts that are added due to heat and quantum tunneling. These extra counts are taken care of in basic image reduction techniques. Image reduction techniques are not discussed here beyond acknowledging that they are done in order to get a true count of the light from a given object. There are also photon counts added due to cosmic rays, stray particles in space. This makes accurately measuring the amount of light from a source a process.

There are two fundamental processes of photometry, aperture photometry and point spread function fitting (PSF) photometry. Each caters to the degree of accuracy needed. Aperture photometry creates circles around a specified object wherein all the photon counts are counted (see Figure 3.1). This method is simple and useful in other astronomical studies, but insufficient for cluster work. PSF photometry uses a variety of parameters in order to determine the objects that truly are light emitting/reflecting. The version of PSF photometry used in this work, DAOPHOT, is capable of discerning faint stars from cosmic rays whether these cosmic rays be near other bright, saturated stars or not. This is crucial for accurate magnitude calculation in cluster work.

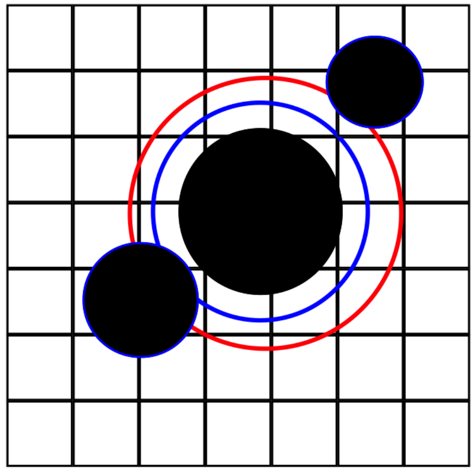


Figure 3.1 An illustration of aperture photometry. Shown are a few objects of interest over the CCD pixel array. The inner circle is the aperture size wherein all photon counts are attributed to an object and the sky between the telescope and said object. The outer circle is accounts for sky counts. Note how in crowded fields, nearby objects encroach the area attributed to the target, thus falsifying the true count from a single target.

Aperture photometry is quick way of calculating brightness. The process here is merely tallying each count inside two circles called apertures (Fig. 3.1). The outer circle is based on the assumption that these counts are only from the ambient light of the sky. The inner circle is based on the assumption that the counts here are only from your object of interest. Counts from the sky are then subtracted from the object counts and the object counts are used to calculate magnitudes.

The problem with aperture photometry is that, for distant objects such as stars, the light does not come as a disk but as a point source. When central point source hits the CCD it may not hit at the center of a pixel, thus offsetting the target. As the photons hit the CCD, the counts fill up one pixel well then spill into the surrounding pixels. This puts the target over various pixels. The result is that the edges of a target cover fractions of pixels. If the apertures of the star or sky run through those edge pixels, the counts are then ambiguous to the calculation and any type of photon count will be counted incorrectly. Thus all photon counts in a given pixel contain counts from the sky, cosmic rays, and star counts. All the counts are then thrown in with the sky counts or the target counts depending on the aperture in which they reside. Worse yet is if a neighboring objects lies within the aperture of the target star, extra star light counts can be added. All these inaccurately change the magnitude for the object.

Aperture photometry is then sufficient only when a star field is very sparse, the target of interest is isolated, or rough magnitudes will suffice for the science at hand. Variable star work is done accurately with aperture photometry when the the stars are isolated. Photometric analysis is most readily done with aperture photometry whenever the science allows.

Generally PSF photometry is much more accurate in its method. Precise calculations are made based on input parameters from the user to accurately indicate which objects are light emitters/reflectors. Here the idea of an aperture doesn't exits, but a light profile. Light appears on a CCD specific to the type of object. Cosmic rays appear as hot pixels, but since they are more than just an electron, they tend to give more counts and appear just as bright as any object in the frame. Rather than counting all the light within an aperture of the object, a point spread function is fit to every object on the frame.

PSF photometry, as done in DAOPHOT, creates a point spread function of the photon counts over the pixel area then counts all the counts under that curve. The PSF of a light emitter/relector is made from parameters that are specific to each image. If the object does not have a profile that matches that of a light emitter/reflector, that object is thrown out. If the object does match but is overlapped by another similar object, counts can be accredited to the object they come from. In an area of overlap the increase in counts, when the counts should be decreasing, can be split between the respective objects.

It is then clear that PSF photometry is able to give much more accurate magnitudes than aperture photometry. If an object is isolated, there is no need for a curve fit to the light spread as all the counts will be accounted for. However if even two objects are within the aperture of a star, it is necessary to utilize PSF photometry so as to differentiate the light from each object.

```
ecl> noao
      artdata.      digiphot.     nobsolete.    onedspec.
      astcat.       focas.        nproto.       rv.
      astrometry.   imred.        observatory   surfphot.
      astutil.      mtlocal.      obsutil.      twodspec.

noao> digi
      apphot.   daophot.  photcal.  ptools.

digiphot> dao
      addstar       daotest       nstar         pexamine      psf
      allstar       datapars@     pcalc         pfmerge       psort
      centerpars@   findpars@     pconcat       phot          pstselect
      daoedit       fitskypars@   pconvert      photpars@     seepsf
      daofind       group         pdump         prenumber     setimpars
      daopars@      grpselect     peak          pselect       substar
```

Figure 3.2 The package navigation from IRAF login to the DAOPHOT package.

Magnitude calculation is then only feasible in clusters when PSF photometry is used. The tightly packed region of space in which all the stars reside leads to many objects, being not only close, but often overlapping. PSF photometry is then able to differentiate stars from cosmic rays and from each other using parameters specific to each image. Accurate magnitudes can be calculate for every individual star in the frame. This becomes all the more powerful when there are hundreds or thousands of stars as is the case of *h* and χ Persei.

3.2 DAOPHOT

A fundamental knowledge of IRAF is assumed. Basic navigation throughout the main IRAF packages is part of that. It is crucial that the reader understands the fact that every package has a parameter file and help file and how to access each and edit the parameter files. This will not only make sense of the following manual, but also provide further assistance where this manual lacks.

Figure 3.2 shows the basic navigation inside IRAF as well as the different packages available.

It also shows where the DAOPHOT package lives. Be aware that if the WMODAO script (see Appendix A) is implemented, you must be in the DAOPHOT package as some commands also exist elsewhere but are slightly different.

Note that all parameter files end in "@". All other commands are valid tasks that can be used. Most tasks will be addressed herein, but again, any fundamental questions can be answered using the "HELP command".

The following is the command procedure for running DAOPHOT.

1. Find Parameters specific to each image
2. Input the Parameters
3. run DAOFIND (optional see Sec 3.4)
4. run PHOT
5. run PSTSELECT
6. run PSF
7. run GROUP
8. (optional) run SEEPSF
9. (a) run NSTAR

   and/or

   (b) run ALLSTAR

Due to the fact that PSF photometry is very robust and mathematical, different parameter inputs dependent on each frame are required by the user in order to produce the best results. In the following section we will discuss the key parameters that are required and in the section after that any other helpful parameters. Note that each command in DAOPHOT has its own parameter file. These can be accessed using EPAR "command". The help file for each command will specify every parameter as well as a brief introduction. I have found that these help files are, at times, rather unhelpful and so will discuss the key parameters herein.

