NEW APERTURE PHOTOMETRY TECHNIQUE FOR BRIGHAM YOUNG UNIVERSITY ASTRONOMY GROUP

by

E. Paul Iverson

A senior thesis submitted to the faculty of

Brigham Young University

in partial fulfillment of the requirements for the degree of

Bachelor of Science

Department of Physics and Astronomy

Brigham Young University

April 2008

BRIGHAM YOUNG UNIVERSITY

DEPARTMENT APPROVAL

of a senior thesis submitted by

E. Paul Iverson

This thesis has been reviewed by the research advisor, research coordinator, and department chair and has been found to be satisfactory.

Date
Michael Joner, Advisor

Date
Eric Hintz, Research Coordinator

Date
Ross Spencer, Department Chair

ABSTRACT

NEW APERTURE PHOTOMETRY TECHNIQUE FOR BRIGHAM YOUNG UNIVERSITY ASTRONOMY GROUP

E. Paul Iverson

Department of Physics and Astronomy

Senior Thesis

This thesis outlines the development and application of a new flux technique of aperture photometry under consideration at Brigham Young University (BYU). Previous to this method, researchers used a set aperture method to determine a star's instrumental magnitude. This method does not compensate for changing atmospheric conditions nor does it optimize itself to the data. The new approach applies a separate aperture per point source. The radius of this aperture is determined by a user-defined percentage of the flux of the object's point spread function (PSF). Apertures are, therefore, optimized for signal and changing atmospheric or telescope conditions and all are forced onto the same magnitude scale. This makes it possible to glean more accurate results from all data. Furthermore, the method is contained within a script that eliminates the need for excessive user interaction. The variable star CY Aquarii and the standard cluster M67 are used to demonstrate the application of this flux method.

ACKNOWLEDGMENTS

The impact of Professor Michael D. Joner's extensive help throughout this project cannot be overstated. His *ten1headfix.cl* script was the progenitor of everything we have written, both SIDAP and *ap_apphot*. His patience, guidance, and knowledge have been indispensable.

This project also would have been impossible without my colleague and friend Craig Swenson. His help in writing and revising SIDAP and ap_apphot have helped them become much better products than they would have been otherwise.

I would also like to acknowledge the help of the following individuals: Frank Valdes and Mike Fitzpatrick at NOAO for their help understanding IRAF and how to program in the SPP environment, Lisa Joner for her hours spent editing this thesis, Ali Porter and Jeremy Schoonmaker for their beta testing, and my wife Melanie for her consistent encouragement and support.

Contents

List of Tables

List of Figures

Chapter 1

Introduction and Background

1.1 Principles of Aperture Photometry

Aperture photometry is the most straightforward method of obtaining the stellar magnitude of a point source, or stellar object, from an astronomical image. However, despite its simplicity, many different techniques exist to perform aperture photometry. Although all are unified by the same basic idea of counting all flux through an aperture of some particular size, the methods used can differ substantially. Here, a simple presentation is given of the basic standard characteristics as well as sources of error common to all methods of aperture photometry.

1.1.1 Standard Process of Aperture Photometry

After astronomical data have been processed (i.e., bias-, dark-, and flat-corrected), the instrumental magnitude of a point source, or stellar object, can be determined from the exposed image (Section 1.1, Swenson 2008).

A software aperture of a particular radius is centered over the object of interest, the total flux within that radius is counted, and a total flux profile is produced. This profile represents the combination of the total flux profile of the object itself, together with non-zero background contamination from other luminous objects or diffuse sources. Luminous objects that may contaminate the profile include other nearby stars (either detected or not) or background galaxies. Diffuse sources augmenting the contamination may include terrestrial night sky emission, zodiacal light, nebulae, etc. (Da Costa 1992). In order to obtain accurate instrumental magnitudes from the total flux profile, this background noise must be removed. Since the particular background contamination cannot be exactly determined, it must be estimated by calculating the noise of a neighboring region centered on the object. This is done by setting an annulus, or ring, around the point source and then calculating the total flux within that annulus. We assume that this total flux count does not contain any contribution from the object itself. This value is then subtracted from each pixel in the combined, contaminated profile to attain the "uncontaminated" stellar profile.

With this "uncontaminated" profile, the instrumental magnitude can now be determined. The profile is integrated to obtain the total flux of the object. This value is then entered into the following equation to determine the instrumental magnitude:

$$mag = zmag - 2.5 * log10(totalflux) + 2.5 * log10(itime). \tag{1.1}$$

Note that this equation takes into account the user-defined zero magnitude and the integration time (i.e., exposure time) of the image when calculating the instrumental magnitude.

1.1.2 Aperture Size Determination in Aperture Photometry Process

Ideally, all background contamination is eliminated before the total flux is determined, thereby resulting in an exact instrumental magnitude determination. In reality, complete removal of all noise in an image is simply not possible and some quantifiable error remains in the final magnitude results. The source of this error generally stems from the choice of software aperture, but can also be due to errors in prior processing (Section 3.1, Swenson 2008).

When determining the proper aperture size to use for a star, it is common to look at an image and think that the ideal choice would be the aperture that contains all of the light seen at the star's location. This misconception can be further amplified when using display programs, such as ds9, that can give the impression of actual spatial resolution. In reality, when examining a stellar profile, it is clear that the aperture would have to be very, very large in order to contain all of the light, since the profile technically never goes to zero at any point (i.e., an aperture of ∞). Additionally, since it is not possible to remove all noise, an increasingly large aperture contains an increasingly large amount of background contamination. For a bright star, this large degree of background noise generally would not be detrimental to magnitude determinations because the signal from the source itself would, hopefully, be several orders of magnitude greater than the contamination. However, for a faint star, high contamination could be the whole difference between good, reliable results and results for the garbage can. Even with bright stars, the increased contamination leads to larger error and limits the possible interpretations that can be gleaned from the results. Essentially, these arguments narrow the choice of aperture size into two distinct cases (Da Costa 1992):

1. "Bright" sources with which large apertures, containing all the light, are acceptable but which contain relatively insignificant contamination, and
2. "Faint" sources with which small apertures, which don't contain all the light, are preferred.

1.2 Traditional Aperture Photometry Method Used in the Field

In ideal practice, the choice of aperture size has required an amalgamation of two different sizes for faint sources and the larger of the two sizes for bright sources. This is due to the need to normalize both faint and bright sources to the same magnitude scale while maintaining high signal-to-noise (S/N) and low error. The magnitude of a bright source is determined using a large aperture, as in case 1. The magnitude of a faint source is then determined using a smaller aperture, as in case 2, and the same large aperture used for the bright object. In this way both the faint and the bright source have S/N optimized. An aperture correction is then applied to the faint source to obtain an accurate magnitude on the same magnitude scale as the bright source. This correction involves taking the magnitude difference between the large and small apertures and adding it to the small aperture magnitude. This results in the collection of the same percentage of light from the faint star as that collected from the large aperture on the bright star, while maintaining relatively the same degree of error (Da Costa 1992, Howell 1992). The size of the small aperture is typically determined as that which collects the majority of the light of the faintest object of interest. Obviously, when this aperture size is used with the brightest objects of interest, some portion less than the majority of the light is collected. The size of the large aperture is such that nearly all the light is collected from the brightest objects independent of telescope factors, such as focus, and atmospheric effects like seeing or zodiacal light (Howell 1992).

Although it has been presented here as performing photometry with two different set apertures on the same frame, aperture corrections are generally used, when doing absolute photometry, to bring fainter program stars from a program frame up to the same magnitude scale as standardized comparison stars from a standard frame.

1.3 Traditional Aperture Photometry Method Used at Brigham Young University

Astronomers at Brigham Young University (BYU) traditionally take a simplified approach to performing aperture photometry. When obtaining data, integration times are pre-determined to ensure that the object of interest can always be studied as a "bright" source (as in case 1 in the choice of aperture size) with high S/N. Instrumental magnitudes are therefore calculated using a large aperture that collects all of the light. Since the output of the photometry is studied differentially at BYU, aperture corrections are not needed in order to obtain decently reliable final results.

Specifically, IRAF parameter values traditionally used at BYU during the *phot* command are listed in Table 1.1. These are the parameters set after running the *newton.cl* script.

1.4 Reasons for Modification of Historical Photometry Method

The traditional photometry method used at BYU has proven acceptable in many cases; however, it contains several distinct limitations. These constraints are described below:

Table 1.1. Traditional Parameters and Values Used With the IRAF *phot* Command

IRAF Package.Param	Value	IRAF Package.Param	Value
centerpars.calgorithm	centroid	fitskypars.salgorithm	ofilter
centerpars.cbox	13.00	fitskypars.annulus	13.00
centerpars.cthreshold	0.75	fitskypars.dannulus	4.00
centerpars.minsnratio	1.00	fitskypars.skyvalue	0.00
centerpars.cmaxiter	10.00	fitskypars.smaxiter	10.00
centerpars.maxshift	8.50	fitskypars.sloclip	0.00
centerpars.clean	no	fitskypars.shiclip	10.00
centerpars.rclean	1.00	fitskypars.snreject	50.00
centerpars.rclip	2.00	fitskypars.sloreject	3.00
centerpars.kclean	3.00	fitskypars.shireject	3.00
centerpars.mkcenter	no	fitskypars.khist	3.00
datapars.scale	1.00	fitskypars.binsize	0.10
datapars.fwhmpsf	3.00	fitskypars.smooth	no
datapars.emission	yes	fitskypars.rgrow	0.00
datapars.sigma	INDEF	fitskypars.mksky	no
datapars.datamin	INDEF	photpars.weighting	constant
datapars.datamax	INDEF	photpars.apertures	13.00
datapars.noise	poisson	photpars.zmag	20.00
datapars.ccdread	rdnoise	photpars.mkapert	no
datapars.gain	gain	...	...
datapars.readnoise	INDEF	...	...
datapars.epadu	1.00	...	...
datapars.exposure	exptime	...	...
datapars.airmass	airmass	...	...
datapars.filter	subset	...	...
datapars.obstime	hjd	...	...

Limited Magnitude Range Large apertures limit research to only the brightest, resolvable stars in any frame and set a narrow window of magnitudes, based on integration time and telescope size, for objects that can be reliably studied. This is due to the fact that large apertures, although satisfactory for bright, high-response objects (i.e., high S/N), collect too much noise in order to reliably study fainter, lower response objects in the same frame (i.e., low S/N in large apertures).

Unnecessarily Large Magnitude Error Set apertures do nothing to eliminate excess contamination from large aperture use or optimize S/N for each point source. This adds an unnecessarily large error into any magnitude determination.

Incorrect Assumption of a Constant Point-Spread Function (PSF) The basic foundation of aperture photometry relies on the assumption that the PSF is constant over the entire frame and, to a first approximation, it is. This assumption is necessary in order to accept that a large set aperture will collect the same percentage of light regardless of the point sources being studied. However, in reality, no image has an exactly constant PSF (not even the Hubble Space Telescope). This introduces a small, but significant, error when comparing stars since, without performing some aperture correction, each star is on a slightly different magnitude scale.

Although, to first order, these limitations result in acceptable error levels, these results are of lower quality than can be achieved with improved methods. Furthermore, since no single image has exclusively bright objects or faint objects, they limit what can be gleaned as reliable results from any particular image.

1.5 Early Deterrents for Modification of Historical Photometry Method

The use of the traditional photometry method at BYU, without modification to correct limitations, persisted due to two major deterrents to change. These impediments were the following: the time requirement of pre-photometry data reduction and the use of IRAF as a "black box".

The first major impediment was the amount of time required to perform the pre-photometry reduction of large data sets. A single night's worth of observing can yield several thousand frames that need to be reduced and then individually have some photometric reduction method applied to them. Because the traditional reduction process was not automated, it was often necessary to devote large amounts of time to simply preparing data for photometric measurement. This commitment made it impractical to use anything but the well established traditional photometry method in order to finish analysis in a reasonable time frame.

The second major impediment was the treatment of the Image Reduction and Analysis Facility (IRAF) software package as a "black box". IRAF is a very large package of commands which can be daunting to users seeking to navigate and fully utilize the program. Thus, rote data reduction and photometry techniques were established and distributed (I also am guilty of this). This type of approach is one that Dr. Benjamin Taylor would call a “cookie cutter” means to analysis that allows the lumpen proletariate (i.e., BYU) to prostrate themselves in reverential awe before the god Rote Tradition (i.e., the traditional photometry method). These memorized lists of commands have inadvertently negated full understanding of IRAF and its capabilities, without which it was impossible for users to determine needed corrections to fix the method’s limitations.

