Searching for Brown Dwarf Binary Systems:

Testing the PSF Fitting Code

Isaac James Christensen

A senior thesis submitted to the faculty of

Brigham Young University

in partial fulfillment of the requirements for the degree of

Bachelor of Science

Denise Stephens, Advisor

Department of Physics and Astronomy

Brigham Young University

April 2022

Copyright © 2022 Isaac Christensen

All Rights Reserved

ABSTRACT

Searching for Brown Dwarf Binary Systems:
Testing the PSF Fitting Code

Isaac Christensen
Department of Physics and Astronomy, BYU
Bachelor of Science

Since the first brown dwarf stars were discovered in 1995 many new observations have been made on them, but there is much left to be discovered about these dim stars. Their existence seems to defy many theories of stellar formation because of their small mass. In order to better understand these types of systems, a point spread function fitting code has been developed here at Brigham Young University that works to identify new brown dwarf binary systems by analyzing photometric data taken by the Hubble Space Telescope. One of the reasons for using this code is to help eventually confirm or dispute the results of previous publications made at BYU which attempt to find new binary brown dwarf systems. Photometric data is normally much easier to acquire than spectral data and therefore it is extremely important that we have an effective method of analyzing it. This is the main purpose of this code. In my research I was primarily responsible for running this code on 42 brown dwarf systems in order to find new binary candidate systems and to test how effective this code is in doing so. These results will be used to improve this code so that we will be able to more confidently identify binary brown dwarf systems and learn more about our universe.

Keywords: Brown Dwarf, Photometry, Hubble Space Telescope, PSF, Point Spread Function, Binary, NICMOS 1

ACKNOWLEDGEMENTS

I would like to acknowledge my research advisor, Dr Denise Stephens, for the many hours she spent teaching, guiding, and supporting me in my pursuit of my degree. She has been a great example to me and has helped inspire me to continue my education.

I would also like to thank the current and former minds that have worked on this PSF code and helped pave the way for my research, namely Loic Beus and Dr Tom Stephens.

Finally I would like to thank my family and my amazing wife for their amazing support in all of my endeavors.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Brown Dwarfs

The first brown dwarf was discovered in 1995 by a team in Tenerife, Spain (Rebolo et al. 1995). This curious object had the properties of both a star and a planet and was given the name “brown dwarf”. Since then over 2500 brown dwarfs have been observed. These objects contain characteristics which do not line up with our current understanding of stellar formation. To better understand these odd objects it is essential to improve our methods of detecting them, specifically binary systems (Beus et al. 2020). The goal of this research is to improve our PSF detection model to help us find new brown dwarfs that will help us better understand how these stars are formed (Eberhard 2020 & Turner 2020).

1.1.1 Background

When discussing stars one must recognize that they come in a variety of sizes, temperatures, and colors. We can determine these characteristics by analyzing the light coming from these objects using photometry.

Photometry is a technique used in astronomy which is used to measure the flux or intensity of light radiated by an astronomical object. This is done by looking at the object

through a telescope which has a photometer attached, often a CCD, which is a device that converts light into electrical currents via the photoelectric effect.This allows astronomers to continuously gather light off of an astronomical object to make an image which contains the information used to make a Point Spread Function. The photoelectric effect is made possible by the quantization of energy. Electrons contained within an atom can be in different energy states. The difference between two states, called n_1 and n_2, is known as the energy gap, or E_{gap} (Equation 1.1).

$$E_{gap} = E_{n1} - E_{n2} \tag{1.1}$$

For an electron to be able to move up or down to a different state it has to gain or lose the amount of energy equal to E_{gap}. The energies of these states of an atom depend on its atomic number Z and state number n (Equation 1.2).

$$E_n = \frac{-13.6Z^2}{n^2} \tag{1.2}$$

This loss of energy occurs when an electron either absorbs or emits a photon with energy equal to $E_{gap.}$ This phenomena allows for photometers within the telescopes to collect information from the photons that hit them, and transfer energy to it via this effect. One of these types of images is seen below in figure 1.1.

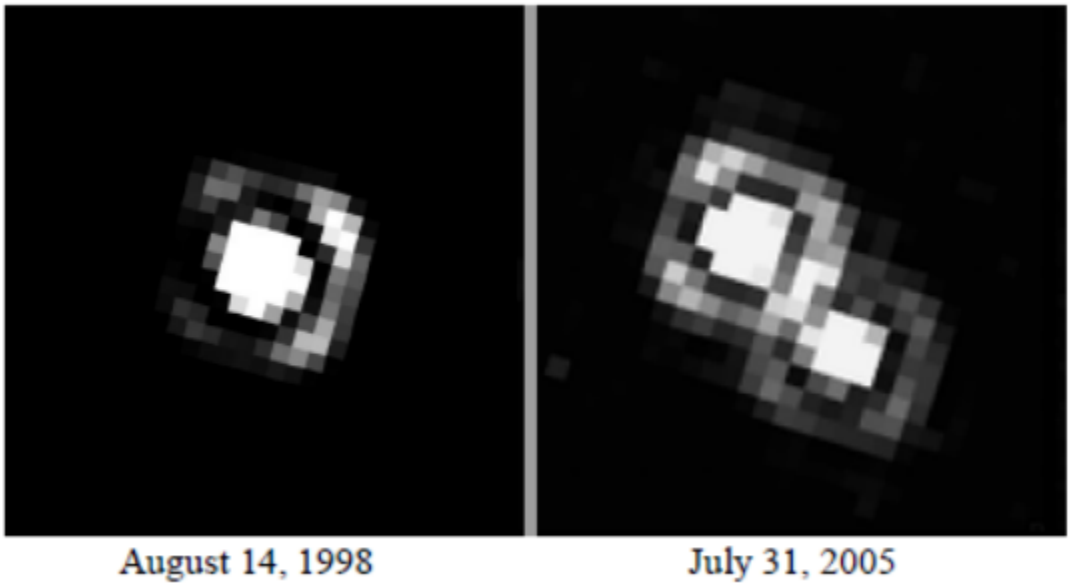

Figure 1.1 The same brown dwarf binary pair imaged in 1998 (left) and 2005 (right) by HST. Note that as technology improved the binary was resolved. Image obtained from (NASA et al. 2009) Credit NASA, ESA,and M. Stumpf (Max-Planck-Institute for Astronomy)

1.1.2 Discovery and Classification of Brown Dwarfs

Although the first brown dwarf was not discovered until 1995 (Rebolo et al. 1995), their existence was theorized years earlier by Shiv S. Kumar (Kumar 1963). He carried out a computational experiment in which he created mathematical models of the formation of stars. Kumar's models showed that, if the new star was below a mass of about 0.09 times the mass of the sun, the star would fail to "ignite," or begin its hydrogen fusion. Kumar initially named these failed stars "black dwarfs," but their name was later changed to "brown dwarf." His project was the beginning of mankind's study of these unique objects.

L and T Dwarfs

In 1999, four years after the discovery of the first brown dwarf, J. Davy Kirkpatrick led a team that searched through data taken by the 2-Micron All-Sky Survey (2MASS) for objects that looked cool enough to be brown dwarfs. The temperature of the object can be estimated from the color, which is found by subtracting the magnitude of the object found using a K telescope filter from the magnitude found using a J filter. This J-K magnitude difference is called the "color." After compiling this list of potential brown dwarfs, Kirkpatrick obtained spectra of these objects using the Low Resolution Imaging Spectrograph (LRIS) located at the W. M. Keck Observatory in Hawaii. The team noticed similarities in 25 of the spectra of these cool objects. Mainly, the spectral shape showed that the objects contained neutral alkali metals and metallic hydrides. The 26th object, a brown dwarf called Gl 229B originally discovered by another team (Golimowski et al. 1998) was the odd-one-out, containing more methane than the other spectra showed. The team decided to group the first 25 objects together into a new spectral type: Type L. In his paper, Kirkpatrick proposed that brown dwarfs cooler than type L objects be called “T dwarfs” and noted that these brown dwarfs were likely to contain a high amount of methane similar to the odd 26th object, Gl 229B (Kirkpatrick et al. 1999). Further investigation into the T spectral type was performed by Adam Burgasser, Kirkpatrick, and others the following year, confirming the high abundance of methane that is now the main notable feature in T type spectra (Burgasser et al. 2000). For the first time, definite characteristics of these elusive objects had been determined. Many of the objects analyzed in my research also came from the 2MASS survey as well.