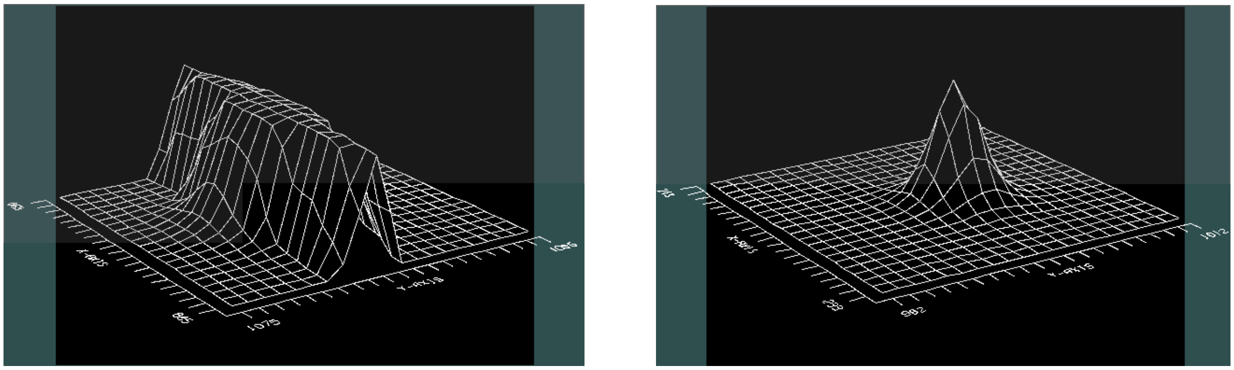
3.3 Picking the Parameters

3.3.1 FWHM, StDev, Good Data Max, Psfrad

When calculating the parameters, it can be helpful to try a rough estimate parameters the first time around. Then run each step of DAOPHOT, taking note of the output files and the errors contained therein. Each file typically gives a brief description of the error, however, many errors are due poor choice in parameters. The main parameters that are used by DAOPHOT are the following:

FWHM: the Full Width at Half Maximum of an average star on the frame. Due to the dispersion of photon counts over the CCD, the profile for a star tends to have a Gaussian distribution. It is for this reason that PSF photometry works. The FWHM is the pixel width of the photon count profile at half the maximum count value. Therefore it is best to find the FHWM value using stars that are not completely saturated, as you will see the full profile of a star on your frame (Fig 3.3 (b)). You can check that you are getting the full profile of the star by pressing S over the center of a star (Fig 3.3). This will give a 3D plot of the count spread over your CCD. The x and y axes here are the pixels of the CCD and the z axis is the count value. Make sure the star you use to find the FWHM has the full peak shown and is not over saturated as is Fig 3.3 (a).

To find a numerical value run IMEXAM and press A over the center of various stars throughout the image frame. A table will appear in the IRAF terminal. The value we are looking for is titled "Enclosed". This value is the Full Width at Half-Maximum (FWHM) of the pixel count for the objects over the image. This value is required in the DATAPARS parameter file, which is used



(a) Over saturated stellar profile

(b) Unsaturated profile of all the star light

Figure 3.3 Shown are graphs of how the photons are distributed over the CCD. Run IMEXAM and press S over the center of a star to get this graph. The x and y axes represent the CCD and the z axis is the counts. Over saturated stars get cut off and estimation of the FWHM is thus overestimated (a). A full star that is not over saturated will appear as a full 3D Gaussian (b).

throughout the DAOPHOT process. Other parameter values depend on the FWHM as well so it's best to be meticulous when picking this value. The FWHM is by far the most crucial parameter in calculating the PSF.

Stdev: The standard deviation (stdev) of the background counts. To get this value, run IMEXAM and press M over various sections of sky for the given frame. A table will appear in the IRAF terminal and the value is under "stddev". The stdev is used in the DATAPARS parameter file which is used by multiple processes. PHOT uses the value when calculating a rough magnitude for each star. When first learning the procedure it can be helpful to know a general value for your set of images so you can go through the process a few times without having to calculate a precise value each time. Though this value is important, the value is generally much smaller than your photon counts from your object. As such there is forgiveness when picking the stdev when running all the processes besides DAOFIND. DAOFIND uses the stdev when calculating the threshold (discussed later in this section). A more precise stdev will yield higher success in determining stars for the

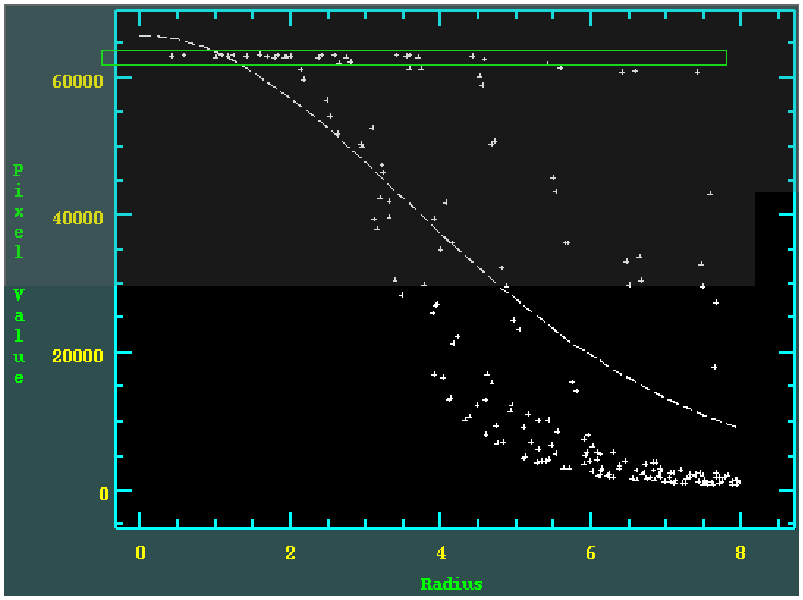


Figure 3.4 A plot of the photon counts vs. the pixel radius those counts cover. The section with a square box is a valid photon count range for the Good Data Max value. The key for the Good Data Max is not the curve that IRAF gives (the dotted line) but where the counts begin to level off, indicating saturation of the CCD.

coordinate file made by DAOFIND.

Good Data Max: The highest photon count that you want DAOPHOT to account for when deciding the peak of the PSF curve for each star. Figure 3.4 shows a rectangle outlining a good range for the Good Data Max. To get this plot run IMEXAM and press R over the center of a star. It is generally safe here to pick a value that is a few thousand counts over the highest peak for your field. Regardless the focus of your work being the dim or bright objects on your field, a high enough value needs to be chosen in order for the PSTSELECT command to find enough PSF stars. As such it is ok here to pick over saturated stars and see where they level off.

Between finding the good data max and the psfrad (the next parameter to be discussed), you want to change the parameters of IMEXAM to see different parts of the same plot. To edit the parameters for this specific plot type RIMEXAM. This will bring up the parameter file for the radial image plot. For the good data make sure that your y axis is set to INDEF for both y1 and y2 (the

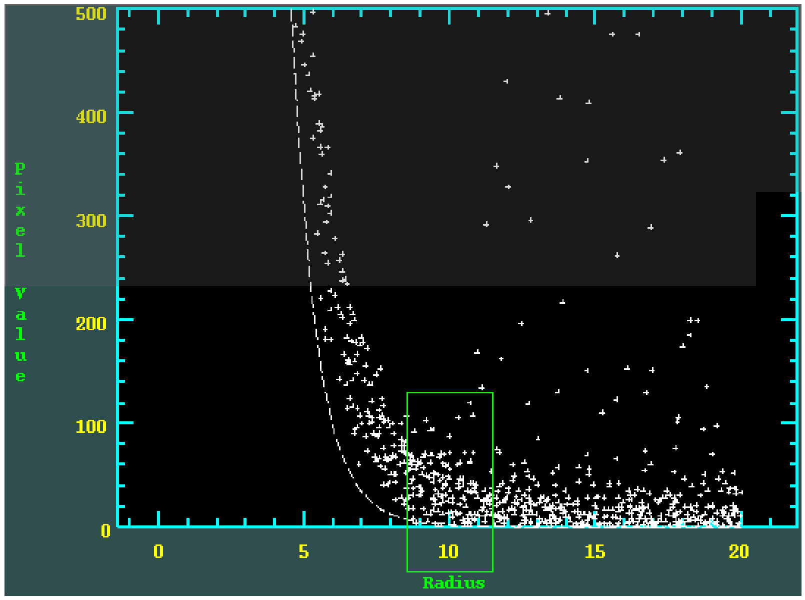


Figure 3.5 A plot of the photon counts vs. the pixel radius the counts cover. Here we show the lower end of the counts as the curve off into the background. Note that there is another star close to the target. The squared area shows a good choice in pixel radius for the PSF radius.

max and min of your y axis). Here the default plotting radius (rplot) of 8 is typically well. You can change it in order to get the same view of the count profile as Fig 3.4. For the psfrad you want to change the y axis to sufficiently low so that you can see the count distribution begin to curve into the background counts. Again, RIMEXAM will allow access to the radial plot parameters and you can change your y1 and y2 values to something small for your image.