Both the issues of time commitment and IRAF use were overcome with the creation of a scripted analysis data-pipeline by Craig Swenson and Paul Iverson (Swenson 2008). The Swenson-Iverson Data Analysis Pipeline (SIDAP) resolves the first impediment by giving astronomers at BYU the ability to reduce thousands of frames in a single day, thus completely eliminating the large time requirement of data reduction. The amount of time saved has not yet been quantified, but it is reasonable to assume that the time requirement has been reduced by approximately 30-50%. The time reduction would be considerably larger but SIDAP has also added several new features that facilitate analysis. As a solution to the second impediment, the scripting of SIDAP required a more extensive understanding of IRAF that departed greatly from the rote methods of analysis previously understood. Thus, while creating SIDAP, the understanding needed of IRAF and its capabilities was achieved.

With a reduced time requirement and greater knowledge of IRAF, it became possible to begin exploring how to best correct the limitations of the traditional photometry method used at BYU. This process eventually developed into an attempt to create an ideal photometry technique.

1.6 Basic Requirements of New Photometry Method

In order to form an ideal aperture photometry technique, it was first necessary to outline its basic capabilities. Not all of these requirements were delineated from the outset, but each one became an integral part as our new photometry method developed. The basic abilities include:

1. The ability to optimize aperture size by target and thus increase S/N;

2. Allowing for changing atmospheric seeing or focus conditions; and

3. Correcting for minute, but significant, changes in the PSF in order to normalize all objects to the same magnitude scale.

Chapter 2

ap_apphot Script Development and Application

2.1 *ap_apphot* Script Development

The development of our ideal aperture photometry method (*ap_apphot*) progressed through several permutations before arriving at its current state. Even the basic language in which it was written evolved from simple IRAF CL to a concoction of SPP procedure language, IRAF CL language, and individual, imported Unix commands. To understand the script and its abilities and limitations, it is helpful to have some knowledge of the steps of its development. Hopefully, this will help the next generation to further improve on the *ap_apphot* photometry technique and avoid some of the problems encountered in its development.

Each individual version is not presented here. That would be astoundingly impractical. Instead, only the basic ideas behind the three most fundamental changes are discussed. These basic permutations represent major milestones in the development of *ap_apphot*. Specifically, the three changes were:

1. The use of smaller set apertures,

2. The use of smaller set apertures based on changing seeing conditions seen in the measurement of the average frame full-width-at-half-maximum (FWHM), and

3. The use of individual, variable apertures for each star determined by the user-desired percentage of the star's total flux.

Obviously, these three changes stemmed specifically from the basic abilities of an ideal photometry method mentioned in the previous chapter. However, it is important to note that, as also mentioned previously, not all of the basic abilities required were initially identified. In fact, some were not even fully understood until this paper was being prepared. Nevertheless, the incipient goal has essentially remained the same and is actually rather simple: reduce aperture size and consequently increase S/N. It is safe to assume that all subsequent variations of *ap_apphot* evolved from and built upon this basic idea.

Comparative results between each permutation and the traditional photometry technique used at BYU are discussed in the next chapter.

2.1.1 Small Aperture-based *ap_apphot*

The first permutation of *ap_apphot* focused solely on using a reduced aperture size that could still collect all the light from the point source. This was based on the assumption that a smaller aperture could be used that would successfully eliminate noise while maintaining good results (i.e., increase S/N by lowering N). At this stage the effects of changing atmospheric seeing or telescope focus were not considered, but rather both were treated as essentially constant. This assumption required that, although the apertures could be reduced in size from the traditional method, there was a sizable limit to that reduction since, in actuality, neither atmospheric seeing nor telescope focus are exactly constant within a given frame. In order to evaluate this technique, the S/N and standard deviation of objects in the frame were observed while reducing the aperture size being used. The aperture radii studied ranged from 6 pixels to 13 pixels (the radius size traditionally used at BYU).

This technique proved successful in achieving our modest initial goal of increasing S/N; however, it still did not meet all the requirements of an ideal photometry method nor did it achieve final results that were completely satisfying. Although this initial form of *ap_apphot* is no longer utilized, its remains are located in the *ap_reg_apphot* script (Section 2.2.13, Swenson 2008).

2.1.2 FWHM-based *ap_apphot*

The second permutation of *ap_apphot* expanded on the idea of using smaller apertures but incorporated seeing and focus instabilities. This version formed from the idea that the best S/N occurs at a diameter of 1 FWHM (Howell 1992). This was a considerably smaller aperture (approximately 3-6 pixel diameters on average) than had been used in the previous version of *ap_apphot*. The previous necessity to collect all of the light from the point source remained in order to maintain all stars on the same magnitude scale; however, this introduced a problem when using very, very small apertures. For the first time, the assumption of treating both atmospheric seeing and telescope focus as constant began to break down. With a large aperture, deviations in seeing or focus are a non-issue since all deviations still remain within the aperture itself, but with aperture sizes approximately 1 FWHM, deviations can have a substantial impact on results. In some cases, it is possible for stars to "bloat" beyond the set aperture on nights of poor seeing or when utilizing a telescope with focus problems. This can result in the collection of portions of light rather than all the light from the object, placing the objects on different magnitude scales and making comparisons between objects statistically impossible. To resolve this issue, this version of *ap_apphot* calculated an average FWHM for the frame, determined a user-defined multiple of that amount as the aperture size, and then applied it to each object. With the ability to compensate for changing seeing and focus issues, the multiple of the FWHM used for aperture size was reduced while simultaneously observing the S/N and standard deviation of objects in the frame. The aperture radii studied had FWHM multiples of 1.0, 1.5, 2.0, 2.5, and 3.0.

This method was problematic and relied heavily on an accurate calculation of the frame's FWHM. Despite this, results initially seemed promising but with further analysis proved to be inferior. The remains of this approach are located in the *ap_reg_apphot* script (Section 2.2.13, Swenson 2008).

2.1.3 Flux-based *ap_apphot*

While the previous versions of *ap_apphot* were, in essence, simple modifications to the historical technique, the third and current permutation was a dramatic modification. It merged the idea of smaller, script-calculated apertures together with the idea of aperture corrections. This merger essentially optimizes S/N while maintaining all objects being studied on the same magnitude scale. In order to accomplish this, the current version of *ap_apphot* calculates an individual PSF for each object and determines an individual aperture size based on a user-defined percentage of the total flux from the PSF. Whereas previous attempts still involved applying the same sized aperture to all stars within the same frame, this new method measures and determines the correct aperture for each individual source of interest within the same frame. With the ability to compensate for changing seeing and focus issues while simultaneously eliminating the need for aperture corrections, the percentage of the flux used to calculate individual aperture sizes was optimized while observing the S/N and standard deviation of objects in the frame. Apertures were studied with radii based on 80%, 85%, 90%, and 95% of the total flux.

Specifically, IRAF parameter values set during the *ap_apphot* script are listed in Table 2.1.

Table 2.1. Traditional Parameters and Values Used With the IRAF *phot* Command

IRAF Package.Param	Value	IRAF Package.Param	Value
centerpars.calgorithm	centroid	fitskypars.salgorithm	mode
centerpars.cbox	3*(Average FWHM)	fitskypars.annulus	10*(Average FWHM)
centerpars.cthreshold	0.75	fitskypars.dannulus	(Necessary for min of 500 pixels)
centerpars.minsnratio	1.00	fitskypars.skyvalue	0.00
centerpars.cmaxiter	20.00	fitskypars.smaxiter	10.00
centerpars.maxshift	8.50	fitskypars.sloclip	0.00
centerpars.clean	no	fitskypars.shiclip	10.00
centerpars.rclean	1.00	fitskypars.snreject	50.00
centerpars.rclip	2.00	fitskypars.sloreject	3.00
centerpars.kclean	3.00	fitskypars.shireject	3.00
centerpars.mkcenter	no	fitskypars.khist	3.00
datapars.scale	1.00	fitskypars.binsize	0.10
datapars.fwhmpsf	(Average FWHM)	fitskypars.smooth	no
datapars.emission	yes	fitskypars.rgrow	0.00
datapars.sigma	(Average sky deviation)	fitskypars.mksky	no
datapars.datamin	INDEF	photpars.weighting	constant
datapars.datamax	INDEF	photpars.apertures	(Set to flux percentage)
datapars.noise	poisson	photpars.zmag	30.00
datapars.ccdread	rdnoise	photpars.mkapert	no
datapars.gain	gain	...	...
datapars.readnoise	(CCD readnoise)	...	...
datapars.epadu	(CCD gain)	...	...
datapars.exposure	exptime	...	...
datapars.airmass	airmass	...	...
datapars.filter	subset	...	...
datapars.obstime	hjd	...	...

This script also creates .info files that track aperture sizes used for each individual star as well as the average FWHM of the frame.

2.2 Current *ap_apphot* Script Abilities and Limitations

The current flux version of the *ap_apphot* script has several clear advantages over the previous methods utilized at BYU despite its reliance on an accurate determination of individual PSF. These advantages are enumerated below:

1. Improved statistically accurate photometry,

2. Increased range of magnitudes that can be reliably studied in a single frame,

3. Optimized scaling of all objects of interest to the same magnitude scale,

4. Minimized user involvement in the photometry process, and

5. Enhanced "book keeping" for debugging purposes.

However, the *ap_apphot* script does have a few possible limitations:

1. It requires the use of SIDAP for data reduction,
2. It relies on an accurate measure of the PSF, and
3. It is unable to do piece-meal photometry on a single night (i.e., it must do an entire night each run and cannot begin halfway through).

2.3 *ap_apphot* Application Guide

The *ap_apphot* script is designed to replace the IRAF *phot* command and to function with minimal user interaction. The actual code is relatively simple and can be understood with little prior knowledge outside of basic programming syntax and IRAF commands. Nevertheless, from experience we know that it can be daunting to a new user to navigate. Therefore, a short guide to its use is presented here. It is important to note that all frames must be run through SIDAP previous to the application of *ap_apphot* (Swenson 2008).

2.3.1 *ap_apphot* User Prompts

The only user inputs required by the *ap_apphot* script are five initial prompts to determine the specific parameters of the command itself. These can be set using the IRAF *epar* command (i.e., *epar ap_apphot_v_*∗), or the user can set each parameter by way of the prompts each time the script is run. The initial prompts are as follows:

i. File Prompt: The user is prompted for a search string describing the files to be imported into *ap_apphot*. This string needs to be sufficiently descriptive as to allow only the desired files through. A string like “a-*.fits” would successfully import only the aligned .fits files coming from SIDAP.

ii. Coordinate Prompt: The user is prompted for the name of the coordinate file to be used for photometry. This file should be the x and y coordinates of the objects to be studied. If using ds9 to mark stars for photometry (as is true for BYU), this file will have the extension “.reg”.

iii. Technique Prompt: The user is prompted to enter the desired photometry method, either FWHM or flux. Although this thesis only introduces the flux technique contained in *ap_apphot*, the script also has the ability to determine aperture sizes based on FWHM measurements from individual PSF determinations. However, it should be noted that the FWHM method has not been tested and the accuracy of its results cannot be confirmed, other than judgements based on the coherence of data sets reduced using this technique.

iv. Aperture Factor Prompt: Depending on the photometry technique (flux or FWHM), the user is prompted for the flux percentage or the FWHM multiple to use when determining individual aperture sizes.

v. By Pier Prompt: The user is prompted to enter whether or not the photometry should be performed by pier side. If a known magnitude gradient exists for the CCD being used (a zero point shift when going "over-the-pier" strongly suggests a gradient) or if the telescope requires an "over-the-pier" move, doing photometry by pier side is highly suggested.