Y Dwarfs

Recently, a group led by Michael Cushing discovered and defined the final spectral type: the Y dwarfs. Cushing’s team sorted through data obtained by the Wide-field Infrared Survey

Explorer (WISE) and identified seven brown dwarfs that were notably cooler than L and T dwarfs (Cushing et al. 2011). The team discovered that these ultra-cool brown dwarfs could be spectrally differentiated from L and T dwarfs by their high abundance of CH4, H2O, and NH3. However, in the few years since Cushing's work there has not been a telescope in space capable of obtaining the high-resolution spectral data needed to more deeply investigate the characteristics of these objects, the coolest and dimmest of the brown dwarfs and therefore the hardest to observe.

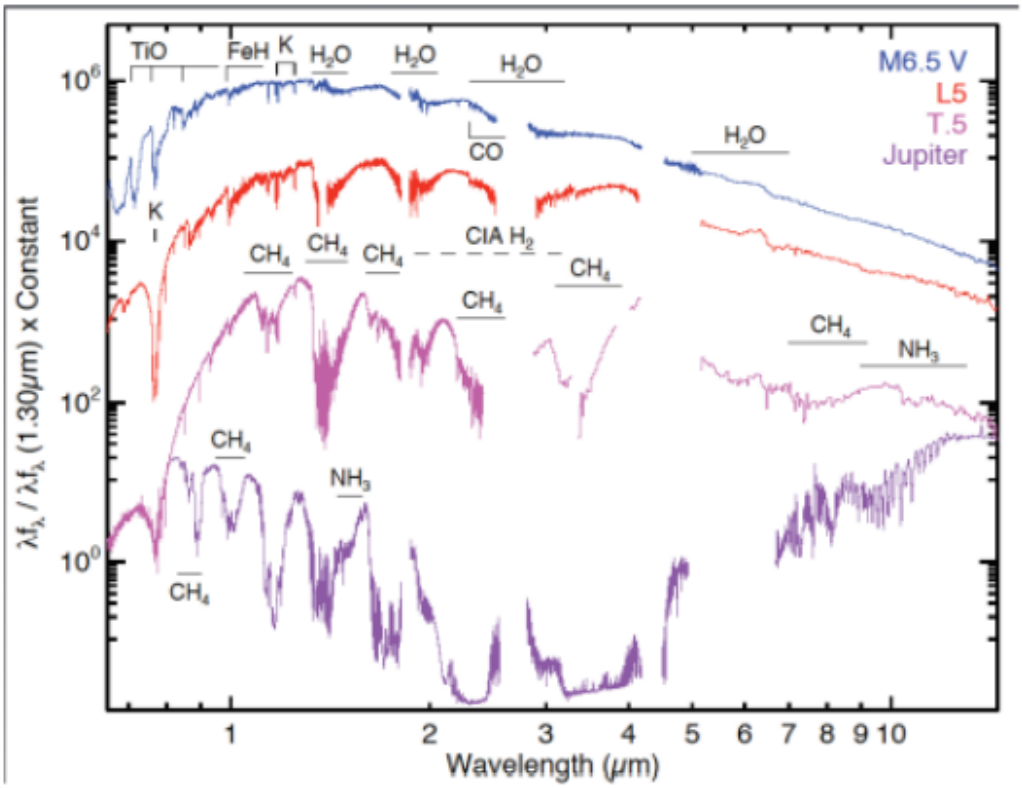

Figure 1.2 A comparison of spectra from various objects. From top to bottom: M star, L dwarf, T dwarf, Jupiter (planet). The gaps in the spectra are due to the earth's atmosphere absorbing specific wavelengths of light. The figure has been modified from (Marley & Leggett 2009).

1.2 The Importance of Binary Systems

Although brown dwarf research has progressed substantially by classifying three types of brown dwarfs and identifying prominent molecules in their atmosphere, there are many questions concerning these unique objects that remain to be answered. This section will address some of these questions, mainly how the current model of stellar formation fails to explain the existence of brown dwarfs and possible theories that have been proposed to rectify this issue.

1.2.1 Questions About Brown Dwarf Formation

The prevailing model of stellar formation involves the collapse of a large gas cloud in young areas of galaxies. Beginning astronomy courses teach that the simple math behind this centers around a critical mass, called Jean's mass, that determines the stability of a gas cloud of radius R and temperature T. This equation includes the Boltzmann constant k, G is the universal gravitational constant, μ is the mean molecular mass, and m_H is the mass of a hydrogen atom (Equation 1.3).

$$M_{Jean} = \frac{h}{f} = \frac{5kTR}{Gm_H\mu} \tag{1.3}$$

When the mass of a cloud becomes greater than Jean's mass, the cloud collapses in on itself, this collapse will fragment so that there are multiple centers of collapse. These centers eventually form protostars, which steadily increase in temperature and mass until their density, temperature, and pressure are high enough for nuclear reactions (mainly the fusion of hydrogen atoms) to ignite, which form a new star.

An important concept to understand is that Jean's mass is more of an approximation which works well to explain the formation of larger objects in the universe but fails to account for the formation of smaller objects such as brown dwarfs. These situations require many questions to be answered to be able to formulate a more complex model to explain their formation.

For one, fragmentation occurs when gas clouds collapse in order to form small objects, but when we analyze the physics of the collapse it is apparent that there must be some limit to how small an object can form via fragmentation. This process is as follows; the smaller the fragment of the gas cloud, the greater the opacity of the fragment. A larger opacity means it is more difficult for photons to escape the gas fragment. Eventually the fragment reaches a point where photons are unable to escape and the energy resulting from gravitational collapse is unable to be radiated away. This energy has to go somewhere in order to obey the law of conservation of energy, and much of it ends up going into causing more turbulent motion of the gas in the fragment. This increased turbulence causes outward pressure which pushes back against the force of gravity and prevents the collapse. In summary, there must be some lower mass limit where fragmentation stops due to turbulence, but is it unknown where exactly this limit is. Fragmentation cannot continue indefinitely, so how small of an object can actually be formed in this manner?

Furthermore, a trend has been observed in the relative abundances of different types of stars. Generally, the less massive spectral types are the most abundant in the universe, explained when we consider our fragmentation model where a gas cloud splits into a few large protostars, a few more medium protostars, and many small protostars. This trend is called the Initial Mass Function (IMF). However, in recent years it has been observed that the IMF drops off

dramatically past M dwarfs. There are significantly fewer brown dwarfs than there are M dwarfs, breaking the trend that we see for the other spectral types, and the astronomical community has yet to learn why (Thies et al. 2015).

1.2.2 Proposed Adaptations and the Binary Fraction

Multiple research groups have tried to come up with adaptations to stellar formation theory that would explain how brown dwarfs are formed. Shortly after the discovery of brown dwarfs, Bo Reipurth and Cathie Clarke (Reipurth & Clarke 2001) proposed that brown dwarfs were normal protostars that were somehow catapulted from the center of the molecular cloud before they could accrete enough mass to become fully-fledged stars and begin hydrogen fusion. The next year, Paolo Padoan and Ake Nordlund (Padoan & Norlund 2002) proposed that the requirements of Jean's Mass could be waived when turbulence in the gas cloud triggered gravitational collapse of smaller-mass clouds. Another research duo theorized that the growth of small protostars forming near stars of the hottest spectral types, O or B stars, could be stunted by the radiation pressure of the large number of photons radiating out from the O or B star. They call this process "photo erosion," and propose that this could stop the small protostar from gaining enough mass to become a true star, leaving behind a brown dwarf instead (Whitworth & Zinnecker 2004). Other models exist, and more are on the way. All of these models seem to make sense in some way, but it is difficult to know which if any are true.

A key to supporting or rejecting these proposed adaptations to stellar formation theory is the brown dwarf binary fraction. This term refers to the number of brown dwarfs that exist in binary pairs orbiting each other divided by the total number of brown dwarfs. Each unique formation model implies its own brown dwarf binary fraction, as different methods of formation are more likely to result in the formation of two brown dwarfs versus one lone brown dwarfs.

Therefore, if an accurate binary fraction can be estimated we will be much closer to understanding the role of brown dwarfs in stellar formation theory (Gagliuffi et al. 2014).