Psfrad: The radius for the point spread function (PSFRAD) fit for objects in your frame. The point spread function is the curve under which all the photon counts generated by your object will be counted. If the PSFRAD is too small, you won't count all the photons from your object. If the PSFRAD is too large, you could include other objects, which in turn won't give an accurate magnitude for a single star.

The PSFRAD is best found by looking at how the counts are distributed for the image. To do this, change the radial plot image examine parameters to display only a narrow range of pixels at a low photon count. This is done by typing "RIMEXAM" and changing the y axis min (y1) to 0 and max (y2) to something relatively small. Typically one tenth to one hundreth of your good data max is a good value for your y max. Then you hold the cursor over the center of your star and type "r". This will bring up a radial plot of the pixel counts with respect to the pixel radius size (Fig 3.5).

After much trial and error on my part, the best value of the PSFRAD is chosen where the pixel counts begin to level off. Thus any radius inside the box in Fig 3.5 would be a good value, with the best being at a value of 10. Here you also want to pick stars that are unsaturated so that you can see how the counts are distributed for an actual star on your image and not just an oversaturated blob of photon counts. Note that decimals are okay but, as you will see with most of the parameters, results are unchanged beyond three significant figures.

A good check to see if your PSFRAD choice was done correctly is to display the "[image].extenion.psf.number.fits" image in DS9. Here you can see the point spread function that was created. Look to see if your object is fairly circular and that no other stars creep onto the main object. These would appear as bright spots on or near your central object. This is especially crucial for crowded fields. If your field is not very crowded, sufficient PSFRAD is 4 to 5*FWHM.

3.3.2 More parameters

In my own practice with DAOPHOT I have found many more parameters that are useful in modifying for different circumstances. The most important of these parameters have been written in the WMODAO package for automatic calculation and input into their respective parameter files. The first few times you run DAOPHOT it will be best to go step by step through the process, manually calculating and inputting the following parameters as it will help familiarize you with the packages and parameter files.

Fitrad: the pixel radius within which photons are attributed to a a star. Peter Stetson said that a good value for this is 1.4*FWHM, this can be found in the DAOPHOT 2 manual. I played with this value and found that his suggestion was sufficient, however, if the pixel space between overlapping stars is less than 1.4*FWHM, use the smaller value. In my work on *h* and χ Persei, I used a value of 1.4*FWHM. This is the value that I put into my WMODAO script (see Appendix), however, that too can be changed to meet the needs of the target.

Skyannul: the inner radius of the sky annulus used by ALLSTAR to recompute the sky values. ALLSTAR (Sec 3.6) uses this value when doing the final calculation for the photon count from the object as opposed to the background sky counts. The skyannul is used in FITSKYPARS. Through trial and error I found that a value of 3 to 4*FWHM is generally good for the inner radius. For the skyannul width, a value of about the FWHM is generally fine.

Cbox: the centering box width defines an area of tolerance that all processes have in finding the center of an object from the coordinate file. A large (2 to 3*FWHM) cbox value is ok for fields that are not very crowded. For crowded fields, like *h* and χ Persei, I found it best to use a value close to the FWHM. This value is used throughout the DAOPHOT process whenever the centers of objects from the coordinate file do night exactly match the pixel coordinates for the peak of the photon count. If you make a master coordinate file and manually light up the coordinate file with the other image frames in your data set then this value is key to keep in mind. When run manually, check the image.ext.mag.number file. You want to minimize the errors and reducing the shift of the center of your coordinates helps with this. The parameter Maxshif, also found in the CENTERPARS, allows for a shift in the cbox area and as such, is typically best left small. For crowded fields stay close to the FWHM for this value, for more sparse fields, a value as high as 2*FWHM is good, but never higher.

Threshold: a multiple of the stdev that determines the photon count limit for which an object will be deemed a star. The threshold limit is used by the DAOFIND command in determining anything that is too faint to be deemed a star for the field. Depending on how low, or high, you put this value you could pick up on very faint objects that might not be of interest. Due to the 3 dimensional nature of clusters, I found that a lower value was best. This would account for any cluster members that were perhaps at the far end.

3.4 Step 1: The Coordinate File

DAOFIND: this command uses the parameters Threshold, Round, and Sharpness in order to determine what objects in a given frame are actually stars, or light emitting/reflecting objects, as opposed to cosmic rays (the random lone particle or photon from anomalies throughout the universe). The pixel location of the object is then stored in a file (name.ext.coo.1) that is used for the rest of the commands that make up DAOPHOT. Not only is every star selected, but it is also centered to the point of greatest photon count, thus eliminating the need for any further manual centering.

A master coordinate file was used to keep consistency of star numbering. A coordinate file was chosen from a night with particularly good seeing, December 11, 2011. This file was then modified to align with other images from other nights. Modifications were done manually with the SAOImage DS9 imaging software. Due to the different CCDs used in the data acquisition, changes also had to be made for the different image sizes (3k CCD $\rightarrow$ 2k CCD)

This command is particularly useful for clusters though may not always be necessary. In less crowded star fields, through ds9, objects of interest can be manually selected and saved into a coordinate file. However, recall that in a cluster there are hundreds of stars that are very close, and sometimes overlapping. This makes the manual method of selecting each star by yourself very difficult.

The parameters that are involved are the threshold and round. The best way to pick them is dependent on your image. A discussion on the outcomes with pictures showing generally the ranges that DAOFIND can achieve -> powerful

3.5 The Intermediary Steps

PHOT: Calculates a preliminary magnitude in order to begin the algorithms required for selecting PSF stars. This is done using aperture photometry on the list of stars found in DAOFIND. PHOT calculates accurate centers, sky values, and magnitudes for said list. Here the centering algorithm (found in CENTERPARS) can be changed to a few different values. The available options are found in the help file of CENTERPARS. I found that centroid gives a nice center for each object. You also want to check the "image.mag.number" file afterwards for "ERR" (errors or objects for which PHOT had problems). You can play with the "maxshif" value in CENTERPARS to reduce the number of "ERR". I found that typically a value of two to three times FWHM was generally good for minimizing the "ERR"s.

PSF: Creates a PSF curve to fit a general star for the frame. This command can be specified in a few different ways using the function parameter. I have found that leaving the default of Gaussian works well for a crowded field, however, if you know how the light of your object is spread over the pixels you can change this to a few other types of functions.

In order to create a sufficient list of PSF stars, the curve needs to be accounting for ALL of the photon counts. This is where the Good Data Max is important. If you chose too low, then the "stars" that are found won't contain all the counts that actually came from that star, thus making the PSF fit too shallow and then the list of generic PSF stars won't be as robust as is required to fully count all the counts from each star in your frame. The main point of the Good Data Max is that it needs to be sufficiently high to account for all the photon counts from each star. I found that even if you are above where the counts level off (recall Fig 3.4) by a thousand counts or so then you have chosen your Good Data Max wisely.

PSTSELECT: Selects a set of PSF stars to use as basis for the rest of the stars. The default on this is to select 25 stars. I found that the final steps don't produce quality results for only 25 PSF stars out of my targeted 500. Depending on the field, fewer PSF stars is a perfectly valid basis to run the final stages, however, in a crowded field it is helpful to up this to 50 stars. Using this many basis PSF stars produced quality results.