2.3.2 *ap_apphot* Automation

After user-defined parameters have been set, the *ap_apphot* script is completely automated. Only the flux method is presented here, since the FWHM technique differs only in the aperture factor prompt and has not yet been tested. The algorithm completes the following twelve actions in order:

i. Initial Data Extraction: The following fields are extracted from each file header: *file name*, *FWHM*, *gain*, *sky deviation*, *HJD*, *readnoise*, *x pixel shift*, *y pixel shift*, *pier*, and *subset*. The values of each of these fields are dumped into a file called "datatotal".

ii. Pier Determination: Each individual value of the pier field is extracted and dumped into a file called "datapier".

iii. Parameter Declarations: The parameter values for the IRAF commands used by *ap_apphot* are set as in Table 2.1.

iv. File Dimensions and Individual Coordinates: If the current iteration is the first file in the imported list, then the x and y dimensions are extracted and the coordinate list is broken down into individual star-n.reg files, where n is the number of the star in the coordinate file.

v. File Deletion Due to Shift: If the *x pixel shift* or *y pixel shift* of the file is greater than 50% of the file's total dimension, then the file is deleted and the script is sent to the next iteration.

vi. Radius Determination: The coordinates of each individual star are evaluated to ensure that the star's location is in a usable portion of the frame. If the star has moved off the frame, been shifted off by the SIDAP *ap_align* script, or is located within 10 pixels of the edge of the frame, then the star is deemed to be unusable and the radius to be used for that individual star on that individual frame is set to 3*FWHM. If the star is determined to be usable, then the radius required to collect the user-defined percentage of its flux is calculated. The coordinates of this star are further output to a coordinate file called "onframe.reg". All radii to be used are output to a file called "radius.info".

vii. Annulus and dAnnulus Parameters: Using the "onframe.reg" coordinate file, the usable coordinates are fed into the IRAF *phot* command using an aperture of 3*FWHM. The number of sky pixels in the annulus are extracted from the resulting .mag file and averaged. If the average is less than 500, then the process is repeated with dAnnulus = dAnnulus + 1. If the average is greater than 500, then the process is repeated with dAnnulus = dAnnulus - 0.1. This ensures that each annulus is set so that an average of 500 pixels are used in the sky value determination.

viii. Individual Star Photometry: Each star is then input to the IRAF *phot* command using the individual star-n.reg file. The radius used in the photometry is extracted from the "radius.info" file. Once all stars on the frame have had their photometry performed, the individual .mag files are concatenated and renumbered.

ix. .lst File Creation: The necessary information is then extracted, the value INDEF is changed to the value of zmag set within the script, and the .lst files are separated into photometry method, aperture factor, filter, integration time, and pier side (if applicable).

x. Duplication Removal: Duplicate entries within the .lst file are removed. This has only been required once, but remains in the script in order to prevent the possibility of future occurrences.

xi. .info File Creation: .info files are then created to keep track of radii used and average frame FWHM.

xii. Date/Time Header Addition: The date and time that the files were processed using the *ap_apphot* script are recorded in the file header.

Chapter 3

Aperture Photometry Technique Comparisons

3.1 Brief Introduction to the Fields Used for Photometry Comparisons

The two fields being utilized for photometry technique comparisons are the sparse stellar field of CY Aquarii (CYAqr) (=BD+00.4900, $\alpha^{2000} = 22^h37^m47.853^s$, $\delta^{2000} = +01°32'03.82''$) and the standard field M67 ($\alpha^{2000} = 08^h51^m18.0^s$, $\delta^{2000} = +11°59'18''$). The analysis of each field provides distinctly different views of the strengths and weaknesses of each method and which technique returns statistically superior results from the data.

The sparse stellar field of CYAqr was ideal for analyzing each technique for two reasons:

1. Each comparison star is isolated and therefore has no external light contamination, and
2. Each comparison star has been confirmed as non-variable, and therefore unexpected deviations should be minimal.

These characteristics make it optimal for study of S/N and magnitude standard deviations resulting from the use of each photometry method, since isolated stars minimize spurious signal and non-variables obviously have very small deviations in magnitude. Figure 3.1 shows CYAqr, along with its two comparison stars (one unnamed star (labeled 2) and TYC 567-2036-1 (labeled 3)). Comparison star 2 was the primary object of study.

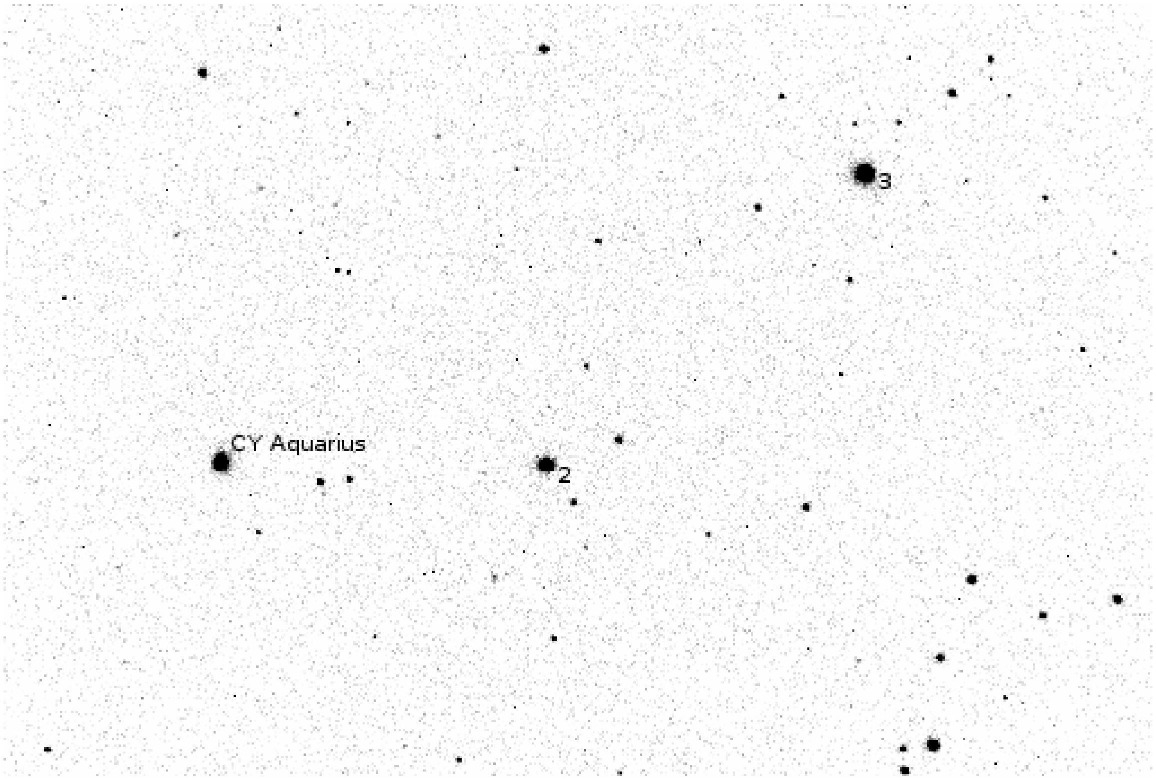

Figure 3.1: The CY Aqr star field with the two comparison stars used for technique analysis.

The standard field of M67 was used also for two reasons:

1. The field is a standard field and its stars have well-established magnitudes and colors, and
2. The range of magnitudes is relatively large.

These features make M67 ideal for determining which method provides the most statistically accurate magnitudes over a large range of instrumental magnitudes. The stars used in M67 for the magnitude study are shown in Figure 3.2. The corresponding F or region numbers are displayed in Table 3.1.

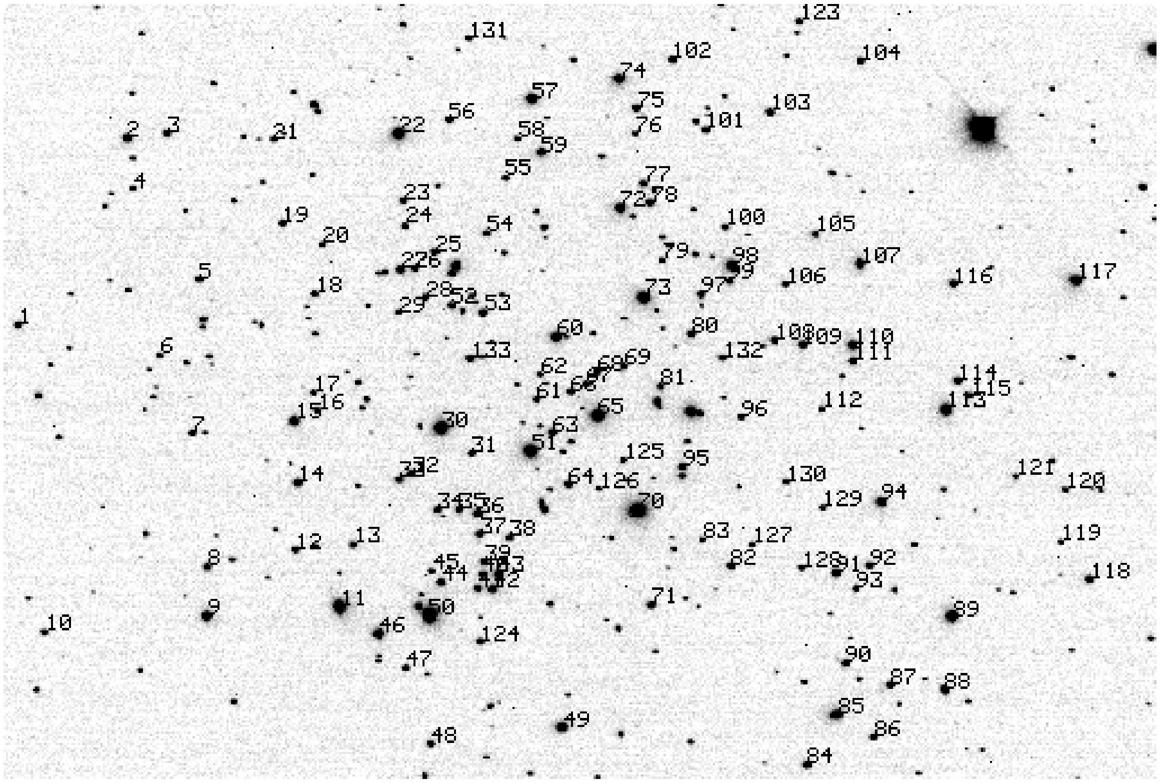

Figure 3.2: The M67 star field with stars used for magnitude determination comparisons.

Table 3.1. Correlation Between User-marked Star Number in M67 Image and Published F or Region Number (WEBDA 2007, Jeffrey 2003)

Image #	F or Region #	Image #	F or Region #	Image #	F or Region #	Image #	F or Region #
1	F20	41	F128	81	F174	121	F255
2	F30	42	F134	82	F195	122	F267
3	F37	43	20000	83	2012	123	F199
4	F33	44	F111	84	F216	124	F132
5	F48	45	F107	85	F224	125	2001
6	F42	46	F90	86	F233	126	5820
7	F52	47	F100	87	2054	127	F201
8	F54	48	F114	88	F245	128	F213
9	F55	49	F153	89	F244	129	2014
10	F27	50	F108	90	F225	130	F209
11	F81	51	F141	91	F221	131	F99
12	F75	52	F103	92	F227	132	F189
13	F83	53	F119	93	F226	133	F115
14	F72	54	4018	94	F231	...	...
15	F70	55	F123	95	F181	...	...
16	F77	56	F98	96	F194	...	...
17	F76	57	F131	97	F184	...	...
18	F71	58	F125	98	F190	...	...
19	F63	59	F136	99	F188	...	...
20	F73	60	F143	100	F187	...	...
21	F59	61	4004	101	F177	...	...
22	F84	62	4006	102	F165	...	...
23	F86	63	F145	103	F193	...	...
24	F87	64	F149	104	F215	...	...
25	F96	65	F155	105	3035	...	...
26	F89	66	F147	106	F205	...	...
27	F88	67	F150	107	F217	...	...
28	F103	68	F152	108	F202	...	...
29	4016	69	F161	109	F210	...	...
30	F105	70	F170	110	F218	...	...
31	F120	71	F176	111	F219	...	...
32	F95	72	F156	112	F214	...	...
33	F91	73	F164	113	F238	...	...
34	F106	74	F151	114	F241	...	...
35	F117	75	F157	115	F243	...	...
36	F124	76	3036	116	F236	...	...
37	F127	77	F162	117	F261	...	...
38	F140	78	F163	118	F265	...	...
39	F130	79	F169	119	2048	...	...
40	F135	80	F180	120	F263	...	...