1.3 The Difficulty of Resolving Binary Pairs

As previously mentioned, finding an accurate binary percentage will help narrow-down the viable options for stellar formation theories. However, there is a physical concept called the Rayleigh criterion that makes visually observing brown dwarf binaries difficult. Brown dwarfs are smaller than your typical star, which thanks to the mathematics of orbits allows brown dwarf binaries to be much closer to each other. On top of this, brown dwarf binaries tend to be abnormally close to each other compared to other stellar binary systems, even when their small masses are taken into account. The smallest angular separation a telescope can resolve is limited by Rayleigh's equation where λ refers to the wavelength of light and D refers to the diameter of the telescope (Equation 1.4).

$$\Theta = \frac{1.22\lambda}{D} \tag{1.4}$$

Many brown dwarf binaries have angular separations that fall under this cutoff angle, meaning that they appear as one over luminous brown dwarf rather than as two separate objects. See Fig. 1.3 for a visual of what happens when two objects are unresolved. As technology has improved to allow us to make better and bigger telescopes, we have been able to resolve smaller separation angles. See Fig. 1.1. However, there are still many brown dwarfs that have been flagged as "over luminous" that are thought to be unresolved binary pairs.

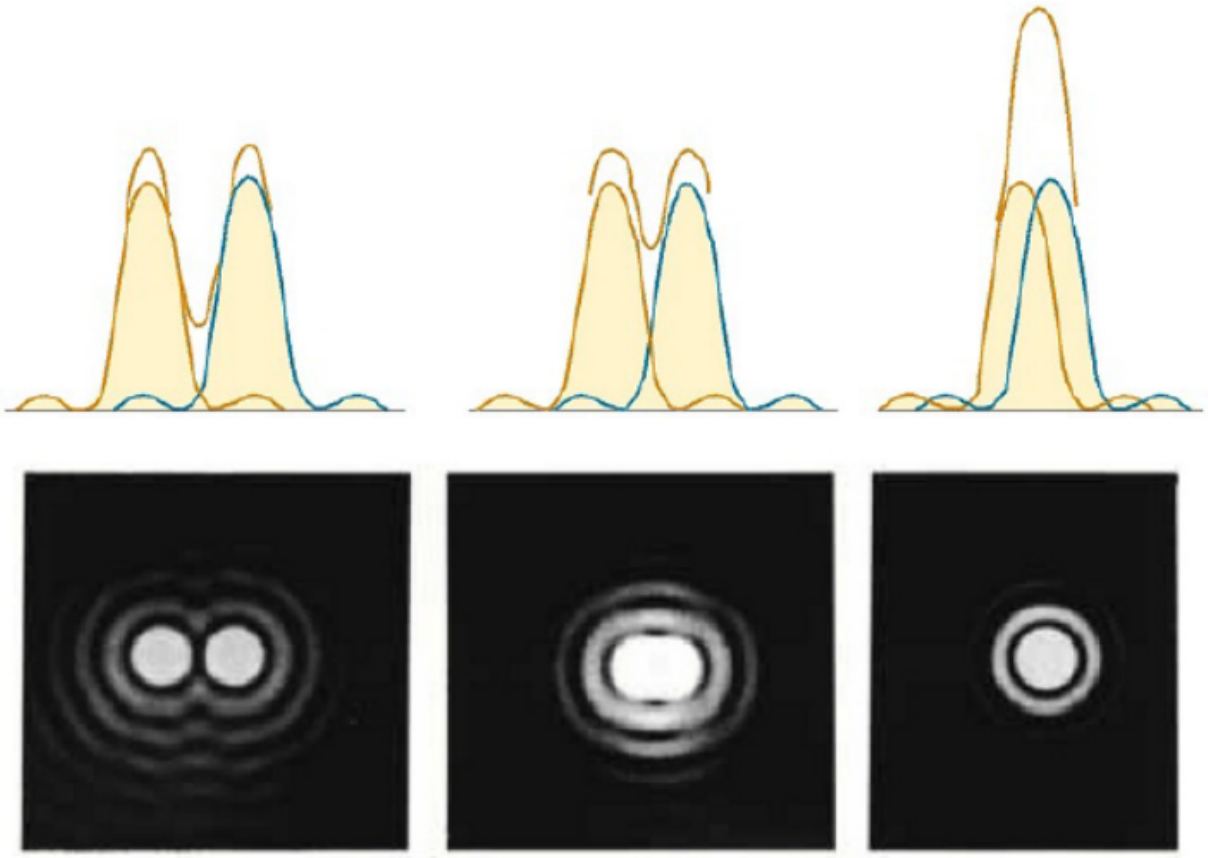

Figure 1.3 The progression of resolved to unresolved telescope images of a pair of objects placed close together. Figure taken from (Gardner 2015), who recreated this image from http://www.kshitijiitjee.com/Study/Physics/Part7/Chapter38/30.jpg

Chapter 2

Point Spread Function Fitting

2.1 Other Methods of Binary Identification

As a result of the difficulty of visually resolving brown dwarf binary pairs, there are some other methods for observing brown dwarf binaries that have emerged in the last few years.

One of these methods is spectral fitting. This involves the statistical analysis of the spectral data of the target object as compared to model spectra. This analysis is used to determine the likelihood of whether a system is a binary star system or not by comparing real spectral data to models of binary spectral data. Other members of my research group have used this in their research (Turner et al. 2020; Eberhard 2020).

The other primary method is Point Spread Function (PSF) fitting. This focuses on using the known characteristics of the telescope being used to determine how the light from a distant source should spread out over the CCD, and then comparing the theoretical PSF function to the actual data obtained by the telescope in order to determine if it is more likely that one or two objects are creating the PSF. I will focus on the process of PSF fitting since this is the method my research used to get its results.

2.2 Prior Work in PSF Fitting

The original PSF fitting program was created to detect binary trans-Neptunian objects (Stephens & Noll 2006), which was subsequently modified to detect binary brown dwarf systems using

data from the HST. The program was effective but time consuming as it required a lot of data preparation and user input. The next iteration was written in Python with the goal of reducing the user interaction of the program (Matt 2017). This code was written in Python 3 which allowed the code to directly process the images. The end-product was a modular program that required little preparation and user interaction and could be easily modified to work with various instruments and cameras. However, even though the program was easy to use, it was very slow. The FORTRAN program only took approximately 10 seconds to run after the initial preparation phase while the python code could take anywhere from 7 to 15 minutes. This is a significantly longer run time and made it extremely time consuming to run the code on large sets of data. In addition to the long run-time, the Python code was discovered to have a few errors that were leading to inaccurate fits. Most of my research was spent on correcting these errors, improving the run-time, and adding a Monte-Carlo simulation. The current code most recently updated by Beus et al. 2020 worked to correct these errors, making it run faster, and added a Monte Carlo simulation.

3 Implementing the PSF Code

I used a master list of approximately 1500 potential brown dwarf systems which was compiled by my research team to search for HST NICMOS data. This was accomplished by inputting the right ascension and declination coordinates into the HST website to see if there was any data for each of these objects taken with the NIC 1 camera. I found that about 150 of the objects on this list had NIC 1 data and picked the first 42 of them to run the code on. Below is the link to this master list.

https://docs.google.com/spreadsheets/d/17i4YQ1D-l-7A_lqZfsIjkEv6Rc5H2uCjwXdrpverYPc/edit#gid=2048182490

3.1 Data Preparation

Before being able to run this Python code there are various steps that need to be done in order to prepare the image frames taken from the Hubble Space Telescope (HST) Near Infrared Camera and Multi-Object Spectrometer (NICMOS). This chapter will cover these steps needed to make the images usable by the code. Since I did not take part in writing the code used in this research, I will point those who are interested in exactly how the code works to **Beus et al. 2020** because that thesis covers how the code works in depth. For those interested in obtaining the code for their own use can go to https://github.com/lmbeus/HST-PSF-Fitting.

3.1.1 HST NICMOS 1

The HST NICMOS instrument has 3 main cameras used to image objects. The python code that I used was best suited to run data taken by the NIC 1 camera because of the particular filters (F110W and F170M) which the NIC 1 camera works best in. These filters are also good for picking up faint objects such as brown dwarfs. More information on this instrument can be found

on https://www.stsci.edu/hst/instrumentation/legacy/nicmos in the handbooks posted there. Tiny Tim, Pedsky, and rnlincor are meant to be executed in a Linux terminal.