PSTSELECT relies on the number of PSF stars that were found in the PSF command. Here you want to make sure that you chose a large enough Good Data Max in order to get the required number of PSF stars. If there aren't enough PSF stars, then PSTSELECT won't finish as it didn't have enough PSF stars.

GROUP: Groups the entire list of stars according to which PSF star they most resemble. This is crucial when running NSTAR, as NSTAR does not group or check the groupings. However, ALLSTAR does check the grouping and through its iterations, actually regroups so this step isn't as vital if you are only running ALLSTAR.

3.6 The Final Step(s): a light at the end of the tunnel

Congratulations! You have made it this far. We now have a fork in the road as to which final step we can take. The two choices are the ALLSTAR or NSTAR commands. Both commands fit the calculated PSF to each star and count the photons underneath, thus calculating an instrumental magnitude for each object.

The key difference between the two is accuracy. In this work we needed the most precise magnitudes possible and thus have chosen to utilize ALLSTAR for calculating H_α and H_β magnitudes for h and χ Persei.

The two are very similar but ALLSTAR does multiple iterations whereas NSTAR only goes through once.

3.6.1 Allstar

ALLSTAR: Counts the photon counts under the specified curve that matches each star. ALLSTAR then regroups the stars and recounts the photons, ensuring that every photon that mostly likely belongs to a given star is counted. This process repeats for as many iterations as you define in the (blank) parameter.

3.6.2 Nstar

NSTAR: Counts the photon counts under the specified curve that matches each star. The output is ".nsr" or ".nrj". These files contain a list of the magnitudes that correlate to any given object using the reference of the pixel coordinate system.

Chapter 4

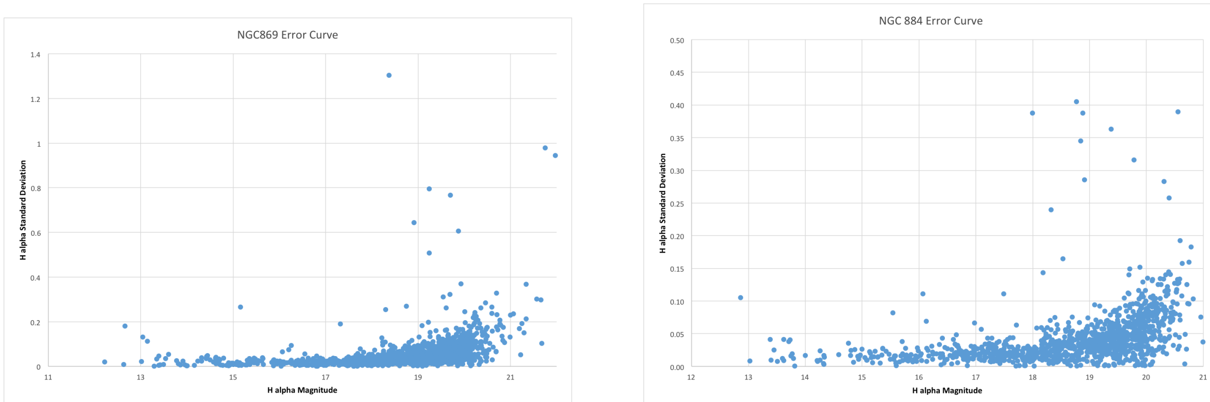
h and χ Persei

In this chapter we analyze the magnitudes calculated from DAOPHOT. We begin by describing the errors in the calculated. These low errors came from the quality data produced by DAOPHOT. Next we will analyze the index magnitude data using methods described in Joner & Hintz (2015). This analysis gives a quick method of finding Be stars in clusters using only photometric data. We then reproduce a Color - Color diagram using the H_α index vs the H_α magnitude and discuss our findings.

4.1 Error per Observation

The error per observation is the the amount of uncertainty that is inherent in the magnitude data from image frame to image frame. This is crucial to note because the magnitude data that were obtained detail the brightness measured from each individual frame, but due to the object, the data is not always self consistent. The magnitude can change due to a number of factors. As discussed previously, LBVs change magnitude as well as temperature, thus leading to a larger error in the magnitude. Variable stars also produce larger errors as they vary in magnitude.

In Figure 4.1 we show the errors in magnitude for h and χ Persei. Plotted is the H_α magnitude



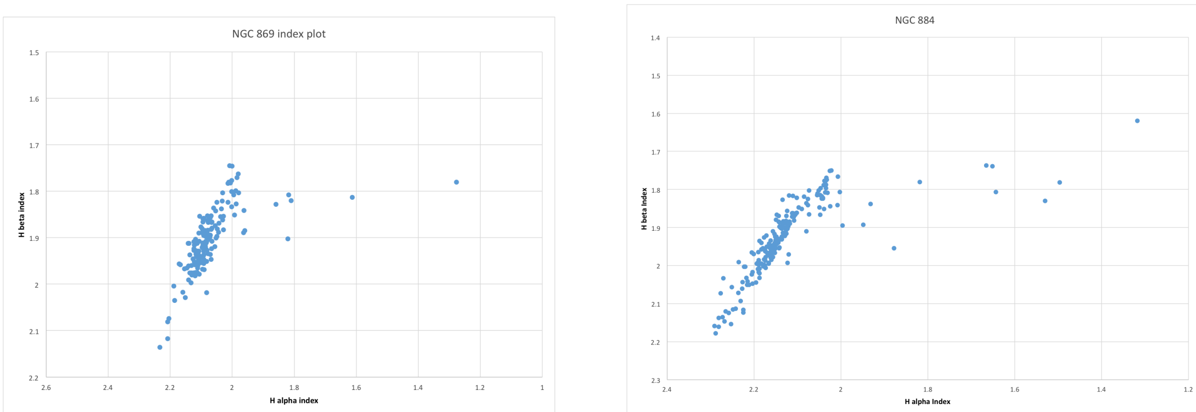
(a) h Per **(b)** χ Per

Figure 4.1 Plotted are the H_α index vs H_β index values for each star in the double cluster. The line has been shown to be the main sequence with O type stars starting at the $H_\beta \sim 1.75$ range and going down then coming back up at $H_\beta \sim 2.2$ for A2 type stars. Outliers to the right of the line are Be type stars.

for each star in the clusters vs the H_α index that will be described in Section 4.2. It can be seen that the magnitude data that we attained from DAOPHOT has a very small error range over a large range of magnitudes. With a value well below 0.05 for h and 0.05 for χ over ~ 5 magnitude. It was for this reason that DAOPHOT was utilized for h and χ Persei.

The magnitude data analyzed here was 4 nights h and 4 for χ. The error curve can be seen as a trend that starts below a 0.05 H_α index value then curves upward as magnitude value increases, or brightness decreases. This curve is expected due to the fact that more dim stars, those with higher magnitude value, have a larger error in the calculated magnitude. With the addition of more nights from the data set we expect the curve to tighten with less error below the curve at higher magnitude.

As was mentioned, variable stars tend to have a higher error due to the fact that their magnitudes are changing. All of the stars that are above the curve are the variable stars in the field. The brighter variables are those to the left with H_α magnitude values of $\sim 13-16$. With this small error we are confident in saying that those stars that lie above the curve here are variable O and B type stars, that is, the Be stars of the clusters.



(a) h Per **(b)** χ Per

Figure 4.2 Plotted are the H_α index vs H_β index values for each star in the double cluster. The line has been shown to be the main sequence with O type stars starting at the $H_\beta \sim 1.75$ range and going down then coming back up at $H_\beta \sim 2.2$ for A2 type stars. Outliers to the right of the line are Be type stars.