3.2 Data Source and Method of Processing

Both data sets were collected at West Mountain Observatory (WMO) using the 0.31 m RCOS telescope fitted with an SBIG ST-10 XME CCD. Data for CY Aqr and M67 were obtained on 25 September 2006 and 4 November 2007, respectively. All data were processed utilizing SIDAP. Admittedly, the use of SIDAP deviates from the traditional method used at BYU. However, it is only necessary to utilize the same method in all reductions in order to meaningfully evaluate differing photometry techniques.

3.3 Results of Application of Different Photometry Methods

The results of the application of each photometry technique are presented in order to demonstrate the comparative strength of the current version of *ap_apphot* to the traditional technique used at BYU, as well as the previous versions of *ap_apphot* itself. Admittedly, determining which method is superior is a multi-faceted problem. However, the areas of comparison presented here provide a good general view of the capabilities and limitations of each technique.

3.3.1 Standard Deviation and Variance Results

The standard deviation and variance results for comparison star 2 in the CYAqr star field are presented in Table 3.2. These were attained utilizing differential analysis techniques. They represent how accurately the same method can replicate the same magnitude for our constant comparison star. In order to better visualize how each technique performs in relation to another, these same results for standard deviation and variance are presented graphically in Figures 3.3 and 3.4, respectively.

Table 3.2. Standard Deviation and Variance Results for Comparison Star 2 in the Field of CY Aqr

Photometry Technique	Standard Deviation (σ)	Variance (σ^2)
AP 6	0.00277	0.000007692
AP 7	0.00274	0.000007499
AP 8	0.00281	0.000007873
AP 9	0.00287	0.000008251
AP 10	0.00296	0.000008753
AP 11	0.00303	0.000009154
AP 12	0.00315	0.000009917
AP 13	0.00328	0.000010726
FWHM 1.0	0.00523	0.000027314
FWHM 1.5	0.00289	0.000008326
FWHM 2.0	0.00273	0.000007443
FWHM 2.5	0.00283	0.000008012
FWHM 3.0	0.00296	0.000008767
OLD AP 13	0.00364	0.000013261
FLUX 80%	0.00406	0.000016507
FLUX 85%	0.00401	0.000016052
FLUX 90%	0.00344	0.000011803
FLUX 95%	0.00331	0.000010977

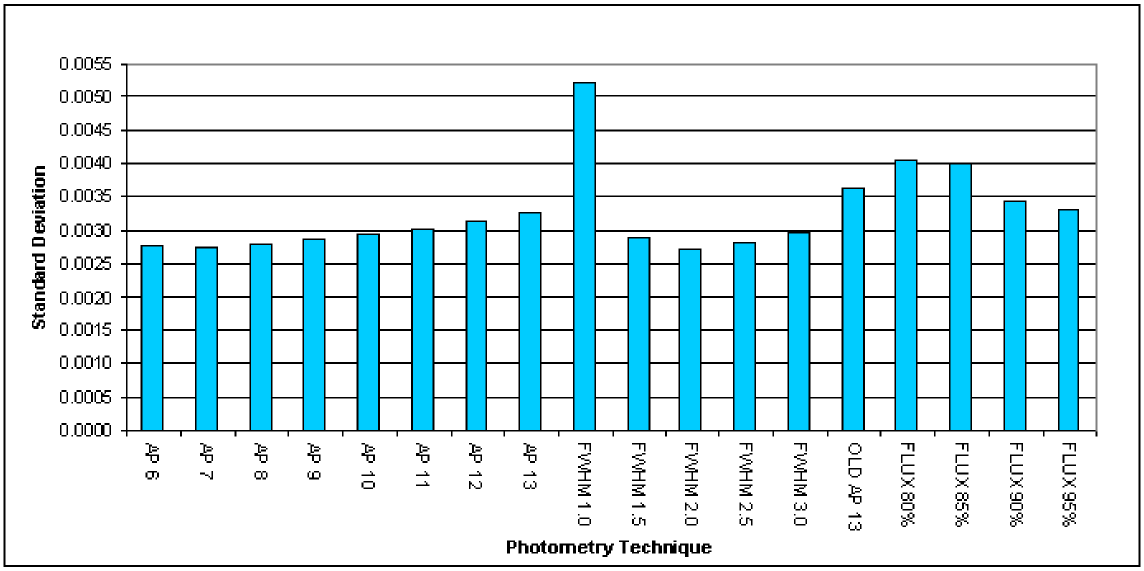

Figure 3.3: Standard deviations of Comparison Star 2 in the Field of CY Aqr - WMO 0.31-m, 25 September 2006.

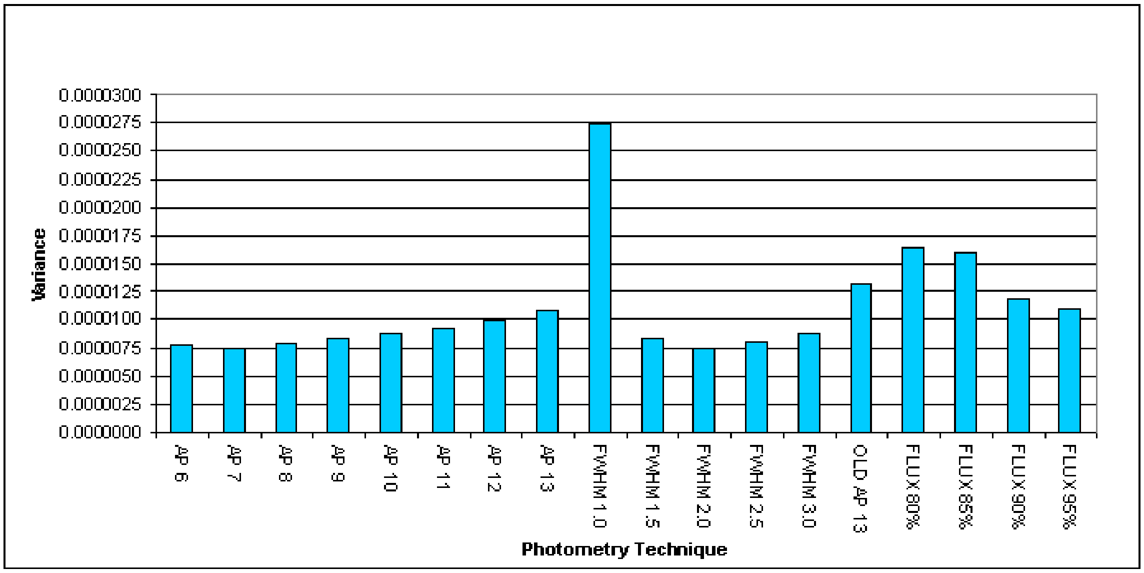

Figure 3.4: Variance of Comparison Star 2 in the Field of CY Aqr - WMO 0.31-m, 25 September 2006.

Based on these results, it appears that all photometry methods, outside of FWHM 1.0, Flux 80%, and Flux 85%, improve upon the traditional aperture photometry technique used at BYU. These differences (in the millionths) appear initially to be statistically insignificant. However, no detailed statistical analysis has yet been made. At this point, the methods that most improve standard deviation and variance are set aperture 7, FWHM 2.0, and Flux 95%.

3.3.2 S/N Results

The S/N results for the traditional, the set aperture, FWHM, and flux techniques are presented graphically in Figures 3.5, 3.6, 3.7, and 3.8, respectively. The standard deviation of S/N is presented in Figure 3.9. These results were also attained using differential analysis techniques. They represent how well a method can optimize itself for S/N. The general upward trend in all S/N graphs is due to the effect

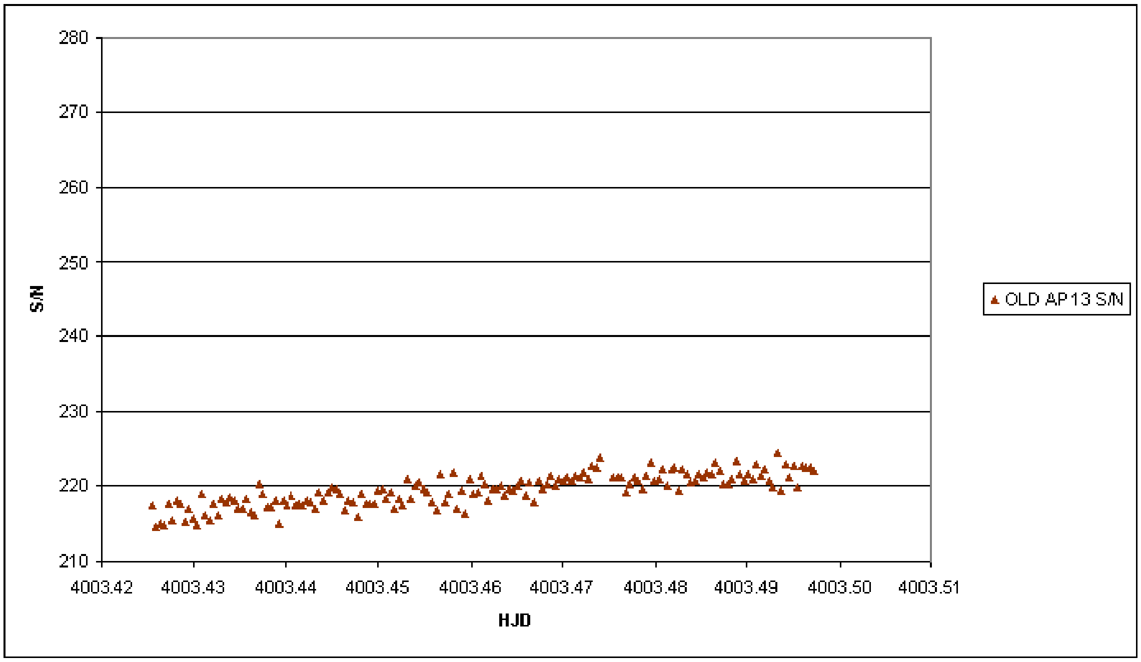

Figure 3.5: S/N of Comparison Star 2 in the Field of CY Aqr using traditional set aperture photometry technique.

of changing airmass. Understandably, this adds some error in the standard deviation calculations. However, even considering the error, the results can be compared reasonably.

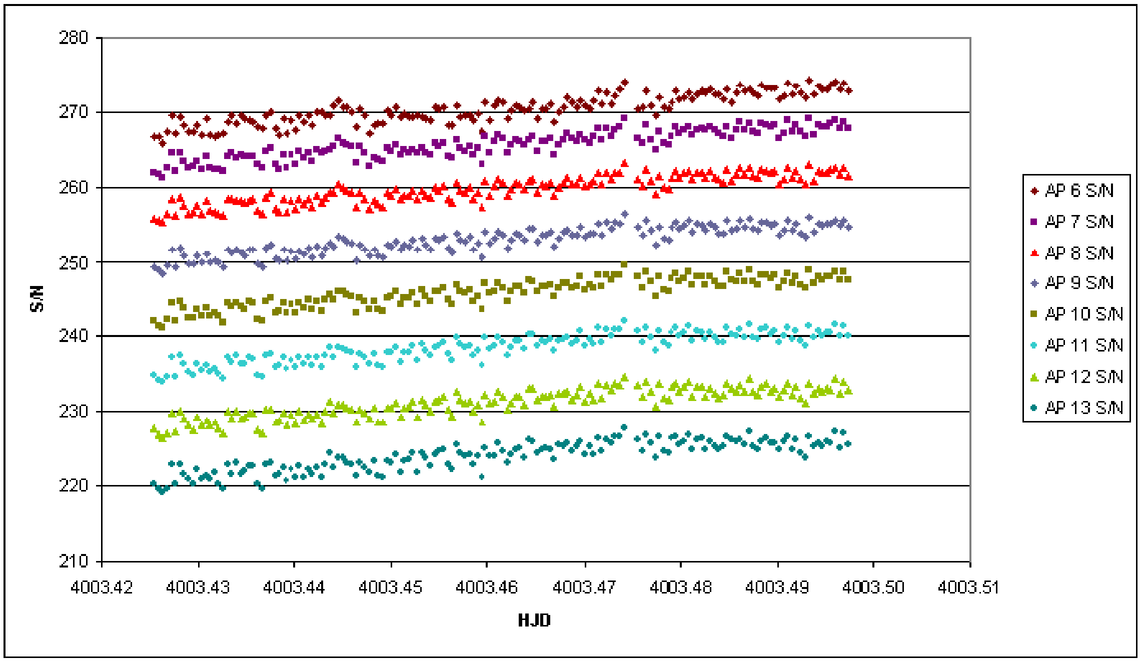

Figure 3.6: S/N of Comparison Star 2 in the Field of CY Aqr using set aperture photometry technique.