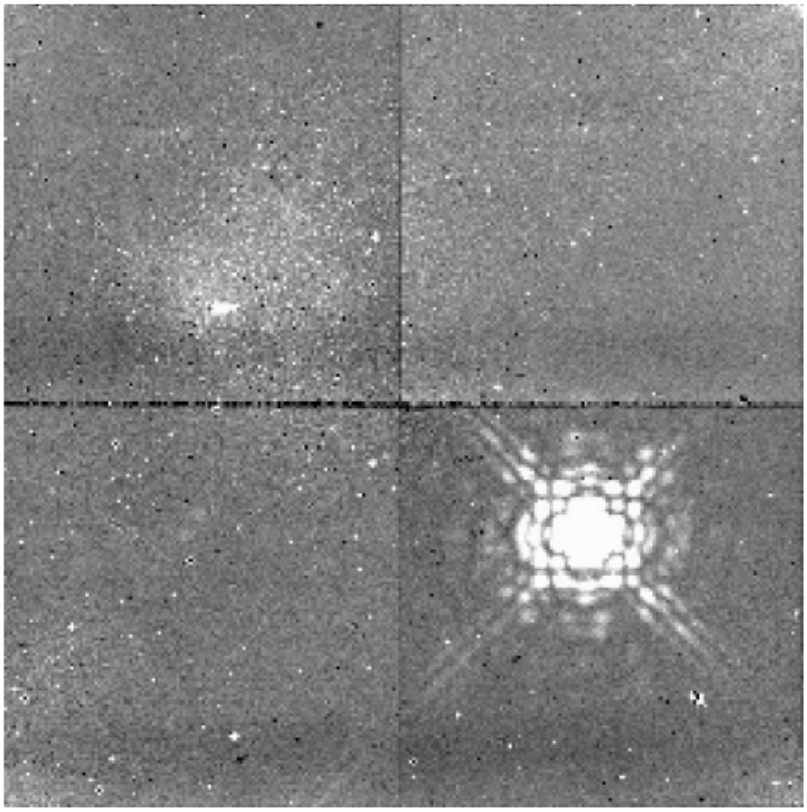

Figure 1.4 HST image of 2MASSJ203603+10 using the NICMOS 1 camera in the F170M filter.

3.1.2 Tiny Tim

The first step in preparing the data for use in the code is to create the model PSF using the Tiny Tim software. Tiny Tim gives us the ability to create model PSF's for NICMOS 1 and NICMOS 2 images in various filters and configurations. The Tiny Tim software and user manual can be found at:

https://www.stsci.edu/hst/instrumentation/focus-and-pointing/focus/tiny-tim-hst-psf-modeling

When running the Tiny Tim software you are prompted to input the camera, instrument, filter, object coordinates, and spectral type of the brown dwarf. The following are the step by step instructions I used to run the Tiny Tim software which is a modified version from Matt (2017).

> tiny1 F170M.in Where F170M is the name of the file where Tiny Tim will save the parameters to. We use the filter name which for this example is F170M.

Tiny Tim will then prompt the user for the instrument and camera. Be aware that the NICMOS1 camera has two options, one for before 2002 and one for after 2002 which is when the new cryocooler was installed.

Next the user is asked to enter the x- and y- coordinates of the center of the object rounded to the nearest integer and each coordinate separated by a space.

: XXX YYY

For the NICMOS camera you will be asked if the image was taken after 2002. If true, enter Y, otherwise enter N.

: Y

The user is then prompted for the filter name in lower caps. In this example we are using F170M as the filter.

: f170m

Tiny Tim then gives a few options to build the spectrum, in this case we use an ASCII table of the brown dwarf's spectrum, which is option 5.

: 5

The description of the ASCII tables are given in Appendix C of the Tiny Tim manual (Krist & Hook 2004). In this example our spectrum is the file L4.txt

: L4.txt

The user is asked whether they want to subsample the PSF. For our model fitting program, we subsampled the PSF by 10 to get our .1 pixel resolutions as described in section 2.1.

: Y

: 10

Tiny Tim prompts for the secondary mirror despace which is 0 for our purposes

: 0

The user is then prompted for the filename of the output filename of the PSF model. Again, we use the filter name which in this case is F170M.

: F170M

Tiny Tim then saves this information to the parameter file which was named in the beginning.

To generate the PSF model for the NICMOS camera we type in the terminal:

> tiny2 F170M.in

Tiny Tim then generates a PSF model with the name we entered with “00” tacked on at the end. For our example this would be F170M00.fits.

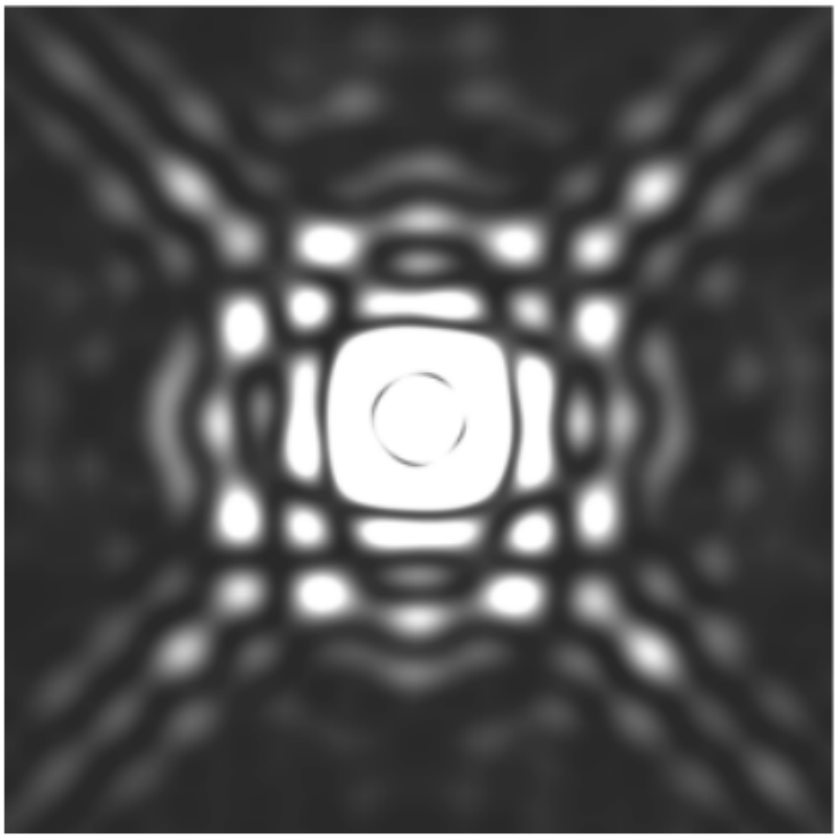

Figure 1.5 A model PSF image generated by Tiny Tim.

3.13 PEDSKY & RNLINCOR

After making the Tiny Tim model PSF's the next step is to process the images and prepare them to be run through the fitting code. NICMOS images are not linear so they need to be corrected in order to accurately calculate magnitudes. The images are corrected in PyRAF which is a Python-based version of IRAF (Image Reduction and Analysis Facility). IRAF and PyRAF are softwares that were developed by the National Optical Astronomy Observatory (NOAO) to reduce and analyze astronomical data. PEDSKY and RNLINCOR are two packages found in PyRAF which I used to remove the estimated sky background, residual bias, and to correct for the non-linearity in the images. The following commands are used to get these packages:

> stsdas

> hst_calib

> nicmos

After inputting these packages I then ran the commands to apply pedsky and rnlincor to the images. I will use the example image name n8qt03mhq_cal.fits. To apply the correction to my images I ran the pedsky function on the cal.fits image followed by the output file name which ends with ped.fits which means that pedsky has been run on it. Then I ran the rnlincor command on the ped.fits file and output it to a rnl.fits file which is the final corrected image which I ran through Beus's PSF fitting code. Below are the example commands that I typed in for each cal.fits file

> pedsky n8qt03mhq_cal.fits n8qt03mhq_ped.fits

> rnlincor n8qt03mhq_ped.fits n8qt03mhq_rnl.fits

3.1.4 Setting Up the Code in Linux and Ubuntu

As mentioned previously, the code runs easiest when running on Ubuntu in a Linux computer or virtual machine with anaconda installed. Installing anaconda on Linux is simple and is explained here: https://docs.anaconda.com/anaconda/install/linux/. This needs to be installed in the same directory that the code files are located. Now that the HST images are corrected for non-linearity the data can be run through the model fitting code. The _rnl.fits image file needs to be placed with its corresponding Tiny Tim generated PSF model in a single directory along with the PSFfitting.py, PSF_prep.cpp, single_fitting.cpp, binary_fitting.cpp, rn_single.cpp, rn_binary.cpp, and cameras.json files discussed in Beus et al. 2020. Before the code can be run

the proper packages must be installed. The code requires the user to have both Python 3 and a C++ compiler that supports C++14 or higher. For Python the numpy, astropy, and photutils packages are required. For C++ the external, third-party xtensor and xsimd packages needs to be installed. Instructions for installing xtensor and xsimd can be found at:

https://xtensor.readthedocs.io/en/latest/installation.html

https://xsimd.readthedocs.io/en/latest/installation.html

As discussed in Beus et al., xtensor is a C++ library designed for numerical analysis with multi-dimensional array expressions. xsimd is essentially a wrapper that can be used for xtensor to make reduction calculations faster (such as "sum" functions). To compile the .cpp files the paths to xtensor, xtl (comes installed with xtensor), and xsimd must be included in the compile command. In addition, -mavx2, -ffast-math, -DXTENSOR_USE_XSIMD -03, and -openmp must also be added to the compile line. For my machine my compile command for the PSF_prep.cpp file would be:

> g++ -mavx2 -ffast-math -DXTENSOR_USE_XSIMD -O3 -openmp -I /home/isaacjc2/anaconda3/pkgs/xtensor-0.21.10-h0efe328_0/include -I /home/isaacjc2/anaconda3/pkgs/xtl-0.6.21-h0efe328_1/include -I /home/isaacjc2/anaconda3/pkgs/xsimd-7.4.9-h0efe328_1/include/ PSF_prep.cpp -o PSF_prep

Once all five of the .cpp files are compiled the PSF fitting code can be called by running the following command in the terminal:

> **Python3 PSFfitting.py F170M00.fits n8qt03mhq_rnl.fits**

The results of the fit can be found in the file imagename_output.txt in the same directory, where “imagename” is the name of your image file. In my example the output file would be n8qt03mhq_output.txt. One of my output.txt files can be seen below:

```
n8nh15xgq - F110W
UT 2004-07-22 21:11:41

Single Fit:
Iteration: 1 PSF: 48 Flux: 743.8499134283066 Min Chi:
15.086654580865527 Residual Error: 0.9524354613553648
Iteration: 2 PSF: 48 Flux: 743.8499134283066 Min Chi:
15.086654580865527 Residual Error: 0.9524354613553648
      ---
Binary Fit:
1   error:0.62780
 Primary: (169.14,101.65)  Secondary: (168.24,101.45)
 Primary PSF:   38 f:  694.5883 (91.46%)
 Secondary PSF: 20 f:   64.8199 ( 8.54%)
 Position Angle: 117.80 Separation: 0.0397
 Primary Magnitude: 15.666 Secondary Magnitude: 18.242
```

3.2 How the Code Works

As mentioned previously, I did not take a major part in the writing of this code so to explain the specifics of how the data is fit to a PSF and compared to the model PSF I defer to Beus et al 2020, specifically the 2nd chapter of his thesis explains how his code works.

Chapter 4 Results and Conclusions

4.1 Results

As mentioned previously one of the main purposes of my research was to see how consistent the code is in successfully identifying binary brown dwarf systems. For each rnl.fits image that I ran through the code, histograms were made from the monte carlo simulation mentioned by Beus et al. 2020. These histograms were meant to show the flux distributions for a single fit as well as a binary fit as shown below in figures 1.6 and 1.7.

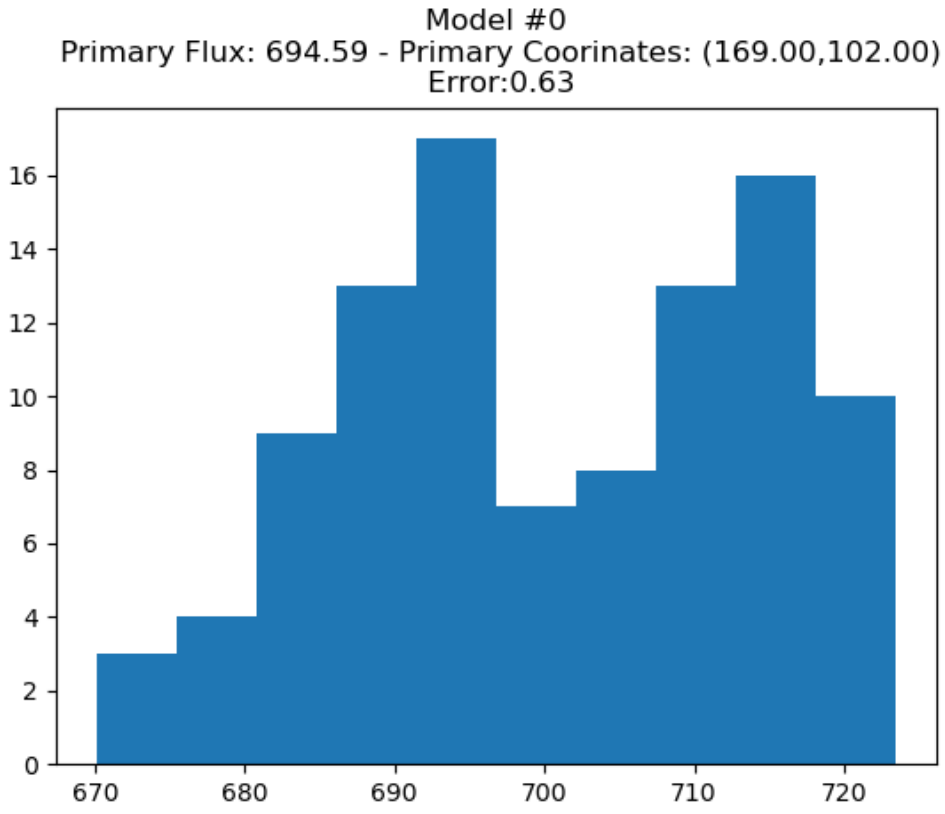

Figure 1.6 The best Binary fit for 2MASSIJ0523382-140302. This PSF has two flux peaks where the two brown dwarfs would peak if this was a binary system. Note that this histogram was only made with 100 runs, more runs would form a smoother PSF but take a lot longer for the code to run.

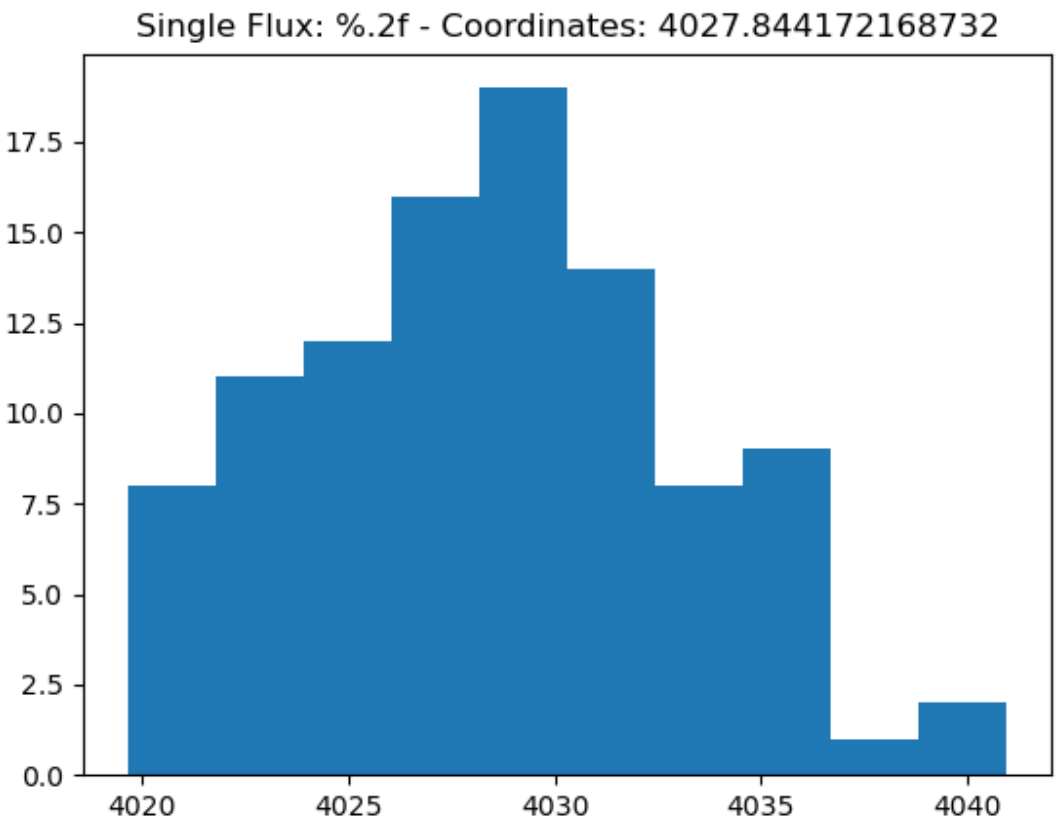

Figure 1.7 Here is a PSF for 2MASSIJ0523382-140302 made in the same method as figure 1.6 but for a single fit.