4.2 H_α H_β Analysis of *h* and χ Persei

We begin by defining the H_α magnitude index that was introduced in the previous section as well as the H_β magnitude index. As described in Joner & Hintz (2015), "the indices are fomed by taking measurements through a narrow and wide filter centered on the hydrogen spectral line[s]". Centering the pair of filters on a single wavelength allows for direct comparison of the magnitudes from each filter. The H_α magnitude index is calculated using

$$\alpha = m_{\alpha-narrow} - m_{\alpha-wide}$$

and the H_α index value is

$$\beta = m_{\beta-narrow} - m_{\beta-wide}$$

where $m_{filter-description}$ is the calculated magnitude from the given filter. These indices are meant to work with the data from Joner & Hintz (2015) and Crawford & Mander (1966).

Plots of the H_α vs H_β for the double cluster are shown in Figure 4.2. On the x-axis is the H_α

index and along the y-axis is the H_β index. The trend here is linear and Joner & Hintz (2015) showed that at the top of this linear trend are the hotter stars in the cluster, O and B types. Following the trend downwards follows the spectral types of the cluster with a small gap around $H_\beta \sim 2.1$. The trend then continues with a turn around, A2 type stars, coming back up the line while continuing down the spectral types. The gap is then reached again at the same H_β index value.

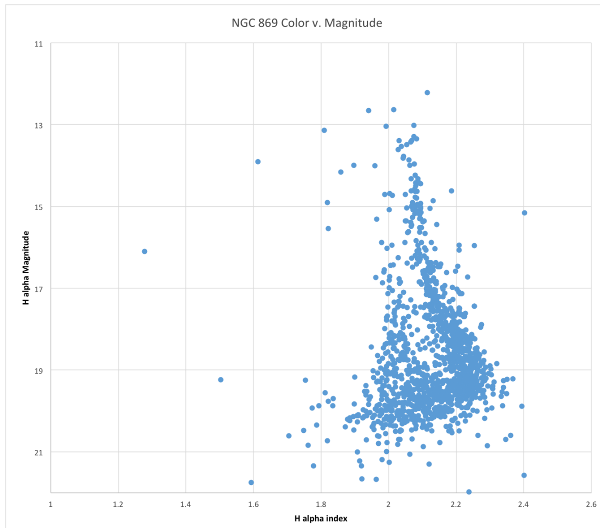
This gap is intriguing because it is due to objects from the instability strip of the Hertsprung-Russell Diagram. On the trip down the variables here are δ Cepheid variables and on the trip back up are the smaller, δ Scuti variable stars. This gap shows us that the magnitudes found using DAOPHOT are accurate.

Due to the unique characteristics of emission, the outliers to the right of the line are Be stars. The H_α and H_β wavelengths are due to energy transitions of electrons from the $n = 3$ to $n = 2$ and $n = 4$ to $n = 2$ energy levels respectively. The energy required for an electron to make these transitions comes from radiative energy from the star hitting the gas surrounding it. Because the H_α transition requires more energy, it continues past the H_β transition. This can be seen in Fig 4.2. The H_α magnitudes for the strong emission objects continue past the end of the H_β magnitudes. For this reason we can confidently say that the outliers are Be stars.

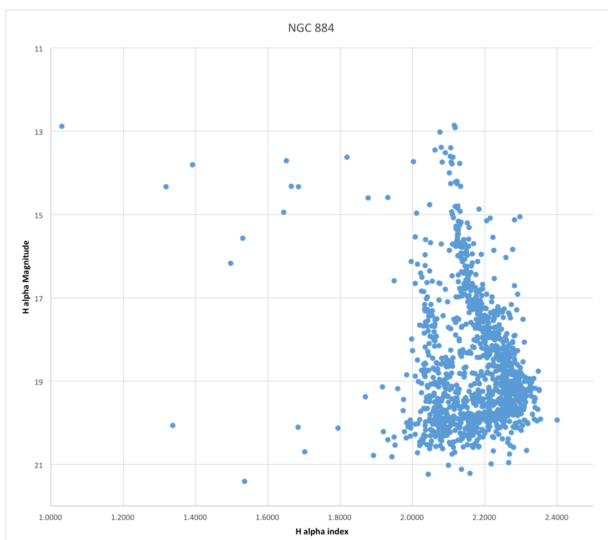
4.3 Findings from the Data

In Figure 4.3 we have an H_α relative magnitude plotted with respect to the H_α index previously described. This plot was first described in Joner & Hintz (2015) using data from M67 and NGC 752. The color - magnitude diagram for h and χ Persei has a few interesting features. At the rightmost end we have a main trend line following the spectral types of stars, there are objects to the left that are outliers, and a group of stars that runs parallel the main curve.

The first thing to note is that the plot follows the stellar spectrum. From the upper end of the



(a) *h* Per



(b) χ Per

Figure 4.3 An H_α wide magnitude value vs H_α index plot is shown following the trend from (Joner & Hintz 2015). Here we point out the parallel lines that are not found in (Joner & Hintz 2015). Reasoning behind this parallel is addressed.

rightmost curve we have O type stars. These have been cross checked with existing SIMBAD information using our plate scale solution. At the bottom of the rightmost group there is a curve to the left that is due to A2 type stars. This curve and flattening out is due to the fact that A spectral type stars have peak Hydrogen Balmer emission characteristics due to the ideal temperatures (7000 - 11000 Kelvin) for the H_α and H_β electron energy transitions.

Next we have the outliers that lie above 15 H_α magnitude. These objects are bright with a higher H_α index that the main stellar group of the cluster. Cross reference with the SIMAD database has shown that for both clusters, these are all Be type stars. Brighter H_α index is due to the emission characteristics of Be stars. As discussed previously, Be stars are unique in the emission spectra caused by their physical characteristics. This emission puts the Be stars to the left of this plot, indicating a unique method in detection of Be stars without needing spectral data.

The main anomaly of the trend is the parallel grouping to the left of the rightmost group. This parallel grouping was not seen in the previous data of Joner & Hintz (2015). Cross reference to the SIMBAD database of the stars in this left parallel group was interesting. These stars are noted as being simply stars in the given cluster. This was the result for both h and χ Persei.

The initial thought was that this parallel might just be low emission objects that have the same linear relation but at a lower H_α index. However, this unique H_α emission would lead to these stars having unique spectral information that would have set them apart from the main cluster stars. No previous study has shown any unique spectral information on these stars.

It is our thought that this left parallel could simply be a group of foreground stars. These stars would have the same spectral characteristics of the main cluster stars yet, having a brighter H_α index, would lie closer to us than the clusters. Perhaps a separation distance could be calculated from the left parallel group to the main stars of the clusters. Further analysis is required to determine if this is possible.

Chapter 5

Conclusions

We have done PSF photometry on the young double cluster h and χ Persei over the H_α and H_β wavelengths. The cluster pair was chosen due to the amount of study that has already been done as well as the broad spectral type sample that is found therein. Analysis has shown that there is a method for finding Be stars that can be done with photometric data over the H_α and H_β wavelengths. PSF photometry was used in order to calculate precise magnitudes for the crowded array of stars in each cluster.

The values we have calculated here are merely apparent magnitudes. We will compare these to the values from Joner & Hintz (2015). The H_α will be zero-pointed and the H_β zero-point will be checked for consistency. These results will then be checked with Crawford & Mander (1966). A conclusion of the work is to calculate accurate zero-points that would then allow for calibration of the H_α magnitudes found herein. This will be done by completing the DAOPHOT process for all data available to us.

As we continue finding magnitudes for more nights of data, we will be able to reduce errors. This will help show with more confidence the findings we have discussed here. Finishing all the nights on h and χ Persei will give more magnitudes that can be used to further increase validity of this work.

With smaller errors we hope to find exactly which stars lie in the left and right spreads of Figure 4.3. Plate scale solution was done and the stars on the left were found as purely stars in each cluster. With a different alpha magnitude we suspect that there is some physical reasoning to these stars. More study is required to resolve this finding.