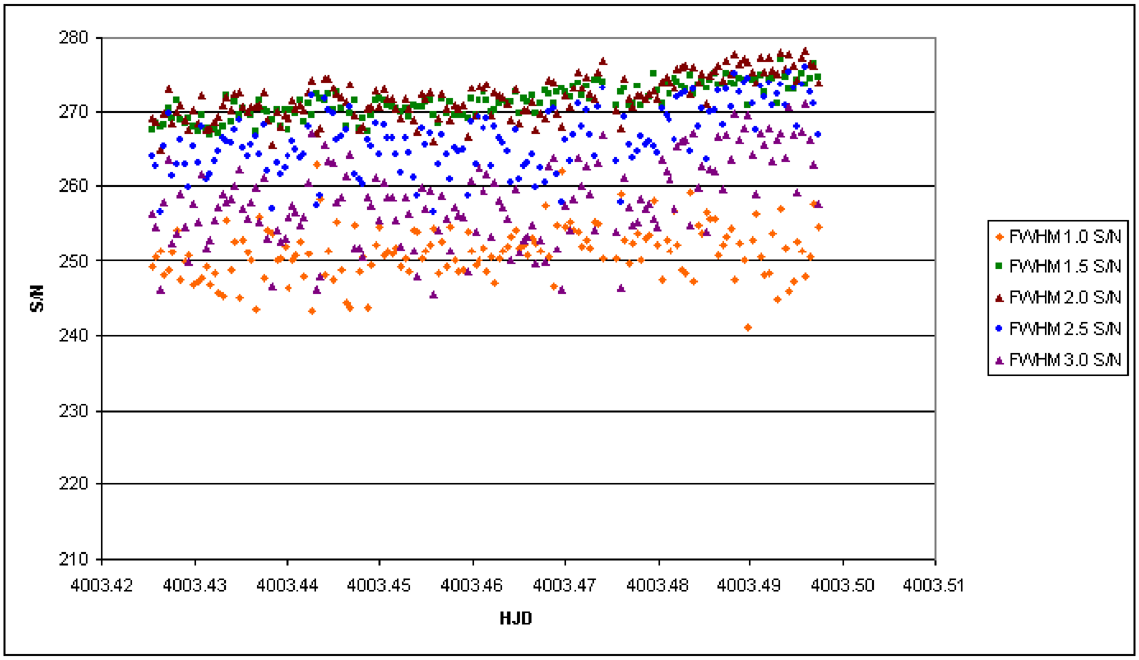

Figure 3.7: S/N of Comparison Star 2 in the Field of CY Aqr using FWHM aperture photometry technique.

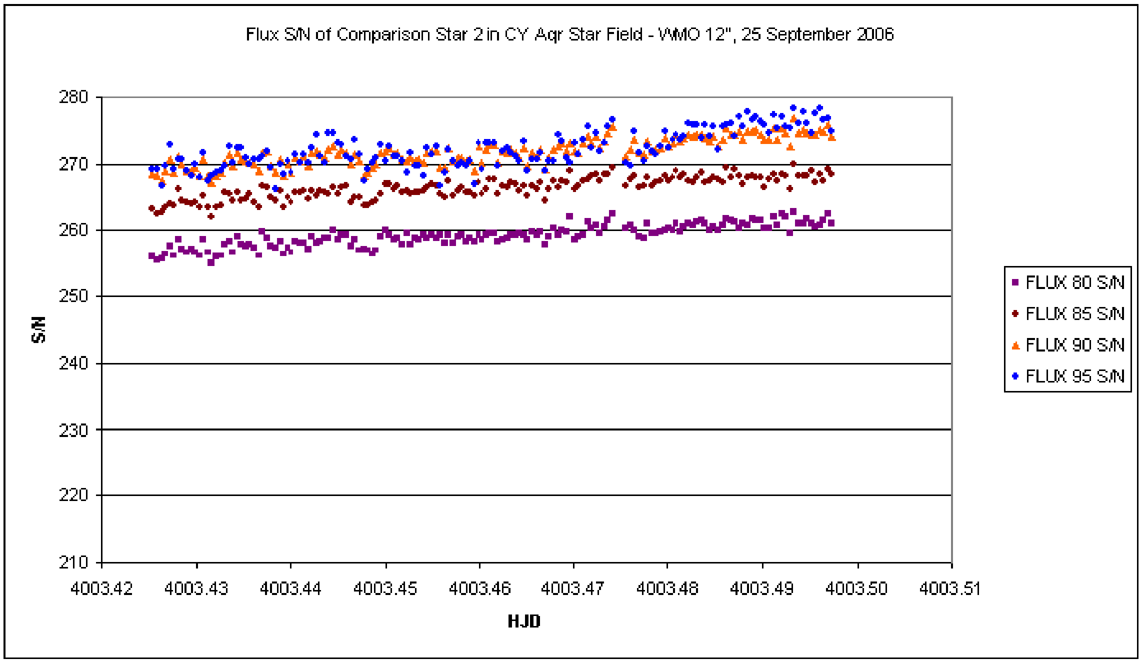

Figure 3.8: S/N of Comparison Star 2 in the Field of CY Aqr using flux aperture photometry technique.

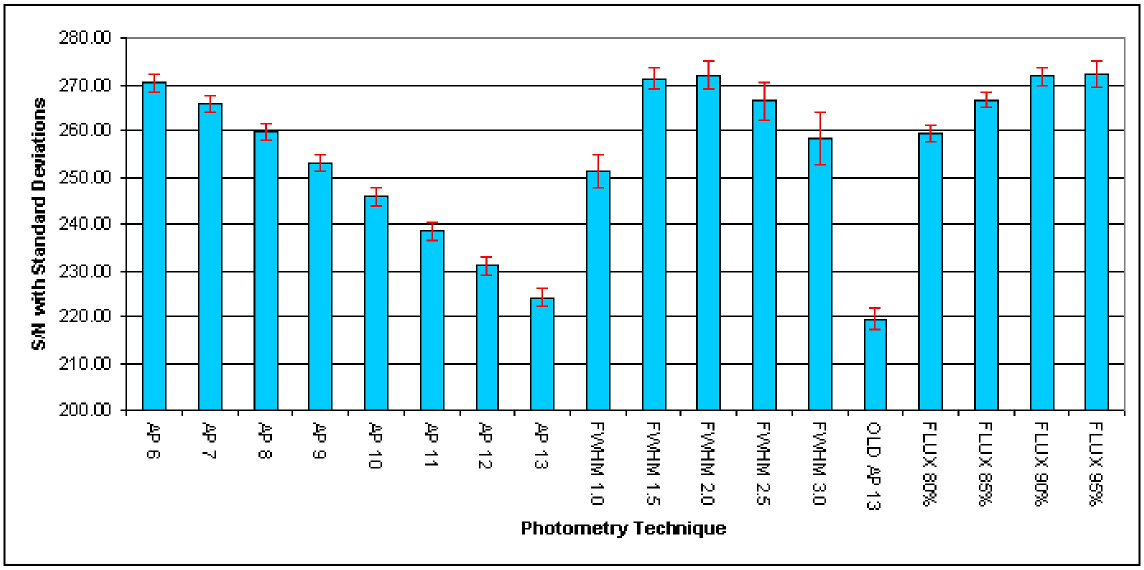

Figure 3.9: S/N and S/N standard deviations of Comparison Star 2 in the Field of CY Aqr.

It is clear from the graphs above that all techniques, except set apertures 12 and 13, improve S/N greatly in comparison to the traditional photometry technique. Furthermore, when standard deviations are considered, it is easy to see that AP 6, FWHM 1.5, FWHM 2.0, Flux 90%, and Flux 95% are superior to all other methods. The best techniques improved average S/N by almost 25%. Within each technique, the methods that most improved standard deviation and variance were set aperture 6, FWHM 2.0, and Flux 95%.

3.3.3 Calculating Statistically Accurate Magnitude Results

The resulting deviations of calculated magnitudes from true magnitudes in M67 are presented in Tables 3.3 and 3.4. Only the new set aperture and flux methods were analyzed. These were attained from a multi-variable least squares analysis performed by Dr. Benjamin Taylor. The traditional and FWHM methods were not analyzed, since we previously concluded that they were statistically inferior to the presented techniques. The residuals of the M67 analysis clearly demonstrated the existence of magnitude gradients in both X and Y directions. These gradients were confirmed with each technique. Although not direct measures of photometric precision, the t test values, indicating the significance of each gradient determination, provide meaningful comparison between various photometric methods.

Table 3.3. Least Squares Analysis Results for 30 and 60 Second Integration Times of M67 in the V Filter

Photometry Technique	Integration Time (s)	σ from Standard Mag (mmag)	X Gradient (mmag/1000 pixels)	T Value	Y Gradient (mmag/1000 pixels)	T Value
AP 6	30	37	-0.006	0.63	-0.003	0.27
...	60	38	-0.006	0.7	-0.003	0.31
...	60	42	-0.006	0.63	-0.012	0.98
AP 8	30	38	-0.01	1.11	-0.009	0.83
...	60	37	-0.01	1.09	-0.008	0.75
...	60	45	-0.009	0.84	-0.021	1.59
AP 10	30	39	-0.011	1.18	-0.013	1.1
...	60	38	-0.012	1.35	-0.011	0.96
...	60	49	-0.01	0.83	-0.026	1.78
AP 12	30	49	-0.007	1.97	-0.017	3.77
...	60	16	-0.009	2.32	-0.018	3.79
...	60	39	-0.005	0.6	-0.035	3.03
AP 14	30	15	-0.007	1.97	-0.017	3.77
...	60	16	-0.009	2.32	-0.018	3.79
...	60	39	-0.005	0.6	-0.035	3.03
Flux 80%	30	20	-0.008	1.66	-0.025	4.31
...	60	18	-0.009	2.22	-0.018	3.41
...	60	52	0	0.02	-0.048	3.11
Flux 85%	30	19	-0.009	1.89	-0.025	4.31
...	60	17	-0.009	2.08	-0.018	3.51
...	60	49	-0.001	0.09	-0.046	3.2
Flux 90%	30	19	-0.008	1.84	-0.024	4.35
...	60	17	-0.008	2.05	-0.017	3.35
...	60	46	-0.001	0.14	-0.044	3.26
Flux 95%	30	18	-0.009	2.03	-0.023	4.37
...	60	16	-0.008	2.06	-0.017	3.57
...	60	43	-0.001	0.14	-0.043	3.36

Table 3.4. Least Squares Analysis Results for 30 and 60 Second Integration Times of M67 in the R Filter

Photometry Technique	Integration Time (s)	σ from Standard Mag (mmag)	X Gradient (mmag/1000 pixels)	T Value	Y Gradient (mmag/1000 pixels)	T Value
AP 6	30	15	-0.003	0.89	-0.013	3
...	60	16	-0.005	1.33	-0.011	2.38
...	60	17	-0.01	2.51	-0.002	0.49
AP 8	30	14	-0.007	2.15	-0.017	4.26
...	60	15	-0.009	2.64	-0.016	3.62
...	60	16	-0.011	2.88	-0.009	1.93
AP 10	30	15	-0.008	2.2	-0.018	3.9
...	60	15	-0.011	2.97	-0.018	3.94
...	60	15	-0.011	2.97	-0.013	2.85
AP 12	30	16	-0.007	1.98	-0.017	3.64
...	60	16	-0.01	2.71	-0.019	4.04
...	60	15	-0.12	3.28	-0.015	3.41
AP 14	30	17	-0.007	1.92	-0.017	3.39
...	60	17	-0.008	2.17	-0.021	4.34
...	60	16	-0.015	4.01	-0.015	3.31
Flux 80%	30	16	-0.007	1.92	-0.022	4.56
...	60	37	-0.023	2.63	-0.029	2.68
...	60	36	-0.025	2.95	-0.036	3.46
Flux 85%	30	16	-0.007	1.97	-0.021	4.43
...	60	46	-0.025	2.3	-0.028	2.06
...	60	36	-0.025	2.91	-0.036	3.29
Flux 90%	30	16	-0.007	1.9	-0.021	4.48
...	60	71	-0.035	2.12	-0.036	1.7
...	60	37	-0.025	2.88	-0.036	3.33
Flux 95%	30	15	-0.007	1.91	-0.021	4.68
...	60	95	-0.044	1.98	-0.043	1.53
...	60	36	-0.025	2.97	-0.036	3.31

The V filter data were the first to be studied, and they demonstrate that the flux technique is substantially superior when determining statistically accurate magnitudes. Only the largest set apertures perform comparably to the flux technique, and it has been shown previously that these apertures have decidedly poorer standard deviation, variance, and S/N. This observation is further confirmed when the much higher t test values of the flux technique are considered. The poor results in all techniques in the second group of 60 second exposures have not been fully explored, but, looking retrospectively from the R results, we believe that they are indicative of right ascension (RA) drift in one of the two contributing frames.