4.1.1 Likely Binaries

This code was run on 42 systems. Due to limitations in the code which will be discussed in the next section, I had to determine the likelihood of a system being a binary system by looking at the consistency between all of the rnl.fits images for that system, the flux agreement, and the separation angles. The flux agreement is the plus or minus width of the PSF given by the histogram. For Figure 1.7 this width would be approximately plus or minus 12. With these methods I was able to rule out 22 systems as not being binaries with another 16 being inconclusive due to limitations in the code and 4 being binary candidates. The relevant PSF output was organized in a color coded spreadsheet which is in the next 5 pages under Table 1.

Table 1 (next 5 pages): The four values used to determine the likelihood of being a binary candidate are contained in this table. They are the binary primary flux, secondary flux , separation angle, and flux agreement. I color coded these systems as follows: green for binary candidate, yellow for inconclusive results, and red for unlikely to be a binary system.

Object Name	RNL files	Filters	Single Fit Flux	Single Fit Resid	Binary Residual	Primary Flux	Secondary Flux	Position Angle	Separation	Flux Agreement
Example Data		170M	1833	0.7275392357		1231.0308	665.2601	31.08	0.0347	
2MASSIJ0213288+444445										
	n8yj63wlq_rnl	170M	2602	1.520761817	0.41974	2262.2543	360.0545	143.82	0.0304	210
	n8yj63wnq_rnl	170M	2620.532765	1.084813527	0.54195	2357.5742	286.0877	0.68	0.0304	170
2MASSIJ0243137-245329										
	n8nh01o9q_rnl	170M	149.5778661	0.09646006771	0.06174	140.1183	10.4211	103.99	0.0193	90
	n8nh01obq_rnl	170M	151.4289141	0.07265893134	0.0383	54.8434	99.8431	22.12	0.0272	100
	n8nh01ocq_rnl	170M	152.9040676	0.03408176392	0.0256	148.0912	5.0625	161.52	0.0251	40
	n8nh01ofq_rnl	170M	147.6909917	0.1226757214	0.07075	131.5392	19.8714	6.86	0.0465	60
2MASSIJ0415195-093506										
	n8nh03koq_rnl	170M	354.8822143	0.9259182233	0.30567	330.4281	32.1988	160.94	0.043	80
	n8nh03kpq_rnl	170M	354.3393662	0.7976379672	0.2978	139.0692	225.011	24.6	0.0262	80
	n8nh03ksq_rnl	170M	355.4884541	0.5848866604	0.16594	138.8781	217.8392	150.46	0.0096	100
2MASSIJ0439010-235308										
	n8yj12b2q_rnl	170M	1308.862304	0.7121127208	0.23029	644.0977	671.8014	8.66	0.0155	1050
	n8yj12b5q_rnl	170M	1318.402802	0.5773498678	0.22776	260.2904	1060.9925	23.92	0.0136	900
2MASSIJ0445538-304820										
	n8yj47udq_rnl	170M	3052.604023	1.372961765	0.44299	634.834	2427.0545	158.15	0.0096	300
	n8yj47ufq_rnl	170M	3023.801062	1.324704404	0.40192	1356.5726	1667.9936	15.02	0.0096	110
2MASSIJ0523382-140302										
	n8yj27yxq_rnl	170M	4027.844172	1.975737825	0.79203	3876.7629	186.885	128.77	0.0396	110
	n8yj27yzq_rnl	170M	4071.699452	3.043057182	0.5235	2042.943	2046.6792	159.73	0.0136	90
2MASSJ00250365+4759191AB										
	n8yj58vjq_rnl	F170M	1044.535536	10.44900511	6.03097	798.9058	473.2219	152.01	0.0735	45
	n8yj58vkq_rnl	F110W	4104.60734	10.93956366	8.42025	4104.5429	0			38
2MASSJ01443536-0716142										
	n8yj26z6q_rnl	F170M	1867.492061	1.176043418	0.30364	1403.5866	479.0637	27.37	0.0261	350
	n8yj26z8q_rnl	F170M	1901.008472	1.041379855	0.38954	1286.0713	621.0485	171.34	0.0192	250
2MASSJ01550354+0950003										
	n8yj55e5q_rnl	F170M	918.8789179	0.3188084613	0.19575	213.5195	713.2738	12.46	0.0136	88
	n8yj55e7q_rnl	F170M	934.9620422	0.6109381488	0.14564	892.7937	56.7231	54.86	0.0423	240
2MASSJ02284243+1639329										
	n9nk02mqq_rnl	F170M	3615.709504	1.08134681	0.75154	3231.0861	419.4322	38.31	0.0301	60
	n9nk02mzq_rnl	F170M	3608.941872	1.760777831	0.43046	3436.9095	226.9629	1.44	0.043	350

2MASSJ03185403-3421292										
	n9nk06vyq_rnl	F170M	547.4096167	0.2077708908	0.10807	530.0126	28.0697	156.93	0.0597	250
	n9nk06w4q_rnl	F170M	561.4108441	0.2228215321	0.12059	337.0718	233.3329	156.22	0.0251	260
2MASSJ03480772-6022270										
	n8nh02kvq_rnl	F110W	493.4063199	0.8549764778	0.44506	455.9219	44.7796	125.14	0.0312	140
2MASSJ03552337+1133437										
	n8yj08t7q_rnl	F170M	2923.528373	1.800734932	0.49659	2247.8065	723.9672	77.68	0.0273	800
	n8yj08t9q_rnl	F170M	3016.881833	2.112227687	0.67182	2614.1136	407.9955	118.28	0.0251	400
2MASSJ05002100+0330501										
	n8yj23jpq_rnl	F170M	2496.730797	1.557291798	0.37774	2343.8825	190.3242	21.83	0.0465	140
	n8yj23jrq_rnl	F170M	2505.762964	1.176612879	0.4234	2057.3798	454.1542	69.56	0.0177	140
2MASSJ05160945-0445499										
	n8nh04p6q_rnl	F170M	100.9471734	0.04069535093	0.02796	87.0371	14.0394	132.42	0.0096	90
	n8nh04p8q_rnl	F170M	102.824704	0.06099918544	0.039	98.3958	5.3718	142.72	0.0431	50
	n8nh04p9q_rnl	F170M	103.0479329	0.06458098122	0.02909	101.0231	4.8283	139.54	0.0776	30
	n8nh04paq_rnl	F170M	102.2005296	0.05981297278	0.0347	94.7492	9.591	174.05	0.0463	50
	n8nh04pcq_rnl	F170M	99.48963503	0.07614882396	0.04362	96.0652	4.1545	57.04	0.0457	50
2MASSJ05185995-2828372AB										
	na2605yzq_rnl	F170M	307.2921388	0.3708049437	0.17043	245.0695	74.238	169.75	0.037	140
	na2605z7q_rnl	F170M	317.1991773	0.2598134277	0.11656	237.0195	82.9792	141.85	0.0193	170
	na2605zfq_rnl	F170M	311.8960284	0.2910451608	0.16353	299.3451	16.0363	108.95	0.039	70
	na2605znq_rnl	F170M	316.8002366	0.3023169003	0.16668	311.9182	10.4916	81.6	0.0621	40
2MASSWJ0036159+182110										
	n8yj06deq_rnl	F110W	7093.327709	13.46550631	3.81685	2017.4656	5146.2367	156.9	0.0122	Inconclusive
	n8yj06dgq_rnl	F110W	34.55359155	0.6323936898	0.44236	34.166	1.354	154.02	0.0579	15
DENIS-PJ0255-4700										
	n9nk04ppq_rnl	F170M	4056.506979	2.901372246	0.82643	2082.2024	2066.9367	56.35	0.0261	1700
	n9nk04psq_rnl	F170M	4126.609077	1.945291448	0.52134	1256.0186	2898.6814	32.12	0.0155	2000
SDSSJ015141.69+124429.6										
	n8nh13djq_rnl	F110W	255.473396	1.548564146	1.51488	255.4734	0	62.95	0.0501	20
	n8nh13dnq_rnl	F110W	260.0960219	1.453640621	1.42808	169.0076	92.9943	121.98	0.0043	250
SDSSJ020742.48+000056.2										
	n8nh14nkq_rnl	F110W	103.0991641	0.1268226178	0.06664	66.6471	39.6388	177.06	0.025	80
	n8nh14nlq_rnl	F110W	105.299327	0.1586594715	0.0866	49.8293	58.9937	6.41	0.0232	100