Further analysis of the magnitudes found here can be applied to a wide variety of work. The age of the clusters is still under study and has yet to be determined. Accurate distances are also needed as the estimated ranges have error. Further study of the H_α and H_β magnitudes can lead to the creation of a color - color diagram that would give details about the evolution of the cluster and distance.

And that's all I have to say about that.

Appendix A

My parameters, login.cl, and WMODAO

The following is a full list of my parameters for my work done on h and χ Persei, my login.cl, and a script that I modified that runs the entire DAOPHOT process in a single run. Note that the script has to change the key parameters each time it is run. The command WMODAO runs the script and must be done while inside the DAOPHOT package as it calls commands specific to DAOPHOT.

```
daophot
     (version = "May00")
        (text = yes)           Set the default output photfile format to text?
       (wcsin = "logical")     The input coordinates wcs
      (wcsout = "logical")     The output coordinates wcs
      (wcspsf = "logical")     The psf coordinates wcs
       (cache = no)            Cache image in memory?
      (verify = no)            Verify critical parameters?
      (update = no)            Update critial parameters?
     (verbose = yes)           Print verbose output?
    (graphics = "stdgraph")    Default graphics device
     (display = "stdimage")    Default display device
        (mode = "ql")

centerpars
  (calgorithm = "centroid")    Centering algorithm
        (cbox = 0.)            Centering box width in scale units
  (cthreshold = 0.)            Centering threshold in sigma above background
```

```
  (minsnratio = 1.)           Minimum signal-to-noise ratio for centering algorithim
    (cmaxiter = 15)           Maximum iterations for centering algorithm
    (maxshift = 0.)           Maximum center shift in scale units
       (clean = no)           Symmetry clean before centering
      (rclean = 1.)           Cleaning radius in scale units
       (rclip = 2.)           Clipping radius in scale units
      (kclean = 3.)           K-sigma rejection criterion in skysigma
    (mkcenter = no)           Mark the computed center
        (mode = "ql")

daopars
    (function = "gauss")      Form of analytic component of psf model
    (varorder = 0)            Order of empirical component of psf model
      (nclean = 0)            Number of cleaning iterations for computing psf model
   (saturated = no)           Use wings of saturated stars in psf model computation?
    (matchrad = 0.)           Object matching radius in scale units
      (psfrad = 11.1)         Radius of psf model in scale units
      (fitrad = 0.)           Fitting radius in scale units
    (recenter = yes)          Recenter stars during fit?
      (fitsky = no)           Recompute group sky value during fit?
    (groupsky = no)           Use group rather than individual sky values?
    (sannulus = 0.)           Inner radius of sky fitting annulus in scale units
   (wsannulus = 0.5)          Width of sky fitting annulus in scale units
     (flaterr = 0.75)         Flat field error in percent
     (proferr = 5.)           Profile error in percent
     (maxiter = 50)           Maximum number of fitting iterations
     (clipexp = 6)            Bad data clipping exponent
   (cliprange = 2.5)          Bad data clipping range in sigma
    (mergerad = INDEF)        Critical object merging radius in scale units
 (critsnratio = 1.)           Critical S/N ratio for group membership
    (maxnstar = 10000)        Maximum number of stars to fit
    (maxgroup = 60)           Maximum number of stars to fit per group
        (mode = "ql")

datapars
       (scale = 1.)           Image scale in units per pixel
     (fwhmpsf = 0.)           FWHM of the PSF in scale units
    (emission = yes)          Features are positive?
```

```
      (sigma = 5.048)         Standard deviation of background in counts
    (datamin = INDEF)         Minimum good data value
    (datamax = 67000.)        Maximum good data value
      (noise = "poisson")     Noise model
    (ccdread = "READNOISE")   CCD readout noise image header keyword
       (gain = "GAIN")        CCD gain image header keyword
  (readnoise = 9.5)           CCD readout noise in electrons
      (epadu = 1.28)          Gain in electrons per count
   (exposure = "EXPOSURE")    Exposure time image header keyword
    (airmass = "AIRMASS")     Airmass image header keyword
     (filter = "SUBSET")      Filter image header keyword
    (obstime = "DATE-OBS")    Time of observation image header keyword
      (itime = 1.)            Exposure time
   (xairmass = INDEF)         Airmass
    (ifilter = "INDEF")       Filter
      (otime = "INDEF")       Time of observation
       (mode = "ql")

findpars
  (threshold = 4.)            Threshold in sigma for feature detection
     (nsigma = 1.5)           Width of convolution kernel in sigma
      (ratio = 1.)            Ratio of minor to major axis of Gaussian kernel
      (theta = 0.)            Position angle of major axis of Gaussian kernel
    (sharplo = 0.2)           Lower bound on sharpness for feature detection
    (sharphi = 1.)            Upper bound on sharpness for  feature detection
    (roundlo = -1.)           Lower bound on roundness for feature detection
    (roundhi = 1.)            Upper bound on roundness for feature detection
(mkdetections = no)           Mark detections on the image display?
       (mode = "ql")

fitskypars
 (salgorithm = "ofilter")     Sky fitting algorithm
    (annulus = 0.)            Inner radius of sky annulus in scale units
   (dannulus = 0.5)           Width of sky annulus in scale units
   (skyvalue = 0.)            User sky value
   (smaxiter = 10)            Maximum number of sky fitting iterations
    (sloclip = 0.)            Lower clipping factor in percent
    (shiclip = 0.)            Upper clipping factor in percent
```

```
    (snreject = 50)           Maximum number of sky fitting rejection iterations
   (sloreject = 3.)           Lower K-sigma rejection limit in sky sigma
   (shireject = 3.)           Upper K-sigma rejection limit in sky sigma
       (khist = 3.)           Half width of histogram in sky sigma
     (binsize = 0.1)          Binsize of histogram in sky sigma
      (smooth = no)           Boxcar smooth the histogram
       (rgrow = 0.)           Region growing radius in scale units
       (mksky = no)           Mark sky annuli on the display
        (mode = "ql")

photpars
   (weighting = "constant")   Photometric weighting scheme
   (apertures = "0.")         List of aperture radii in scale units
        (zmag = 25.)          Zero point of magnitude scale
     (mkapert = yes)          Draw apertures on the display
        (mode = "ql")

DAOFIND
        image = "NGC884-001NA.fits" Input image(s)
       output = "default"     Output coordinate file(s) (default: image.coo.?)
     (starmap = "")           Output density enhancement image(s)
      (skymap = "")           Output sky image(s)
    (datapars = "")           Data dependent parameters
    (findpars = "")           Object detection parameters
    (boundary = "nearest")    Boundary extension (constant|nearest|reflect|wrap)
    (constant = 0.)           Constant for boundary extension
 (interactive = no)           Interactive mode?
   (icommands = "")           Image cursor: [x y wcs] key [cmd]
   (gcommands = "")           Graphics cursor: [x y wcs] key [cmd]
      (wcsout = )_.wcsout)    The output coordinate system (logical,tv,physical)
       (cache = )_.cache)     Cache the image pixels?
      (verify = )_.verify)    Verify critical daofind parameters?
      (update = )_.update)    Update critical daofind parameters?
     (verbose = )_.verbose)   Print daofind messages?
    (graphics = )_.graphics)  Graphics device
     (display = )_.display)   Display device
        (mode = "ql")
```

PHOT

```
        image = "NGC869-001WB.fits" Input image(s)
       coords = "default"       Input coordinate list(s) (default: image.coo.?)
       output = "default"       Output photometry file(s) (default: image.mag.?)
      skyfile = ""              Input sky value file(s)
    (plotfile = "")             Output plot metacode file
    (datapars = "")             Data dependent parameters
  (centerpars = "")             Centering parameters
  (fitskypars = "")             Sky fitting parameters
    (photpars = "")             Photometry parameters
 (interactive = no)             Interactive mode?
    (radplots = no)             Plot the radial profiles?
   (icommands = "")             Image cursor: [x y wcs] key [cmd]
   (gcommands = "")             Graphics cursor: [x y wcs] key [cmd]
       (wcsin = )_.wcsin)       The input coordinate system (logical,tv,physical,world)
      (wcsout = )_.wcsout)      The output coordinate system (logical,tv,physical)
       (cache = )_.cache)       Cache the input image pixels in memory?
      (verify = yes)            Verify critical phot parameters?
      (update = )_.update)      Update critical phot parameters?
     (verbose = yes)            Print phot messages?
    (graphics = )_.graphics)    Graphics device
     (display = )_.display)     Display device
        (mode = "ql")
```