In contrast to the V filter, the R filter data gave distinctly different results. Interestingly, all apertures of the set aperture method performed in a statistically indistinguishable manner. Furthermore, the flux technique appeared to be comparable in the 30 second results but inferior in the 60 second results. Subsequent exploration of the pre-analysis data revealed several frames with significant RA drift (i.e., stars were large ovals rather than circles). The possibility that these frames caused the poor results was confirmed when looking at the .info files generated by *ap_apphot*, which showed uncharacteristically large apertures being used for certain stars on those frames. Obviously, the RA drift resulted in very inaccurate calculations of the PSF used by the flux technique. These errors skewed all the flux technique results, but did not interfere with the set aperture technique. At this time the analysis has not been repeated with the poor frames removed. However, it is assumed that the flux method would show substantial gains over the set aperture technique similar to the V filter results.

Chapter 4

Conclusions and Suggestions

4.1 Technique Analysis

When comparing all aperture photometry techniques included in this analysis, it appears that the *ap_apphot* flux method (specifically the 95%) most closely approaches the abilities of an ideal aperture photometry technique. Although not necessarily the best method in all tests, it consistently performed well in standard deviation, variance, S/N, and especially in accuracy when calculating true magnitudes. This result does not mean that *ap_apphot* should be used exclusively for aperture photometry. Along with its strengths, this analysis also highlighted a large constraint to its use. *ap_apphot* relies heavily on an accurate determination of a point source's PSF. This was no more clearly observed than in the M67 results. Therefore, although *ap_apphot* is a fairly robust program that can obtain reliable results from "ugly" data, it is incapable of handling "very ugly" data (i.e., star images cannot be highly elongated). Nevertheless, it is clear from this analysis that, under normal conditions, *ap_apphot* will produce results superior to all other methods considered here.

4.2 Current Plans

Current plans exist to further improve *ap_apphot* over the coming summer of 2008. These plans include rewriting the script to make it more accessible to undergraduate students and others desiring to utilize it or improve upon it. While rewriting, the following changes are expected:

- Merge the individual Flux phot and annulus calculations,
- Remove the individual FWHM method,
- Improve .info file contents so as to make them more user friendly,

- Add more prompts to set parameters (i.e., zmag),
- Add extensive in-code documentation, and
- Modify algorithms to allow for piecemeal photometry.

Further analysis of the abilities of *ap_apphot* are also planned in order to determine the optimal flux percentage for use with the technique. Currently, plans include the testing of Flux 95-98%.

Although these changes will be implemented as soon as possible, the underlying principles that *ap_apphot* is built on will remain the same as those discussed in this thesis. Hopefully, future students will be able to utilize this in their work and further increase the quality of research being performed at BYU.

