# Star Track

Sariah Cassidy, Keith Madsen, Clint Wilkinson
Electrical and Computer Engineering
University of Utah

***Abstract*—A tracking system was designed and built for a telescope without obstructing the view from the eyepiece. A webcam was connected to an off-axis adapter to prevent obstructing the users view. Motors were used to control the telescope based off of the images taken from the camera.**

## I. Introduction

Telescopes bring the shiny celestial objects of the night sky closer to observers on earth. However, amateur astronomers are snapped back to earth by the frustration of having to adjust the telescope every few minutes to center the image again. Matters worsen when the image is completely lost and the equipment must be completely realigned in the dark.

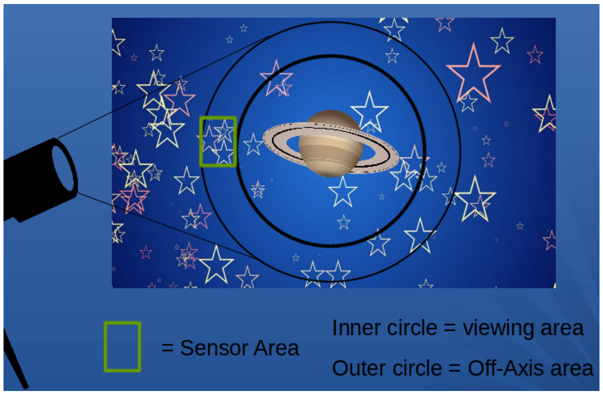


Fig. 1. Off-axis guidance concept.

This work describes the design for an off-axis guidance system for tracking the stars. The telescope takes in a larger area of sky than the eyepiece utilizes. In Fig. 1, the outer ring is the whole area that is captured by the telescope. The smaller ring is the portion used by the eye piece for observing. The area between the large ring and smaller ring is called off-axis and using any area in this boundary won't obstruct the view of the observer. This portion is what is used to track the images. Perpendicular to the eyepiece will be attached a small prism angled to snatch a piece of the unused image and reflect it out to a camera. This t-intersection can be seen in Fig. 2 with the prism shown in Fig. 3. The camera will be used to capture that image and send it to a Fast Fourier Transform (FFT). The FFT will convolve the tiny pinpoints of stars that may be found and send the information to a specially designed chip. The chip will decide which is the highest or brightest point and where it is located on the snatched image. If the image has moved the chip will send a signal to the microcontroller that the motors need to move and in which direction. The microcontroller then sends a direction signal and clock pulses to the motor causing it to correct the drift detected in the image.

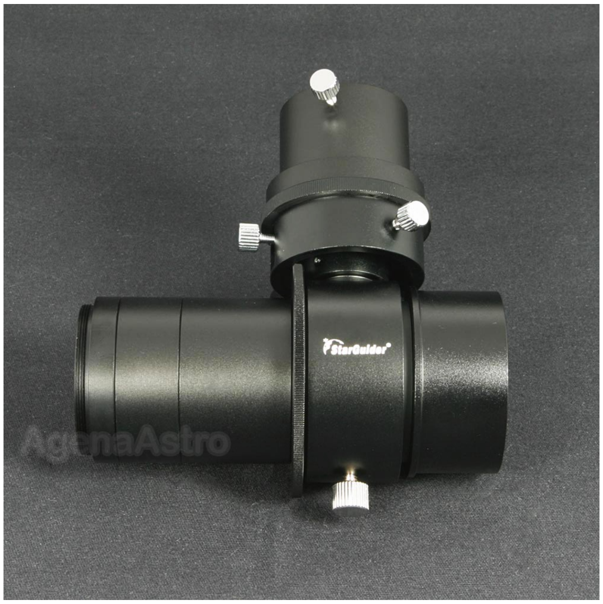


Fig. 2. Off-axis guidance eye piece [1].

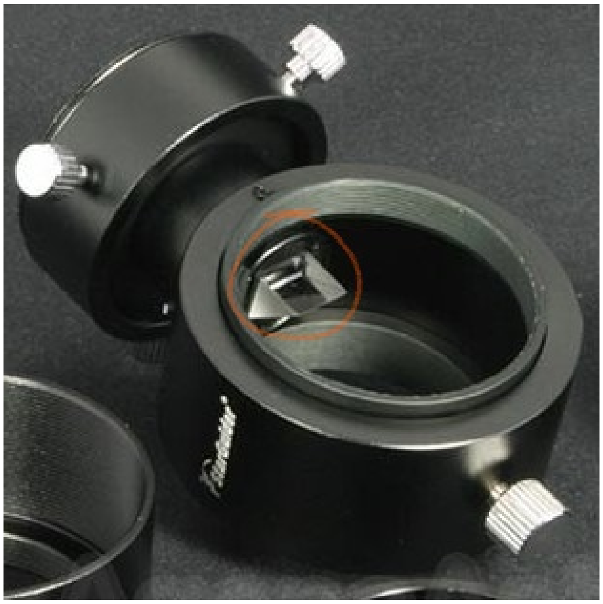


Fig. 3. Off-axis guidance mirror circled in orange [1].