SDSSpJ010752.33+004156.1										
	n8yyj39p5q_rnl	F170M	475.5512424	0.1623402244	0.10703	389.077	91.4226	18.64	0.0258	450
	n8yyj39p7q_rnl	F170M	473.1502451	0.1944626026	0.08729	384.4561	96.6648	10.51	0.0304	450
SDSSpJ042348.57-041403.5										
	n8nh15xeq_rnl	F110W	717.2081776	1.350673284	0.59092	643.1599	100.1811	137.31	0.0406	140
	n8nh15xfq_rnl	F110W	737.6935391	1.304449825	0.78219	695.5162	55.1822	92.84	0.0397	140
	n8nh15xgq_rnl	F110W	743.8499134	0.9524354614	0.6278	694.5883	64.8199	117.8	0.0397	55
	n8nh15xhq_rnl	F110W	750.0650419	1.027504552	0.77775	721.2324	49.8587	114.02	0.0568	140
	n8nh15xjq_rnl	F110W	750.261163	0.9445956546	0.68411	687.698	81.9098	86.83	0.0409	100
	n8nh15xkq_rnl	F110W	758.8664455	0.9683682166	0.6175	682.077	97.0472	92.74	0.0398	100
SSSPMJ0124-4240										
	n8yj65k6q_rnl	F110W	4525.750265	5.726643705	4.55551	4416.0695	132.1517	162.4	0.0356	30
	n8yj65k8q_rnl	F110W	4534.882539	8.421590764	3.82074	3996.0059	576.028	72.26	0.0177	60
SDSSpJ092615.38+584720.9										
	n8nh17hzq_rnl	F110W	271.6066574	1.517344265	0.37908	171.6218	158.2485	135.66	0.0646	60
	n8nh17i0q_rnl	F110W	272.5840644	1.429756322	0.20105	201.7875	133.3808	132.36	0.0693	50
	n8nh17i1q_rnl	F110W	286.6761706	1.729339867	0.33029	208.5673	136.8848	135.9	0.0689	60
	n8nh17i2q_rnl	F170M	146.0235964	0.1291819839	0.06163	54.221	106.4289	131.33	0.0607	130
	n8nh17i3q_rnl	F170M	146.2950133	0.1457317801	0.07116	77.2019	84.484	139.46	0.0601	110
	n8nh17i5q_rnl	F170M	148.7032089	0.1152647718	0.04826	96.5786	66.2704	131.33	0.0607	110
	n8nh17i6q_rnl	F170M	147.2356286	0.1753809748	0.04313	127.1538	32.3898	136.1	0.0731	80
	n8nh17i7q_rnl	F170M	146.3008023	0.1451360231	0.04649	102.8372	59.5818	128.15	0.0657	100
SDSSpJ083008.12+482847.4										
	n9nk03u5q_rnl	F110W	451.4720241	0.6484060255	0.27282	320.6377	136.5326	37.02	0.0155	350
	n9nk03u7q_rnl	F110W	457.058718	0.7095236087	0.25076	299.9877	161.0312	43.99	0.0096	350
2MASSsJ0850359+105716										
	n8qm01qeq_rnl	F170M	191.2958696	0.4309257209	0.14228	114.666	118.9117	125.05	0.0867	65
2MASSJ09211410-2104446										
	n8yj20tyq_rnl	F110W	5323.044378	5.493057304	1.97857	1120.4213	4292.3654	25.7	0.0244	3500
	n8yj20u3q_rnl	F110W	5503.743138	9.707100131	3.01596	4669.854	875.6461	14.46	0.0155	70
	n8yj20txq_rnl	F170M	3994.013873	2.253231264	0.72965	2642.9185	1437.8122	37	0.031	3000
2MASSJ09153413+0422045										
	n8yj10xqq_rnl	F110W	550.0695941	0.7997596088	0.23609	442.6378	111.4357	38.17	0.0137	200
	n8yj10xoq_rnl	F110W	545.8062368	0.6875451447	0.33812	531.6879	22.8112	71.87	0.049	40
	n8yj10xnq_rnl	F170M	627.2197241	0.376624349	0.13058	460.8531	173.9548	83.15	0.0192	200

2MASSJ09111297+7401081										
	n8yj52fnq_rnl	F110W	5269.615342	5.67597696	3.21138	4683.921	670.645	173.62	0.0262	Inconclusive
	n8yj52fpq_rnl	F110W	5282.100514	6.768715883	2.84562	4543.9433	782.7246	145.74	0.0135	Inconclusive
	n8yj52fmq_rnl	F170M	4043.729427	1.861942012	0.99695	364.7583	3727.5978	164.21	0.0301	Inconclusive
	n8yj52foq_rnl	F170M	4069.247868	2.252707906	0.66392	3901.0967	181.9907	142.41	0.0463	Inconclusive
2MASSJ07003664+3157266										
	n8yj19l7q_rnl	F110W	3835.546537	6.749680841	2.78941	1942.444	1978.4499	148.06	0.0232	25
	n8yj19lbq_rnl	F110W	3928.220146	6.632327459	2.69147	2591.5277	1447.11	160.35	0.0261	25
	n8yj19l6q_rnl	F170M	3974.713419	3.15738493	1.0136	2533.1297	1644.6567	136.16	0.0465	250
	n8yj19laq_rnl	F170M	4033.628035	2.736912688	1.3797	1907.3155	2374.7743	131.98	0.049	175
2MASSJ06244595-4521548										
	n8yj36k5q_rnl	F110W	1118.29549	1.195041442	0.89451	1116.7587	2.3808	145.94	0.0423	10
	n8yj36k7q_rnl	F110W	1110.666213	1.708341011	0.26602	995.0267	128.1691	140.58	0.0271	55
	n8yj36k4q_rnl	F170M	1433.436033	0.7603702034	0.25014	785.7347	667.8456	103.61	0.0136	400
	n8yj36k6q_rnl	F170M	1438.501152	0.4919308902	0.15315	1375.1254	75.195	151.79	0.0347	Inconclusive
2MASSJ05591914-1404488										
	n8qt03pwq_rnl	F145M	408.7504588	0.3945185054	0.16082	391.5458	20.9187	130.89	0.0464	15
	n8qt03q0q_rnl	F145M	409.7077323	0.4038789261	0.24361	365.0343	46.644	152.7	0.0216	50
	n8qt03pzq_rnl	F165M	1099.697557	0.4928982929	0.24231	1039.1179	83.6148	162.16	0.0525	60
	n8qt03q1q_rnl	F165M	1106.347149	0.5376323588	0.25268	1003.0052	126.7081	130.89	0.0464	100
2MASSJ03140344+1603056										
	n8yj31bhq_rnl	F110W	7506.386526	10.07492947	4.25582	7149.1604	461.96	1.8	0.0385	30
	n8yj31bjq_rnl	F110W	7556.814168	10.36825247	2.97198	6533.8227	1095.4758	28.32	0.0172	800
	n8yj31bgq_rnl	F170M	6122.112195	2.324872557	0.97929	4816.7134	1401.4129	18.86	0.0261	Inconclusive
	n8yj31biq_rnl	F170M	6217.555405	3.676252881	0.85563	489.3402	5789.4066	28.32	0.0301	Inconclusive
2MASSJ02572581-3105523										
	n8yj13zjq_rnl	F110W	956.443176	1.139867901	0.36217	515.9963	450.5542	148.67	0.0155	300
	n8yj13zlq_rnl	F110W	968.6607204	1.641143563	0.30968	749.48	235.8646	160.39	0.0232	200
	n8yj13ziq_rnl	F170M	1184.149277	0.9462420512	0.23176	1100.5839	103.8852	137.19	0.0548	250
	n8yj13zkq_rnl	F170M	1207.919257	0.628532486	0.24214	657.0825	558.1796	148.5	0.0155	300
2MASSIJ0908380+503208										
	n8yj43fyq_rnl	F110W	1118.632509	1.396264	0.63202	158.015	972.1911	94.39	0.0192	200
	n8yj43g0q_rnl	F110W	1132.242659	1.853815113	0.55275	330.9774	811.9473	131.28	0.0096	200
	n8yj43fxq_rnl	F170M	1250.821701	0.4649056902	0.13303	928.9832	326.8527	49.38	0.0136	200
	n8yj43fzq_rnl	F170M	1244.666864	0.8635217912	0.2222	103.7814	1153.1963	94.38	0.0192	200