Appendix A

Appendix

A.1 *ap_apphot* Script

The *ap_apphot* script is presented here.

```
#DOCUMENTATION:ap_apphot_v_4.cl
#    ap_apphot_v_4.clbyPaulIverson

procedure ap_apphot_v_4(search_align_phot,coordinate_file,phot_method,pier_side_method,flux_per,fwhm_mult)

string search_align_phot {prompt="Enter search parameters for the aligned frames to phot (i.e. a-*star*B)"}
string coordinate_file   {prompt="Enter the name of the coordinatefile (i.e. ds9.reg)"}
string phot_method       {prompt="Enter the phot method to be used",enum="Flux|FWHM"}
string flux_per          {prompt="Enter the percentage of flux to use for each star",enum="80|85|90|95|98"}
string fwhm_mult         {prompt="Enter the multiple of fwhm to use for each star",enum="1.0|1.5|2.0|2.5|3.0"}
string pier_side_method  {prompt="Phot frames by pier side",enum="Yes|No"}

begin
     string searchalignphotometry
     string coordinate
     string phot_meth
     string flux_percent
     string fwhm_multiple
     string pier_side
end

#variable declarations
int         i=1
int         n=1
int         m=1
int         l=1
int         xlen,ylen
int         cont=0
real        skypix
real        xloc,yloc
real        dann=1
real        xshift,yshift
real        xstar,ystar
string      s1=""
string      s2=""
string      s3=""
string      s4=""
string      s5=""
string      s6=""
string      s7=""
string      s8=""
string      s9=""
string      s10=""
string      files
string      filter
string      checkfilter=""
string      inttime=""
string      checkinttime=""
string      lstfiles=""
string      star=""
string      checkstar=""
string      magnitude=""
string      HJD=""
string      airmass=""
string      pier=""
string      checkpier=""
string      datestring=""
string      timestring=""
string      X=""
string      Y=""
string      temp
struct      *list1
struct      *list2
struct      *list3

#deletes possibly incompatible datalists
delete ("data*")
delete ("onframe*")
delete ("nomo*")
delete ("radius.*")
delete ("fwhm.*")
delete ("*temp*")
delete ("star-*")

searchalignphotometry=search_align_phot
coordinate=coordinate_file
phot_meth=phot_method
```

```
if (phot_meth=="Flux"){
     flux_percent=flux_per
}
;
if (phot_meth=="FWHM"){
     fwhm_multiple=fwhm_mult
}
;

pier_side=pier_side_method

sections ((searchalignphotometry)//"*.fits",opt="fullname",>>"datalistfile")
if (sections.nimages>0){
     hselect
((searchalignphotometry)//"*.fits","$I,FWHM,GAIN,SKYDEV,HJD,RDNOISE,X_PIXEL_,Y_PIXEL_,PIER,SUBSET",yes,>>"datat
otal")
     tsort ("datatotal",5)
}
;

hselect ((searchalignphotometry)//"*.fits","PIER",yes,>>"datapier")
tsort ("datapier",1)
list2="datapier"
while (fscan(list2,s1)!=EOF){
     pier=(s1)
     if (pier!=checkpier){
            print (pier,>>"datalistpier")
     }
     ;
     checkpier=pier
}

#datalist contains files, fwhmpsf, epadu, sigma, HJD, rdnoise, pixel shifts, pier, subset
sections ("@datatotal")
if (sections.nimages>0){
     list1="datatotal"
     while (fscan(list1,s1,s2,s3,s4,s5,s6,s7,s8,s9,s10)!=EOF){
            clear
            print ("File:",s1)
            print ("HJD:",s5)
            print ("")
            print ("Subset:",s10)
            print ("")
            print ("Pier:",s9)
            print ("")
            print ("FWHM:",s2)
            print ("")
            print ("Gain:",s3)
            print ("RDNoise:",s6)
            print ("Sky Deviation:",s4)
            print ("")
            print ("X Pixel Shift:",s7)
            print ("Y Pixel Shift:",s8)

            sleep (1)

            centerpars.calg="centroid"
            centerpars.cbox=3*real(s2)
            #cthresholdsettingfromHintz'snewton
            centerpars.cthreshold=0.75
            centerpars.cmaxiter=20
            centerpars.maxshift=8.5

            fitskypars.salgorithm="mode"
            fitskypars.annulus=10*real(s2)
            fitskypars.dannulu=dann
            #sloclipandshiclipsettingfromHintz'snewton
            fitskypars.sloclip=0.
            fitskypars.shiclip=10.

            photpars.weighting="constant"
            photpars.apertur=3*real(s2)
            photpars.zmag=30

            datapars.fwhmpsf=real(s2)
            datapars.sigma=real(s4)
datapars.datamin=INDEF
datapars.datamax=INDEF
datapars.ccdread="RDNOISE"
datapars.gain="GAIN"
datapars.readnoise=real(s6)
datapars.epadu=real(s3)
datapars.exposure="EXPTIME"
datapars.airmass="AIRMASS"
datapars.filter="SUBSET"
datapars.obstime="HJD"

psfmeasure.radius=5
psfmeasure.sbuffer=5
psfmeasure.swidth=5

files=(s1)
xshift=real(s7)
yshift=real(s8)

pier=(s9)

if (m==1){
       imgets (files,param="i_naxis1")
       xlen=int (imgets.value)
       imgets (files,param="i_naxis2")
       ylen=int (imgets.value)


       list2=(coordinate)
       while (fscan(list2,X,Y)!=EOF){
              print (X,>"datalistx")
              print (Y,>"datalisty")
              joinlines ("datalistx,datalisty",>"star-"//n//".reg")

              delete ("datalistx")
              delete ("datalisty")

              n=n+1
       }
       n=1

       m=2
}
;

if (xshift<0 && xshift<0-0.5*xlen){
       delete (files)
       next
}
;
if (xshift>0 && xshift>0.5*xlen){
       delete (files)
       next
}
;
if (yshift<0 && yshift<0-0.5*ylen){
       delete (files)
       next
}
;
if (yshift>0 && yshift>0.5*ylen){
       delete (files)
       next
}
;

if (phot_meth=="Flux"){
       print (3*real(s2),>>"temp_frame_fwhm_aperture_flux"//flux_percent//".info")
       print (real(s2),>>"temp_frame_fwhm_flux"//flux_percent//".info")
       print ((s10),>>"temp_frame_subset_flux"//flux_percent//".info")
       print ((s5),>>"temp_HJD.info")
}
;
if (phot_meth=="FWHM"){
       print (3*real(s2),>>"temp_frame_fwhm_aperture_FWHM"//fwhm_multiple//".info")
       print (real(s2),>>"temp_frame_fwhm_FWHM"//fwhm_multiple//".info")
       print ((s10),>>"temp_frame_subset_FWHM"//fwhm_multiple//".info")
```

```
                print ((s5),>>"temp_HJD.info")
        }
        ;

        sections ("@"//(coordinate))
        for (n=1;n<=sections.nimages;n+=1){
                tabpar ((coordinate),1,n)
                xstar=real(tabpar.value)
                tabpar ((coordinate),2,n)
                ystar=real(tabpar.value)

                if (phot_meth=="Flux"){
                        if (xshift<=0&&xstar>xlen+xshift-10){
                                print (3*real(s2),>"star_radi-"//n)
                                cont=1
                                print (3*real(s2),>>"temp_radii_star-"//n//"_flux"//flux_percent//".info")
                                print ((s5),>>"temp_star-"//n//"_HJD.info")
                        }
                        ;
                        if (xshift>=0&&xstar<0+xshift+10&&cont!=1){
                                print (3*real(s2),>"star_radi-"//n)
                                cont=1
                                print (3*real(s2),>>"temp_radii_star-"//n//"_flux"//flux_percent//".info")
                                print ((s5),>>"temp_star-"//n//"_HJD.info")
                        }
                        ;
                        if (xstar>xlen-10&&cont!=1){
                                print (3*real(s2),>"star_radi-"//n)
                                cont=1
                                print (3*real(s2),>>"temp_radii_star-"//n//"_flux"//flux_percent//".info")
                                print ((s5),>>"temp_star-"//n//"_HJD.info")
                        }
                        ;
                        if (xstar<10&&cont!=1){
                                print (3*real(s2),>"star_radi-"//n)
                                cont=1
                                print (3*real(s2),>>"temp_radii_star-"//n//"_flux"//flux_percent//".info")
                                print ((s5),>>"temp_star-"//n//"_HJD.info")
                        }
                        ;
                        if (yshift<=0&&ystar>ylen+yshift-10&&cont!=1){
                                print (3*real(s2),>"star_radi-"//n)
                                cont=1
                                print (3*real(s2),>>"temp_radii_star-"//n//"_flux"//flux_percent//".info")
                                print ((s5),>>"temp_star-"//n//"_HJD.info")
                        }
                        ;
                        if (yshift>=0&&ystar<0+yshift+10&&cont!=1){
                                print (3*real(s2),>"star_radi-"//n)
                                cont=1
                                print (3*real(s2),>>"temp_radii_star-"//n//"_flux"//flux_percent//".info")
                                print ((s5),>>"temp_star-"//n//"_HJD.info")
                        }
                        ;
                        if (ystar>ylen-10&&cont!=1){
                                print (3*real(s2),>"star_radi-"//n)
                                cont=1
                                print (3*real(s2),>>"temp_radii_star-"//n//"_flux"//flux_percent//".info")
                                print ((s5),>>"temp_star-"//n//"_HJD.info")
                        }
                        ;
                        if (ystar<10&&cont!=1){
                                print (3*real(s2),>"star_radi-"//n)
                                cont=1
                                print (3*real(s2),>>"temp_radii_star-"//n//"_flux"//flux_percent//".info")
                                print ((s5),>>"temp_star-"//n//"_HJD.info")
                        }
                        ;
                        if (cont==0){
                                print ("q",>>"nomoreqs")

                                clear
                                print ("Current file",files)
                                print ("Calculating PSF Flux of star: star-",n,".reg")

                                psfmeasure (files,disp-,level=int (flux_percent),size="Radius",imagecur="star-
"//n//".reg",graphcur="nomoreqs",>"dataradA")
```

```
                                sed -e '/NOAO/d' -e '/Image/d' -e '/Average/d' -e '/flux/d' -e 's/" "/d/g' -e
's/obj-.........../ddddddddddddddd/g' "dataradA" >> "dataradB"

                                list2="dataradB"
                                while (fscan(list2,s3)!=EOF){
                                       temp=(s3)
                                       i=strlen(temp)
                                       if (l!=1&&l<=2){
                                              temp=substr (temp,41,i-15)
                                              print (temp,>>"dataradC")
                                       }
                                       ;
                                       l=l+1
                                }
                                l=1

                                sed -e 's/d//g' "dataradC" >> "dataradD"
                                rename ("dataradD","star_radi-"//n)

                                print (xstar,>>"onframexstar")
                                print (ystar,>>"onframeystar")

                                tabpar ("star_radi-"//n,1,1)
                                temp=tabpar.value
                                print (temp,>>"temp_radii_star-"//n//"_flux"//flux_percent//".info")
                                print ((s5),>>"temp_star-"//n//"_HJD.info")
                         }
                         ;

                         if (n==1){
                                rename ("star_radi-1","radius.info")
                         }
                         ;
                         if (n!=1){
                                concat ("radius.info"//","//"star_radi-"//n,>"tempradii")
                                delete ("radius.info")
                                rename ("tempradii","radius.info")
                         }
                         ;

                         delete ("datarad*")
                         cont=0
                  }
                  ;

                  if (phot_meth=="FWHM"){
                         if (xshift<=0&&xstar>xlen+xshift-10){
                                print (3*real(s2),>"star_fwhm-"//n)
                                cont=1
                                print (3*real(s2),>>"temp_fwhm_star-"//n//"_FWHM"//fwhm_multiple//".info")
                                print ((s5),>>"temp_star-"//n//"_HJD.info")
                         }
                         ;
                         if (xshift>=0&&xstar<0+xshift+10&&cont!=1){
                                print (3*real(s2),>"star_fwhm-"//n)
                                cont=1
                                print (3*real(s2),>>"temp_fwhm_star-"//n//"_FWHM"//fwhm_multiple//".info")
                                print ((s5),>>"temp_star-"//n//"_HJD.info")
                         }
                         ;
                         if (xstar>xlen-10&&cont!=1){
                                print (3*real(s2),>"star_fwhm-"//n)
                                cont=1
                                print (3*real(s2),>>"temp_fwhm_star-"//n//"_FWHM"//fwhm_multiple//".info")
                                print ((s5),>>"temp_star-"//n//"_HJD.info")
                         }
                         ;
                         if (xstar<10&&cont!=1){
                                print (3*real(s2),>"star_fwhm-"//n)
                                cont=1
                                print (3*real(s2),>>"temp_fwhm_star-"//n//"_FWHM"//fwhm_multiple//".info")
                                print ((s5),>>"temp_star-"//n//"_HJD.info")
                         }
                         ;
                         if (yshift<=0&&ystar>ylen+yshift-10&&cont!=1){
                                print (3*real(s2),>"star_fwhm-"//n)
                                cont=1
                                print (3*real(s2),>>"temp_fwhm_star-"//n//"_FWHM"//fwhm_multiple//".info")
                                print ((s5),>>"temp_star-"//n//"_HJD.info")
                        }
                        ;
                        if (yshift>=0&&ystar<0+yshift+10&&cont!=1){
                                print (3*real(s2),>"star_fwhm-"//n)
                                cont=1
                                print (3*real(s2),>>"temp_fwhm_star-"//n//"_FWHM"//fwhm_multiple//".info")
                                print ((s5),>>"temp_star-"//n//"_HJD.info")
                        }
                        ;
                        if (ystar>ylen-10&&cont!=1){
                                print (3*real(s2),>"star_fwhm-"//n)
                                cont=1
                                print (3*real(s2),>>"temp_fwhm_star-"//n//"_FWHM"//fwhm_multiple//".info")
                                print ((s5),>>"temp_star-"//n//"_HJD.info")
                        }
                        ;
                        if (ystar<10&&cont!=1){
                                print (3*real(s2),>"star_fwhm-"//n)
                                cont=1
                                print (3*real(s2),>>"temp_fwhm_star-"//n//"_FWHM"//fwhm_multiple//".info")
                                print ((s5),>>"temp_star-"//n//"_HJD.info")
                        }
                        ;
                        if (cont==0){
                                print ("q",>>"nomoreqs")

                                clear
                                print ("Current file",files)
                                print ("Calculating PSF FWHM of star: star-",n,".reg")

                                psfmeasure (files,disp-,level=0.5,size="FWHM",imagecur="star-
"//n//".reg",graphcur="nomoreqs",>"datafwhmA")

                                sed -e '/NOAO/d' -e '/Image/d' -e '/Average/d' -e 's/" "/d/g' -e 's/obj-
.........../dddddddddddddddd/g' "datafwhmA" >> "datafwhmB"

                                list2="datafwhmB"
                                while (fscan(list2,s3)!=EOF){
                                        temp=(s3)
                                        i=strlen(temp)
                                        if (l!=1&&l<=2){
                                                temp=substr (temp,41,i-15)
                                                print (temp,>>"datafwhmC")
                                        }
                                        ;
                                        l=l+1
                                }
                                l=1

                                sed -e 's/d//g' "datafwhmC" >> "datafwhmD"
                                rename ("datafwhmD","star_fwhm-"//n)

                                print (xstar,>>"onframexstar")
                                print (ystar,>>"onframeystar")

                                tabpar ("star_fwhm-"//n,1,1)
                                temp=tabpar.value
                                print (temp,>>"temp_fwhm_star-"//n//"_FWHM"//fwhm_multiple//".info")