This project involves a webcam and possibly other imaging devices to be connected to an off-axis viewing port of a telescope. The image taken is then manipulated by an image processing system which in this case is MATLAB based. The processed image is transmitted via serial communication to an embedded system designed to interface with a specially designed Very Large Scale Integration (VLSI) chip that converts the image into a motor direction command. The commands from the chip are sent to the motor control portion of the project and result in tracking control of the telescope.

Since the VLSI production was slow this year the project contains two versions. The one described in the above paragraph and a non VLSI version. The difference between the VLSI version and the non VLSI version is that the chip is replaced with a software equivalent. The software equivalent turns out to be faster due to removing the restriction of sending the peak correlation matrices over serial communications. The project is the same with or without the VLSI chip since the chips job is done by a software equivalent. The only noticeable difference comes in the update time between images.

## II. Equipment

The project is split into three parts:

1) Mechanical interface and design.
2) Telescope configuration and optics.
3) Image processing and camera interface.

### A. Telescope

An equatorial mount was used for the unique feature that when the mount is correctly aligned only one motor would be needed to track an object. A mount for the motor was found on Thingiverse.com and an extension for the mount was designed and used since the motor shaft was longer than the mount was designed for.

A Newtonian telescope was available for this project. Newtonian telescopes are considered to be reflective due to the light path that is taken through the optics as can be seen in Fig. 4. The telescope used here has a 4.5” reflection mirror.

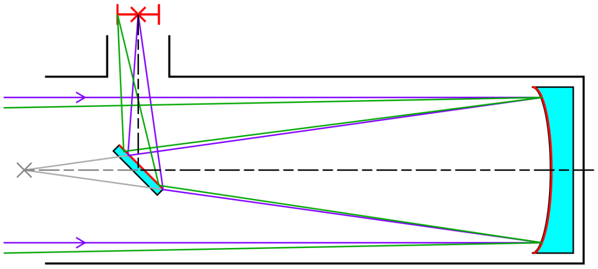


Fig. 4. The path of light through a Newtonian telescope [2]. The focal length is represented by the dotted line down the axis, reflected off the planar mirror and ending on the red x and line.

### B. Off-axis Guider

The off-axis guider had an input port of 2” while the telescopes output ports were 1.25”. Adapters were necessary to compensate for the difference and are sold widely for telescope applications.

Due to the added length of the guider to the eye piece, a Barlow lens was necessary to add length to the focal length of the reflection mirror of the telescope. The Barlow lens used is 3X. The adapter was redesigned to compensate for this addition. Compensation was made for the focal length differences between the outputs of the guider.

Another obstacle found in this system is that the prism on the off-axis guider is very small and pinches out a tiny piece of sky. The Barlow lens furthers the complication by scraping an edge out of what gets through to the prism. The webcam wasn’t sensitive enough to detect images dimmer than Saturn. An eye piece splitter was obtained that takes 50% of the whole image which gives our camera a better chance of capturing something from the mirror while allowing the other 50% to go to the eye piece.

### C. Optics

All parts were assembled and the telescope aligned properly. The Moon being the easiest night sky object to center in the scope, it was the test subject. The focal length of the light from the telescope was too short for the off-axis system to produce a clear image. A Barlow lens was introduced to the system to mitigate the issue. With the Barlow lens added onto the back end of the guider, an image was able to come into focus with the focuser at the shortest position. Further length adjustments can be made on the guider by removing extenders if needed.

### D. Barlow Lens 3X

During fitting and testing on the actual telescope it was found that a Barlow lens was necessary to be able to bring the image into focus for a 20 mm eyepiece extended out about 6”. A Barlow lens utilizes a convex lens to increase the magnification of the image according to the specifications of the lens. A 3X Barlow lens was used which magnified the image 3 times larger.

### E. Cameras

A USB webcam that puts out images in a 720p format was used. An adapter was designed to mount the camera without its lens to the perpendicular eye port of the guider or splitter.

The interface for the webcam was accomplished using MATLAB. The commands required to acquire images are included in the MATLAB image acquisition toolbox [3]. The images are taken into MATLAB and then via software the FFT and convolution are completed.