2MASSIJ0859254-194926										
	n9nk08brq_rnl	F110W	387.6045176	0.4223430564	0.22923	380.6365	12.9373	87.58	0.0586	20
	n9nk08btq_rnl	F110W	394.0123727	0.7168012512	0.28905	364.8064	36.2594	111.89	0.0347	20
	n9nk08bqq_rnl	F170M	502.6426034	0.2737792863	0.10384	438.2801	72.6088	120.99	0.0367	100
	n9nk08bsq_rnl	F170M	518.2134556	0.178935509	0.08826	506.1242	15.3573	114.98	0.0481	30
2MASSIJ0847287-153237										
	n8yj53vfq_rnl	F110W	2830.024444	4.235242659	1.30735	2432.8979	453.2486	160.58	0.031	30
	n8yj53vkq_rnl	F110W	2904.856943	2.208645562	1.26139	2415.9382	494.8477	149.4	0.0061	Inconclusive
	n8yj53vjq_rnl	F170M	2720.176191	0.8732780433	0.37919	2621.7407	114.9219	149.27	0.0365	30
	n8yj53vkq_rnl	F170M	2671.348141	0.7807652853	0.69917	2172.3735	533.1128	4.81	0.0261	300
2MASSIJ0835425-081923										
	n8yj03tgq_rnl	F110W	3801.817697	6.909033316	2.48009	2418.1386	1432.5947	61.97	0.0177	150
2MASSIJ0825196+211552										
	n8yj11bgq_rnl	F110W	617.6754606	1.109602909	0.30257	191.53	439.0077	3.75	0.0252	150
	n8yj11biq_rnl	F110W	661.6178192	0.9443060223	0.32619	589.2633	77.9554	48.6	0.0177	30
	n8yj11bfq_rnl	F170M	904.439396	0.2373927574	0.16386	880.6109	26.8255	53.25	0.0261	50
	n8yj11bhq_rnl	F170M	936.9165681	0.5576731099	0.10816	197.5382	741.369	179.28	0.0096	500
2MASSIJ0755480+221218										
	n8nh06bkq_rnl	F110W	318.8741619	0.4042275948	0.15688	262.903	62.5616	38.39	0.0271	60
	n8nh06blq_rnl	F110W	326.5566314	0.2289609022	0.18095	326.5669	0	111.91	0.0526	7
	n8nh06bnq_rnl	F110W	330.9903096	0.4901666018	0.28773	311.9244	22.7711	11.92	0.0243	30
	n8nh06boq_rnl	F170M	111.8525713	0.06039050597	0.03172	108.3193	4.3458	35.12	0.0463	20
	n8nh06bqq_rnl	F170M	110.9653305	0.05916446957	0.03558	68.0765	43.0376	42.88	0.0177	50
2MASSIJ0652307+471034										
	n8yj14dmq_rnl	F110W	2795.62822	4.645308105	1.74732	2641.9167	201.9362	103.21	0.0424	30
	n8yj14dlq_rnl	F170M	3492.648242	1.862262807	0.47744	1852.8922	1660.293	3.44	0.0155	650
	n8yj14dnq_rnl	F170M	3536.849843	1.652763413	0.5921	3115.3616	486.0705	162.67	0.037	200

I determined the likelihood of a system being a binary in the following way. First I had to look at how the flux was distributed between the primary binary flux and the secondary binary flux. If the flux was distributed to which enough flux was in the secondary flux that it could be another brown dwarf. For example for SDSSJ020742.48+000056.2 the flux is approximately 50-50 between the primary and binary flux, so this points to a binary system. On the other hand, SSSPMJ0124-4240 has a flux distribution of 97-3, which would not be a binary system.

The next step was to check the separation angle discussed in equation 1.4. Most cameras including NIC 1 are not accurate for separation angles much lower than 0.02. So for each rnl.fits image for each object I checked the separation angle that the code outputted and if it was below 0.02 I flagged that object as inconclusive or not a binary candidate depending on the next step.

For the final step in determining binary likelihood I determined the error on the binary flux fit by using the PSF histograms. If these plus or minus errors were reasonable when compared to the total flux then I flagged this object as a likely binary candidate. For example, for 2MASSIJ0213288+444445 there was a total flux of 2602 and 2620 for the two rnl.fits files and an error of plus or minus 210 and 170. This is a flux for about 2610 ∓ 185 which is well within reasonable errors.

4.1.2 Limitations of the Code and Future Improvements

Beus et al. 2020 identified some limitations of the code in his conclusion section as well, but I have found some other inconsistencies that will need to be corrected in the future.

One issue that I found is that sometimes the primary flux and the secondary flux get switched. This means that sometimes the secondary flux has more flux than the primary, which

by definition would make it the primary. This did not affect the accuracy of the overall distribution of the flux but sometimes it would mislabel the fluxes.

Another inconsistency was that if there were more than one object in the image frame the code would not know which object to focus on. This did not happen often but it is something that may need to be addressed in later versions of this code.

Finally, another issue found when running this code was that the width of the PSF histograms was often much too wide to be able to have conclusive results. For example, 2MASSIJ0439010-235308 had a flux that peaked at 1308 and 1318 but had a flux agreement of $\mp$ 1050 and 900. These large errors basically invalidate the results taken from their PSF histograms. Beus et al. mentions that he believes that the code should be run more than the 100 times it currently runs during the monte carlo simulation. The main problem with this is that the code would take much longer to run, but it would also make the histograms much more accurate.

4.2 Conclusion and Future Work

This code is fairly effective in identifying brown dwarf binary candidates, but the mentioned limitations in this thesis as well as Beus et al. need to be implemented before I can confidently trust its results.

Beus et al. 2020 mentions various improvements that could be made to the code to expand its utility, including making it compatible with other HST instruments. After doing my research I would not recommend doing this until the previously mentioned issues are fixed, so this would be the bottom of my priority list in fixing the code. In addition to fixing the inconsistencies in the code, this code needs to be further optimized in order to support some of

these improvements. Some of the inconsistencies would be fixed if the Monte Carlo simulation was run many more times, but the code needs to be updated to be able to run many more times in a reasonable amount of time. The code now takes about 5 minutes to run for each rnl.fits image, so if we increased the monte carlo simulation by an order of magnitude then it would take around an hour for each rnl.fits to be processed by the current code.

I support the idea mentioned in Beus et al. that the code should automate the data preparation and reduction process. By automating the Tiny Tim process, pedsky, and rnlincor processes the use of this code is made much easier and the time needed to process each object would be reduced tenfold.

Bibliography

Beus, L. M., Miles, M., & Stephens, D. 2020, presented at the 235th American Astronomical Society Meeting,

Burgasser, A. J. 2008, Proceedings of the International Astronomical Union, 4(S258), 317

Burgasser, A. J., Cruz, K. L., Cushing, M., Gelino, C. R., Looper, D. L., Faherty, J. K., Kirkpatrick, J. D., & Reid, I. N. 2010, Astrophysical Journal, 710, 1142

Burgasser, A. J., Geballe, T. R., Leggett, S. K., Kirkpatrick, J. D., & Golimowski, D. A. 2006, The Astrophysical Journal, 637, 1067

Burgasser, A. J., et al. 2000, From Giant Planets to Cool Stars ASP Conference Series, 212, 65

Cushing, M. C., et al. 2011, The Astrophysical Journal, 743(1), 50

Eberhard, J. 2020, (Unpublished Senior Thesis) Brigham Young University

Gagliuffi, D. C. B., et al. 2014, The Astrophysical Journal, 794, 143

Golimowski, D. A., Burrows, C. J., Kulkarni, S. R., Oppenheimer, B. R., & Brukardt, R. A. 1998, The Astronomical Journal, 115, 2579

Kirkpatrick, J. D., et al. 1999, The Astrophysical Journal, 519(2), 802

Krist, J., & Hook, R. 2004,

https://www.stsci.edu/hst/instrumentation/focus-and-pointing/focus/tinytim-hst-psf-modeling

Kumar, S. 1963, The Astrophysical Journal, 137, 1121

Marley, & Leggett. 2009, Procedures of Astrophysics in the Next Decade, 101

Matt, K. 2017, Bachelors Thesis (Brigham Young University)

Padoan, P., & Norlund, A. 2002, The Astrophysical Journal, 576(2), 870

Rebolo, R., Osario, M., & Martin, E. 1995, Nature, 377, 129

Reipurth, B., & Clarke, C. 2001, The Astronomical Journal, 122(1), 432

Salway, E. 2015, Bachelors Thesis (Brigham Young University)

Stephens, D. C., & Noll, K. 2006, The Astronomical Journal, 131, 1142

Thies, I., Pflamm-Altenburg, J., Kroupa, P., & Marks, M. 2015, The Astrophysical Journal, 800(1), 72

Turner, S. 2020, Bachelors Thesis (Brigham Young University)

Whitworth, A. P., & Zinnecker, H. 2004, Astronomy Astrophysics, 427(1), 299