PSTSELECT

```
        image = "NGC869-001WB.fits" Image for which to build psf star list
     photfile = "default"       Photometry file (default: image.mag.?)
      pstfile = "default"       Output psf star list file (default: image.pst.?)
      maxnpsf = 50              Maximum number of psf stars
     (mkstars = yes)            Mark deleted and accepted psf stars
    (plotfile = "")             Output plot metacode file
    (datapars = "")             Data dependent parameters
     (daopars = "")             Psf fitting parameters
 (interactive = no)             Select psf stars interactively?
    (plottype = "mesh")         Default plot type (mesh|contour|radial)
   (icommands = "")             Image cursor: [x y wcs] key [cmd]
   (gcommands = "")             Graphics cursor: [x y wcs] key [cmd]
       (wcsin = )_.wcsin)       The input coordinate system (logical,tv,physical,world)
```

```
      (wcsout = )_.wcsout)      The output coordinate system (logical,tv,physical)
       (cache = )_.cache)       Cache the input image pixels in memory?
      (verify = )_.verify)      Verify critical pstselect parameters?
      (update = )_.update)      Update critical pstselect parameters?
     (verbose = )_.verbose)     Print pstselect messages?
    (graphics = )_.graphics)    Graphics device
     (display = )_.display)     Image display device
        (mode = "ql")
```

PSF

```
        image = "NGC869-001WB.fits" Input image(s) for which to build PSF
     photfile = "default"       Input photometry file(s) (default: image.mag.?)
      pstfile = "default"       Input psf star list(s) (default: image.pst.?)
     psfimage = "default"       Output PSF image(s) (default: image.psf.?)
     opstfile = "default"       Output PSF star list(s) (default: image.pst.?)
    groupfile = "default"       Output PSF star group file(s) (default: image.psg.?)
    (plotfile = "")             Output plot metacode file
    (datapars = "")             Data dependent parameters
     (daopars = "")             Psf fitting parameters
   (matchbyid = yes)            Match psf star list to photometry file(s) by id number?
 (interactive = no)             Compute the psf interactively?
     (mkstars = no)             Mark deleted and accepted psf stars?
   (showplots = yes)            Show plots of PSF stars?
    (plottype = "mesh")         Default plot type (mesh|contour|radial)
   (icommands = "")             Image cursor: [x y wcs] key [cmd]
   (gcommands = "")             Graphics cursor: [x y wcs] key [cmd]
       (wcsin = )_.wcsin)       The input coordinate system (logical,tv,physical,world)
      (wcsout = )_.wcsout)      The output coordinate system (logical,tv,physical)
       (cache = )_.cache)       Cache the input image pixels in memory?
      (verify = )_.verify)      Verify critical psf parameters?
      (update = )_.update)      Update critical psf parameters?
     (verbose = )_.verbose)     Print psf messages?
    (graphics = "stdgraph")     Graphics device
     (display = )_.display)     Display device
        (mode = "ql")
```

GROUP

```
        image = "NGC869-001WB.fits" Image corresponding to the photometry file
```

```
     photfile = "default"       Photometry file (default: image.mag.?)
     psfimage = "default"       PSF image (default: image.psf.?)
    groupfile = "default"       Output group file (default: image.grp.?)
    (datapars = "datapars")     Data dependent parameters
     (daopars = "daopars")      Psf fitting parameters
       (wcsin = )_.wcsin)       The input coordinate system (logical,tv,physical,world)
      (wcsout = )_.wcsout)      The output coordinate system (logical,tv,physical)
      (wcspsf = )_.wcspsf)      The psf coordinate system (logical,tv,physical)
       (cache = )_.cache)       Cache the input image pixels in memory?
      (verify = )_.verify)      Verify critical group parameters?
      (update = )_.update)      Update critical group parameters?
     (verbose = )_.verbose)     Print group messages?
        (mode = "ql")
```

SEEPSF

```
     psfimage = "NGC869NA.fits.psf.1.fits" PSF image name
        image = "na"            Output image name
   (dimension = INDEF)          Dimension of the output PSF image
        (xpsf = INDEF)          X distance from the PSF star
        (ypsf = INDEF)          Y distance from the PSF star
   (magnitude = INDEF)          Magnitude of the PSF star
        (mode = "ql")
```

NSTAR

```
        image = "NGC869-001WB.fits" Image corresponding to photometry
    groupfile = "default"       Input group file (image.grp.?)
     psfimage = "default"       PSF image (default: image.psf.?)
    nstarfile = "default"       Output photometry file (default: image.nst.?)
      rejfile = "default"       Output rejections file (default: image.nrj.?)
    (datapars = "datapars")     Data parameters
     (daopars = "daopars")      Psf fitting parameters
       (wcsin = )_.wcsin)       The input coordinate system (logical,tv,physical,world)
      (wcsout = )_.wcsout)      The output coordinate system (logical,tv,physical)
      (wcspsf = )_.wcspsf)      The psf coordinate system (logical,tv,physical)
       (cache = )_.cache)       Cache the input image pixels in memory?
      (verify = )_.verify)      Verify critical nstar parameters?
      (update = )_.update)      Update critical nstar parameters?
     (verbose = )_.verbose)     Print nstar messages?
```

```
        (mode = "ql")

ALLSTAR
        image = "NGC869-001WB.fits" Image corresponding to photometry
     photfile = "default"       Input photometry file (default: image.mag.?)
     psfimage = "default"       PSF image (default: image.psf.?)
  allstarfile = "default"       Output photometry file (default: image.als.?)
      rejfile = "default"       Output rejections file (default: image.arj.?)
     subimage = "default"       Subtracted image (default: image.sub.?)
    (datapars = "")             Data dependent parameters
     (daopars = "")             Psf fitting parameters
       (wcsin = )_.wcsin)       The input coordinate system (logical,tv,physical,world)
      (wcsout = )_.wcsout)      The output coordinate system (logical,tv,physical)
      (wcspsf = )_.wcspsf)      The psf coordinate system (logical,tv,physical)
       (cache = yes)            Cache the data in memory?
      (verify = )_.verify)      Verify critical allstar parameters?
      (update = )_.update)      Update critical allstar parameters?
     (verbose = )_.verbose)     Print allstar messages?
     (version = 2)              Version
        (mode = "ql")

login.cl
# LOGIN.CL -- User login file for the IRAF command language.

# Identify login.cl version (checked in images.cl).
if (defpar ("logver"))
    logver = "IRAF V2.14.1 September 2008"

set        home                = "/home/andyh1568/"
set        imdir                = "HDR$/"
set        uparm                = "home$uparm/"
set        userid                = "andyh1568"

# Set the terminal type.  We assume the user has defined this correctly
# when issuing the MKIRAF and no longer key off the unix TERM to set a
# default.
if (access (".hushiraf") == no)
    print "setting terminal type to xgterm..."
```

```
stty xgterm


# Uncomment and edit to change the defaults.
#set       editor            = vi
#set       printer           = lp
#set       pspage            = "letter"
#set       stdimage          = imt2048
#set       stdimcur          = stdimage
#set       stdplot           = lw
#set       clobber           = no
#set       filewait          = yes
#set       cmbuflen          = 512000
#set       min_lenuserarea   = 64000
#set       imtype            = "imh"
set        imextn            = "oif:imh fxf:fits,fit,fts fxb:fxb plf:pl qpf:qp stf:hhh,??h"


# XIMTOOL/DISPLAY stuff.  Set node to the name of your workstation to
# enable remote image display.  The trailing "!" is required.
#set       node              = "my_workstation!"

# CL parameters you mighth want to change.
#ehinit   = "nostandout eol noverify"
#epinit   = "standout showall"
showtype = yes


# Load the default CL package.  Doing so here allows us to override package
# paths and load personalized packages from our loginuser.cl.
clpackage


# Default USER package; extend or modify as you wish.  Note that this can
# be used to call FORTRAN programs from IRAF.

package user
```

```
task        $adb $bc $cal $cat $comm $cp $csh $date $dbx $df $diff        = "$foreign"
task        $du $find $finger $ftp $grep $lpq $lprm $ls $mail $make       = "$foreign"
task        $man $mon $mv $nm $od $ps $rcp $rlogin $rsh $ruptime        = "$foreign"
task        $rwho $sh $spell $sps $strings $su $telnet $tip $top        = "$foreign"
task        $vi $emacs $w $wc $less $rusers $sync $pwd $gdb                = "$foreign"

task        $xc $mkpkg $generic $rtar $wtar $buglog                        = "$foreign"
#task        $fc = "$xc -h $* -limfort -lsys -lvops -los"
task        $fc = ("$" // envget("iraf") // "unix/hlib/fc.csh" //
            " -h $* -limfort -lsys -lvops -los")
task        $nbugs = ("$(setenv EDITOR 'buglog -e';" //
            "less -Cqm +G " // envget ("iraf") // "local/bugs.*)")
task        $cls = "$clear;ls"
task        $clw = "$clear;w"
task        $pg = ("$(less -Cqm $*)")

task    $sed = "$foreign"
task    $nightphot4 = /home/andyh1568/programs/nightphot4.cl
task    $varstar =$/home/andyh1568/programs/varstar
task        $wmodaophot2 = /home/andyh1568/programs/wmodaophot2.cl

if (access ("home$loginuser.cl"))
    cl < "home$loginuser.cl"
;

keep



prcache directory
cache   directory page type help

# Print the message of the day.
if (access (".hushiraf"))
    menus = no
else {
    clear; type hlib$motd
}
```

```
# Delete any old MTIO lock (magtape position) files.
if (deftask ("mtclean"))
    mtclean
else
    delete uparm$mt?.lok,uparm$*.wcs verify-

# List any packages you want loaded at login time, ONE PER LINE.
images          # general image operators
plot            # graphics tasks
dataio          # data conversions, import export
lists           # list processing

# The if(deftask...) is needed for V2.9 compatibility.
if (deftask ("proto"))
    proto       # prototype or ad hoc tasks

tv              # image display
utilities       # miscellaneous utilities
noao            # optical astronomy packages

keep
```