### F. Discovery Board Interface vs Raspberry Pi

Several STM32F07 Discovery Boards were available for this project. The image processing tasks were programmed through MATLAB along with getting the serial communications between MATLAB and the embedded system working. The Discovery board was used in place of the VLSI chip since the chip will not be fabricated in time for this projects completion date.

The STM boards do not have the capacity to process the images taken in by the webcam. MATLAB was installed on a laptop which ran the image processing information via

USART cable to the STM board. From there if the image was found out of alignment enough, signals were sent to the motor determining direction, speed, and duration. However, this setup was bulky and concerns were raised about the telescope hitting the table containing all the external equipment.

A reasonably inexpensive solution was devised using a Raspberry Pi running Raspbian. The Raspberry Pi is a card sized linux machine that is powerful enough to run image processing, with the ability to interact with peripherals through GPIO pins. The Raspberry Pi configuration allowed the system to be miniaturized and the process time significantly reduced. Webcams specifically designed for the Raspberry Pi were procured as well. The whole Raspberry Pi setup cost less than $100. Raspberry Pi case and mount to the telescope were 3D printed using designs from Thingiverse.com.

A few Raspberry Pi camera modules were acquired to see if better resolution could be accomplished with better technology. The camera modules were a bit more expensive than the webcam.

A Python image processing program was found that tracks a center rectangle in the camera image and puts out a value in the variable dx and dy [4]. Using just the dx variable, code was written and inserted to send enable, direction and clock signals to the motor driver when the tracked center moved within a determined tolerance.

### G. Chip

The VLSI chip was not fabricated in time to be included in this project. Another work will address testing of the chip.

### H. Motors

The driving force behind the software image correction are the motors. As such, the requirement to was to manipulate a 10 to 20 lb. telescope on a rotational axis about the X-Y dimensions. These manipulations are short precise adjustments when in tracking mode. With small motor movements about ever 30 seconds, the Gear Ratio 4:1 Planetary Gearbox Nema 17 Geared Stepper Motor 23HS22-2804S-PG4 won't overheat.

Stepper motors require a motor driving circuit to operate. This circuit requires a power supply to operate the motor, a direction input to determine which way current is applied to the motor, and a step/clock input to regulate the power delivered to the motor. Stepper motors do not operate by receiving a constant DC supply. The driver pulses the motor on a periodic interval. A higher frequency interval will cause the motor to spin faster under no load conditions, or deliver increased torque to its application when loaded. The Leadshine M542 Stepper Motor Driver 20-50 VDC with 1.0-4.2A capacity was chosen to for it's high precision for working in connection with the stepper-motor.

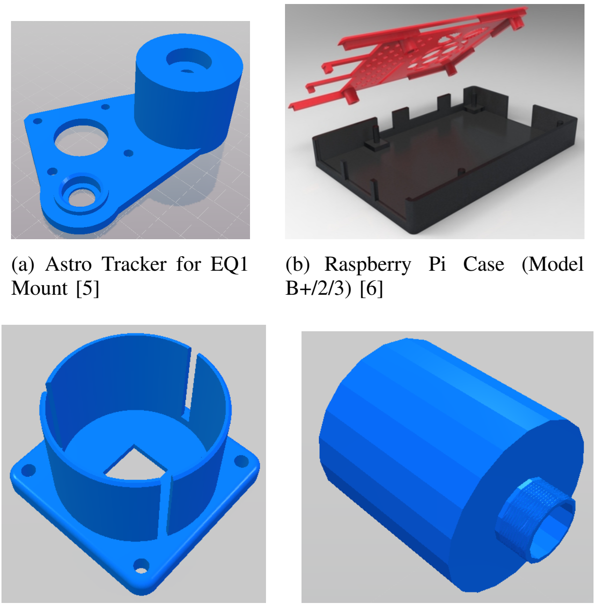
The power requirements for this system were met using a 24 volt battery. The discovery board required 5 volts to operate and so a 24 volt to 5 volt buck converter was used to meet that criteria. The battery also provided the power for the motors, but was directly hooked to the drivers.

### I. 3D printed parts

Specialized parts were printed with a LulzBot Taz 5 3D printer at the Marriot Library, with Ultimaker 2+ machines at Lassonde Studios, and the 3D printers available in the MEB. Guidance and assistance was provided by their respective staff members. The library requires payment for filament used and Lassonde Studios requires that print jobs be limited up to 10 hours and is operated by the first-come-first-serve rule.

The available filament was 3mm PLA plastic of various bright colors. Black would have been optimal but it was unavailable. However, 3D printed plastic is porous enough to hold paint and super glue. During testing it was found that the hot pink plastic reflected too much light so all printed adapters and parts in the light path were spray-painted black. Flat black is best but shiny black is sufficient as the application here isn't highly sensitive.