                                print ((s5),>>"temp_star-"//n//"_HJD.info")
                        }
                        ;

                        if (n==1){
                                rename ("star_fwhm-1","fwhm.info")
                        }
                        ;
                        if (n!=1){
                                concat ("fwhm.info"//","//"star_fwhm-"//n,>"tempfwhm")
                                delete ("fwhm.info")
                                rename ("tempfwhm","fwhm.info")
                        }
                        ;

                        delete ("datafwhm*")
```

```
              cont=0
       }
       ;
}
n=1

joinlines ("onframexstar,onframeystar",>"onframe.reg")

phot (files,coords="onframe.reg",interac-,verif-)

txdump (files//".mag.1","nsky",yes,>>"nsky.info")
type "nsky.info" | average > "sky.info"

tabpar ("sky.info",1,1)
skypix=real(tabpar.value)

delete ("*sky*")

while (skypix<500){
       dann=dann+1
       fitskypars.dannulu=dann

       clear
       print ("Determining proper annulus and dannulus"
       print ("")

       delete (files//".mag.1")
       phot (files,coords="onframe.reg",interac-,verif-)

       txdump (files//".mag.1","nsky",yes,>>"nsky.info")
       type "nsky.info" | average > "sky.info"

       tabpar ("sky.info",1,1)
       skypix=real(tabpar.value)

       delete ("*sky*")
}
while (skypix>505){
       dann=dann-0.1
       fitskypars.dannulu=dann

       clear
       print ("Determining proper annulus and dannulus"
       print ("")

       delete (files//".mag.1")
       phot (files,coords="onframe.reg",interac-,verif-)

       txdump (files//".mag.1","nsky",yes,>"nsky.info")
       type "nsky.info" | average > "sky.info"

       tabpar ("sky.info",1,1)
       skypix=real(tabpar.value)

       delete ("*sky*")
}
delete (files//".mag.1")

if (phot_meth=="Flux"){
       list2="radius.info"
}
;
if (phot_meth=="FWHM"){
       list2="fwhm.info"
}
;
while (fscan(list2,s1)!=EOF){
       photpars.apertur=real(s1)

       clear
       print ("Photting: star-"//n//".reg")
       phot (files,output=files//".mag."//n,coords="star-"//n//".reg",interac-,verif-)

       if (n==1){
              rename (files//".mag.1","magfile")
       }
       ;
```

```
                   if (n!=1){
                          pconcat ("magfile"//","//files//".mag."//n,"tempmag")
                          delete ("magfile")
                          rename ("tempmag","magfile")
                   }
                   ;
                   n=n+1
            }
            n=1

            delete (files//".mag*")

            prenumber ("magfile")

            if (phot_meth=="Flux"&&pier_side=="Yes"){
                   rename ("magfile",files//"_"//pier//"_flux"//flux_percent//".mag.1")
            }
            ;
            if (phot_meth=="Flux"&&pier_side=="No"){
                   rename ("magfile",files//"_NA_flux"//flux_percent//".mag.1")
            }
            ;
            if (phot_meth=="FWHM"&&pier_side=="Yes"){
                   rename ("magfile",files//"_"//pier//"_FWHM"//fwhm_multiple//".mag.1")
            }
            ;
            if (phot_meth=="FWHM"&&pier_side=="No"){
                   rename ("magfile",files//"_NA_FWHM"//fwhm_multiple//".mag.1")
            }
            ;

            delete ("onframe*star")
            delete ("star_*")

            if (phot_meth=="Flux"){
                   delete ("radius.info")
            }
            ;
            if (phot_meth=="FWHM"){
                   delete ("fwhm.info")
            }
            ;

            delete ("onframe.reg")

            dann=1
     }
}
;

if (pier_side=="Yes"){
     list1="datalistpier"
     while (fscan(list1,s1)!=EOF){
            pier=(s1)
            if (phot_meth=="Flux"){
                   sections
("*"//pier//"_flux"//flux_percent//".mag*",opt="fullname",>>"datalist_"//pier//"_magfiles")
            }
            ;
            if (phot_meth=="FWHM"){
                   sections
("*"//pier//"_FWHM"//fwhm_multiple//".mag*",opt="fullname",>>"datalist_"//pier//"_magfiles")
            }
            ;
            list2="datalist_"//pier//"_magfiles"
            while (fscan(list2,s2)!=EOF){
                   files=(s2)

                   txdump (files,"id,mag,otime,xairmass,ifilter,itime",yes,>>"star.lst")
            }

            #changes INDEFs to 30.0 since photpars.zmag = 30
            clear
            print ("Changing INDEFs to zmag = 30 in .lst files")
            sleep (1)

            sed -e 's/INDEF/30.0/g' "star.lst" > "startemp.lst"
            delete ("star.lst")
            rename ("startemp.lst","star.lst")

            fields ("star.lst",5,>"datalist_"//pier//"_filter")
            tsort ("datalist_"//pier//"_filter",1)
            list2="datalist_"//pier//"_filter"
            while (fscan(list2,s2)!=EOF){
                   filter=(s2)

                   if (checkfilter!=filter){
                          print (filter,>>"datalist_"//pier//"_filtertable")
                   }
                   ;
                   checkfilter=filter
            }
            checkfilter=""

            fields ("star.lst",6,>"datalist_"//pier//"_inttime")
            tsort ("datalist_"//pier//"_inttime",1)
            list2="datalist_"//pier//"_inttime"
            while (fscan(list2,s2)!=EOF){
                   inttime=(s2)

                   if (checkinttime!=inttime){
                          print (inttime,>>"datalist_"//pier//"_inttimetable")
                   }
                   ;
                   checkinttime=inttime
            }
            checkinttime=""

            list2="datalist_"//pier//"_filtertable"
            while (fscan(list2,s2)!=EOF){
                   filter=(s2)
                   list3="datalist_"//pier//"_inttimetable"
                   while (fscan(list3,s3)!=EOF){
                          inttime=(s3)

                          if (phot_meth=="Flux"){
                                 match (filter,"star.lst",>"startemp.lst")
                                 match (inttime,"startemp.lst",>"startemp2.lst")
                                 fields ("startemp2.lst","1-
5",>"star"//filter//"_"//inttime//"_"//pier//"_flux"//flux_percent//".lst")
                                 sections
("@star"//filter//"_"//inttime//"_"//pier//"_flux"//flux_percent//".lst")
                                 if (sections.nimages==0){
                                        delete
("star"//filter//"_"//inttime//"_"//pier//"_flux"//flux_percent//".lst")
                                 }
                                 ;
                          }
                          ;
                          if (phot_meth=="FWHM"){
                                 match (filter,"star.lst",>"startemp.lst")
                                 match (inttime,"startemp.lst",>"startemp2.lst")
                                 fields ("startemp2.lst","1-
5",>"star"//filter//"_"//inttime//"_"//pier//"_FWHM"//fwhm_multiple//".lst")
                                 sections
("@star"//filter//"_"//inttime//"_"//pier//"_FWHM"//fwhm_multiple//".lst")
                                 if (sections.nimages==0){
                                        delete
("star"//filter//"_"//inttime//"_"//pier//"_FWHM"//fwhm_multiple//".lst")
                                 }
                                 ;
                          }
                          ;
                          delete ("startemp.lst")
                          delete ("startemp2.lst")
                   }
            }

            delete ("star.lst")

            if (phot_meth=="Flux"){
                   tsort ("*_"//pier//"_flux"//flux_percent//".lst","3,1")
                   sections ("*_"//pier//"_flux"//flux_percent//".lst",>>"datalistlstfiles")
            }
            ;
            if (phot_meth=="FWHM"){
                   tsort ("*_"//pier//"_FWHM"//fwhm_multiple//".lst","3,1")
                   sections ("*_"//pier//"_FWHM"//fwhm_multiple//".lst",>>"datalistlstfiles")
            }
            ;
     }
}
;
if (pier_side=="No"){
     if (phot_meth=="Flux"){
            sections ("*NA_flux"//flux_percent//".mag*",opt="fullname",>>"datalistmagfiles")
     }
     ;
     if (phot_meth=="FWHM"){
            sections ("*NA_FWHM"//fwhm_multiple//".mag*",opt="fullname",>>"datalistmagfiles")
     }
     ;

     list1="datalistmagfiles"
     while (fscan(list1,s1)!=EOF){
            files=(s1)

            txdump (files,"id,mag,otime,xairmass,ifilter,itime",yes,>>"star.lst")
     }

     #changes INDEFs to 30.0 since photpars.zmag = 30
     clear
     print ("Changing INDEFs to zmag = 30 in .lst files")
     sleep (1)

     sed -e 's/INDEF/30.0/g' "star.lst" > "startemp.lst"
     delete ("star.lst")
     rename ("startemp.lst","star.lst")

     fields ("star.lst",5,>"datalistfilter")
     tsort ("datalistfilter",1)
     list1="datalistfilter"
     while (fscan(list1,s1)!=EOF){
            filter=(s1)

            if (checkfilter!=filter){
                   print (filter,>>"datalistfiltertable")
            }
            ;
            checkfilter=filter
     }

     fields ("star.lst",6,>"datalist_inttime")
     tsort ("datalist_inttime",1)
     list1="datalist_inttime"
     while (fscan(list1,s1)!=EOF){
            inttime=(s1)

            if (checkinttime!=inttime){
                   print (inttime,>>"datalist_inttimetable")
            }
            ;
            checkinttime=inttime
     }
     checkinttime=""

     list1="datalistfiltertable"
     while (fscan(list1,s1)!=EOF){
            filter=(s1)
            list2="datalist_inttimetable"
            while (fscan(list2,s2)!=EOF){
                   inttime=(s2)
                   if (phot_meth=="Flux"){
                          match (filter,"star.lst",>"startemp.lst")
                          match (inttime,"startemp.lst",>"startemp2.lst")
                          fields ("startemp2.lst","1-
5",>"star"//filter//"_"//inttime//"_NA_flux"//flux_percent//".lst")
                          sections ("@star"//filter//"_"//inttime//"_NA_flux"//flux_percent//".lst")
                          if (sections.nimages==0){
                                 delete ("star"//filter//"_"//inttime//"_NA_flux"//flux_percent//".lst")
                          }
                          ;
                   }
                   ;
                   if (phot_meth=="FWHM"){
                          match (filter,"star.lst",>"startemp.lst")
                          match (inttime,"startemp.lst",>"startemp2.lst")
                          fields ("startemp2.lst","1-
5",>"star"//filter//"_"//inttime//"_NA_FWHM"//fwhm_multiple//".lst")
                          sections ("@star"//filter//"_"//inttime//"_NA_FWHM"//fwhm_multiple//".lst")
                          if (sections.nimages==0){
                                 delete ("star"//filter//"_"//inttime//"_NA_FWHM"//fwhm_multiple//".lst")
                          }
                          ;
                   }
                   ;
                   delete ("startemp.lst")
                   delete ("startemp2.lst")
            }
     }

     delete ("star.lst")

     if (phot_meth=="Flux"){
            tsort ("*_NA_flux"//flux_percent//".lst","3,1")
            sections ("*_NA_flux"//flux_percent//".lst",>>"datalistlstfiles")
     }
     ;
     if (phot_meth=="FWHM"){
            tsort ("*_NA_FWHM"//fwhm_multiple//".lst","3,1")
            sections ("*_NA_FWHM"//fwhm_multiple//".lst",>>"datalistlstfiles")
     }
     ;
}
;

clear
print ("Removing possible duplicates in .lst files")
sleep (1)

list1="datalistlstfiles"
while (fscan(list1,s1)!=EOF){
     lstfiles=(s1)
     list2=lstfiles
     while (fscan(list2,s2,s3,s4,s5,s6)!=EOF){
            star=(s2)
            magnitude=(s3)
            HJD=(s4)
            airmass=(s5)
            filter=(s6)

            if (star!=checkstar){
                   print (star,>>"dataliststar")
                   print (magnitude,>>"datalistmag")
                   print (HJD,>>"datalistHJD")
                   print (airmass,>>"datalistairmass")
                   print (filter,>>"datalistfilter")
            }
            ;
            checkstar=star
     }
     joinlines ("dataliststar,datalistmag,datalistHJD,datalistairmass,datalistfilter",>lstfiles//".corr")

     delete ("dataliststar")
     delete ("datalistmag")
     delete ("datalistHJD")
     delete ("datalistairmass")
     delete ("datalistfilter")

     delete (lstfiles)
     rename (lstfiles//".corr",lstfiles)
}

clear
print ("Creating .info files")
print ("")
sleep (1)
```

```
sections ("@"//(coordinate))
for (n=1;n<=sections.nimages;n+=1){
     if (phot_meth=="Flux"&&pier_side=="Yes"){
            print ("Creating radii_star-"//n//"_EW_flux"//flux_percent//".info")
            joinlines ("temp_star-"//n//"_HJD.info,temp_radii_star-
"//n//"_flux"//flux_percent//".info",>"radii_star-"//n//"_EW_flux"//flux_percent//".info")
     }
     ;
     if (phot_meth=="FWHM"&&pier_side=="Yes"){
            print ("Creating fwhm_star-"//n//"_EW_FWHM"//fwhm_multiple//".info")
            joinlines ("temp_star-"//n//"_HJD.info,temp_fwhm_star-
"//n//"_FWHM"//fwhm_multiple//".info",>"fwhm_star-"//n//"_EW_FWHM"//fwhm_multiple//".info")
     }
     ;
     if (phot_meth=="Flux"&&pier_side=="No"){
            print ("Creating radii_star-"//n//"_NA_flux"//flux_percent//".info")
            joinlines ("temp_star-"//n//"_HJD.info,temp_radii_star-
"//n//"_flux"//flux_percent//".info",>"radii_star-"//n//"_NA_flux"//flux_percent//".info")
     }
     ;
     if (phot_meth=="FWHM"&&pier_side=="No"){
            print ("Creating fwhm_star-"//n//"_NA_FWHM"//fwhm_multiple//".info")
            joinlines ("temp_star-"//n//"_HJD.info,temp_fwhm_star-
"//n//"_FWHM"//fwhm_multiple//".info",>"fwhm_star-"//n//"_NA_FWHM"//fwhm_multiple//".info")
     }
     ;
}

if (phot_meth=="Flux"&&pier_side=="Yes"){
     print ("")
     print ("Creating avg frame fwhm aperture and frame fwhm .info file")
     joinlines
("temp_HJD.info,temp_frame_fwhm_aperture_flux"//flux_percent//".info,temp_frame_subset_flux"//flux_percent//".i
nfo",>"avg_frame_fwhm_aperture_EW_flux"//flux_percent//".info")
     joinlines
("temp_HJD.info,temp_frame_fwhm_flux"//flux_percent//".info,temp_frame_subset_flux"//flux_percent//".info",>"av
g_frame_fwhm_EW_flux"//flux_percent//".info")
}
;

if (phot_meth=="FWHM"&&pier_side=="Yes"){
     print ("")
     print ("Creating avg frame fwhm aperture and frame fwhm .info file")
     joinlines
("temp_HJD.info,temp_frame_fwhm_aperture_FWHM"//fwhm_multiple//".info,temp_frame_subset_FWHM"//fwhm_multiple//"
.info",>"avg_frame_fwhm_aperture_EW_FWHM"//fwhm_multiple//".info")
     joinlines
("temp_HJD.info,temp_frame_fwhm_FWHM"//fwhm_multiple//".info,temp_frame_subset_FWHM"//fwhm_multiple//".info",>"
avg_frame_fwhm_EW_FWHM"//fwhm_multiple//".info")
}
;
if (phot_meth=="Flux"&&pier_side=="No"){
     print ("")
     print ("Creating avg frame fwhm aperture and frame fwhm .info file")
     joinlines
("temp_HJD.info,temp_frame_fwhm_aperture_flux"//flux_percent//".info,temp_frame_subset_flux"//flux_percent//".i
nfo",>"avg_frame_fwhm_aperture_NA_flux"//flux_percent//".info")
     joinlines
("temp_HJD.info,temp_frame_fwhm_flux"//flux_percent//".info,temp_frame_subset_flux"//flux_percent//".info",>"av
g_frame_fwhm_NA_flux"//flux_percent//".info")
}
;
if (phot_meth=="FWHM"&&pier_side=="No"){
     print ("")
     print ("Creating avg frame fwhm aperture and frame fwhm .info file")
     joinlines
("temp_HJD.info,temp_frame_fwhm_aperture_FWHM"//fwhm_multiple//".info,temp_frame_subset_FWHM"//fwhm_multiple//"
.info",>"avg_frame_fwhm_aperture_NA_FWHM"//fwhm_multiple//".info")
     joinlines
("temp_HJD.info,temp_frame_fwhm_FWHM"//fwhm_multiple//".info,temp_frame_subset_FWHM"//fwhm_multiple//".info",>"
avg_frame_fwhm_NA_FWHM"//fwhm_multiple//".info")
}
;

#deletes possibly incompatible datalists and other files
delete ("*temp*")
delete ("star-*")
delete ("*.comp")
delete ("nomoreqs")

#adds date and time of when script was applied to frame headers
date '+DATE:%m/%d/%y%nTIME:%H:%M:%S' > "datalist"
tabpar ("datalist",1,1)
datestring=tabpar.value
tabpar ("datalist",1,2)
timestring=tabpar.value

hedit ("*"//searchalignphotometry//"*.fits","AP_APPHOT",datestring//""//timestring,add+,ver-)

#deletes possibly incompatible datalists
delete ("data*")

beep
```

References

Da Costa, G.S., 1992, PASPConference Series, Vol. 23, p. 91-97

Howell, S.B., 1992, PASPConference Series, Vol. 23, p. 112-119

Jeffery, E.J., 2003, senior thesis

Joner, M.D., 2008, private communications

SIMBAD Database 2006, (http://simbad.harvard.edu/cgi-bin/WSimbad.pl)

Swenson, C.A., 2008, senior thesis

WEBDA Database 2007, (http://www.univie.ac.at/webda)