WMODAO

```
# WMODAO -- Fix West Mountain Headers for DAO work
# Edited by Andy Hernandez Nov 2017 from notes I made in 2013 for H and Chi Persei

#note that this modified file will only work if you add the following line to login.cl
#task $sed = "$foreign"

procedure wmodao (images)

string images               {prompt="Root name of images to fix"}
struct *list, *list2

begin
        # Local variables
        string imagelist, imagenst, imagenrj
        string img, output
```

```
 string imgroot
string coordfile, tmp5
int    i, end1, end2, tmp1, tmp2, tmp3, tmp4
real   temp, FWHM, temp2, temp3, intfwhm, inthwhm, STDE, PSFRAD, GDMX
real   fwb, sigb, psf, gdmx

# Create a text file list of images to phot.
imagelist = mktemp ("tmp$night")
imgroot = images
#sections (images//"*", > imagelist)
sections ("*.fits,*.fit", > imagelist)

# Open the list of images and scan through it.
list = imagelist
#coordfile = center_file
while (fscan (list, img) != EOF) {
    print (img)
    print ("Point to open spots and press (m). Do 11-15 spots(odd number so as to be statistically more accurate).")
    print ("Point to 11-15 stars and press (a). Again choose an odd number of unsaturated stars.")
    print ("Point to the same stars and press (r): Depending on the parameters of rimexam,
this will give you both the PSF and the Good Data Max for the field.")
    imexamine (img, 1)
    print ("Input the FWHM for this frame.")
            scanf ("%f", FWHM)
    print ("Input the average STDEV of the Background.")
    scanf ("%f", STDE)
         print ("Input the PSF for the stars in the field.")
    scanf ("%f", PSFRAD)
        print ("Input the Good Data Max for the frame.")
            scanf ("%f", GDMX)
    fwb = FWHM
    sigb = STDE
        psf = PSFRAD
        gdmx = GDMX
# set parameters based on value of FWHM and the deviation in the sky
    centerpars.maxshift=2*fwb
    centerpars.cbox=2.5*fwb
    daopars.matchra=3*fwb
```

```
            daopars.psfrad=psf
            daopars.fitrad=1.4*fwb
            daopars.sannulu=3.5*fwb
            daopars.wsannul=fwb+0.5
            datapars.fwhmpsf=fwb
            datapars.sigma=sigb
                        datapars.datamax=gdmx
            fitskypars.annulus=4*fwb
            fitskypars.dannulu=fwb+0.5
            photpars.aperture=1.3*fwb
            photpars.zmag=25
        # Run Photometry with default values
            phot (img, "", coords="default", output="default", verify=no, update=yes, verbose=no)
            print ("Phot ", img, " done")
        # Select PSF Stars
            pstselect (img, photfile="default", pstfile="default", verbose=no, maxnpsf=50)
            print ("PSTSELECT ", img, " done")
        # Make PSF
            psf (img, photfile="default", pstfile="default", psfimage="default",
opstfile="default", groupfil="default", verbose=no)
            print ("PSF ", img, " done")
        # Group stars
            group (img, photfile="default", psfimage="default", groupfil="default", verbose=no)
            print ("Group ", img, " done")
        # Run Point Spread Function Photometry
            nstar (img, groupfil="default", psfimage="default", nstarfil="default", rejfile="default", verbose=no)
            print (img, " completely done. TXDumping.")
        # Run Point Spread Function Photometry Allstar
            allstar (img, photfil="default", psfimage="default", allstarf="default", rejfile="default", verbose=no)
            print (img, " completely done. TXDumping.")
    # TXDump command that compiles the magnitudes
        txdump (img//"*.nst.1", "id,mag", "yes", headers=no, parameters=yes, > img//".txt")
            txdump (img//"*.nrj.1", "id,mag", "yes", headers=no, parameters=yes, >> img//".txt")
                txdump (img//"*.als.1", "id,mag", "yes", headers=no, parameters=yes, > img//"als.txt")
            txdump (img//"*.arj.1", "id,mag", "yes", headers=no, parameters=yes, >> img//"als.txt")
        }
end
```

Here will go table including the magnitudes and their zero point. This might be a very large table including over 500 objects in which I would imagine that a multi-columned table would look best. The zero-pointing process is done referencing work done on M 67 and NGC 752 by Joner & Hintz (2015).

Bibliography

Boyer, A. N. M., McSwain, M. V., Aragona, C., & Ou-Yang, B. 2012, Astronomical Journal, 144, 158, pT: J; UT: WOS:000311165400001

Crawford, D. L., & Mander, J. 1966, Astronomical Journal, 71, 114

Currie, T., et al. 2010, Astrophysical Journal Supplement Series, 186, 191, pT: J; UT: WOS:000274283900002

Joner, M. D., & Hintz, E. G. 2015, Astronomical Journal, 150, 204

Currie, T. and Kenyon, S. J. and Balog, Z. and Rieke, G. and Bragg, A. and Bromley, B. 2008, Astrophysical Journal, 672, 558

S. C. Keller and E. K. Grebel and G. J. Miller and K. M. Yoss 2001, Astronomical Journal, 122, 1