Some parts are remixes of things found on Thingiverse.com as shown in Fig. 5. All these parts were reconfigured to fit our needs.


(a) Astro Tracker for EQ1 Mount [5]

(b) Raspberry Pi Case (Model B+/2/3) [6]

(c) Raspberry Pi Camera Telescope Mount [7]

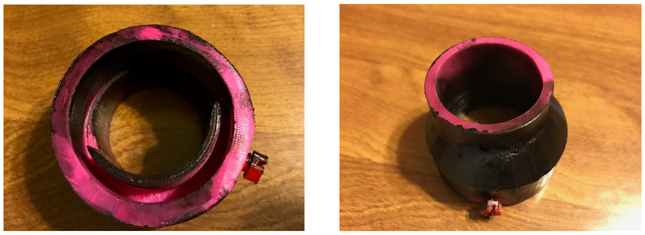
(d) Webcam Nosepiece Adapter for Telescope [8]

Fig. 5. Thingiverse original designs.

Using software on TinkerCAD.com, the adapters were designed to fit over the eye-port, guider, and eyepiece connections. The Barlow lens-to-guider adapter for the telescope to guider adapter will be refered as BGA from here on out. The guider to eyepiece adapter will be called GEA.

The BGA was designed with size and holding weight in mind. This system may be designed and able to support astrophotography in which case the BGA should be able to hold at least 10lbs. The thickness of the part is about 1/8" and was printed as a clamping piece.

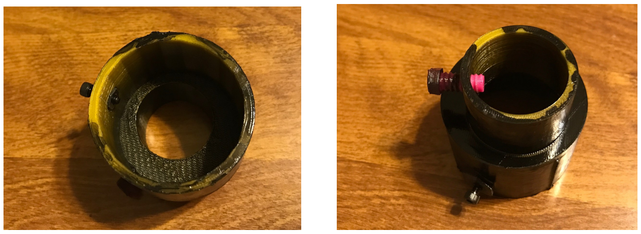
The GEA was designed with much lighter parameters. It's thickness is about 1/8" but won't hold a camera for astrophotography. It was printed in dense fill mode with light blue filament and was not painted as it doesn't effect light in



(a) BGA top view (b) BGA guider holding bolt view

Fig. 6. BGA painted black.

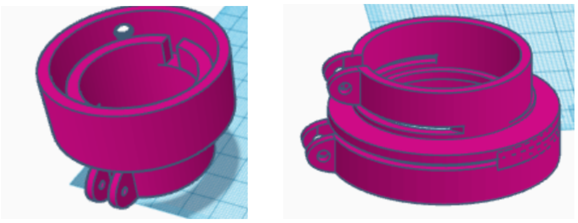
the path. The guider port is longer on this side with options to shorten the distance by removing an extender or two. The adapter worked best on the guider as long as only the short extension was installed.



(a) GEA top view (b) GEA guider holding bolts view

Fig. 7. GEA painted black.

Initial adapter designs were bulky and hard to manage. A tight fit was needed but the interpretation of the design by the printing software decreased the aperture size. Great amounts of filing was done to open the inner diameter wide enough for the desired fit. When the need for the Barlow lens was discovered, the weight support was mostly shifted onto the Barlow allowing for a more sleek design for the adapters. As can be seen in Fig. 8, a clamp style was used to allow for various sizes and eliminate the need for endless filing.



(a) Guider to eyepiece adapter. (b) Barlow lens to guider adapter.

Fig. 8. Adapters designed to connect optics to the telescope.

An Agena Deluxe Off-axis guider was installed on the eyeport of the telescope. A webcam nosepiece mount was downloaded from Thingiverse.com website [9]. The nosepiece allows for the webcam to be mounted to the off-axis port on the guider without its lens. It should be noted that it was required that the dimensions in the .stl file be changed to inches so the piece would be the correct size.

The print had to be high resolution as the tip of the nosepiece included fine threading as can be seen in Fig. 9. The first print done at Lassonde with a 0.8mm nozzle failed to keep the threading structure intact. This piece required professional help from the 3D printing experts at the Marriot Library on the University of Utah campus. The intricate threading of the nosepiece had to be set up on a Taz 5 machine with .40mm nozzle and Slic3r software was used to set up the precision needed on this part. Slic3r allows for a more intimate look at the nosepiece and what it would print with the printer settings uploaded. It was found that the structure for the threading needed to be slightly thicker and was adjusted accordingly. After printing, the points around the barrel of the nosepiece had to be filed slightly to fit the guider port.

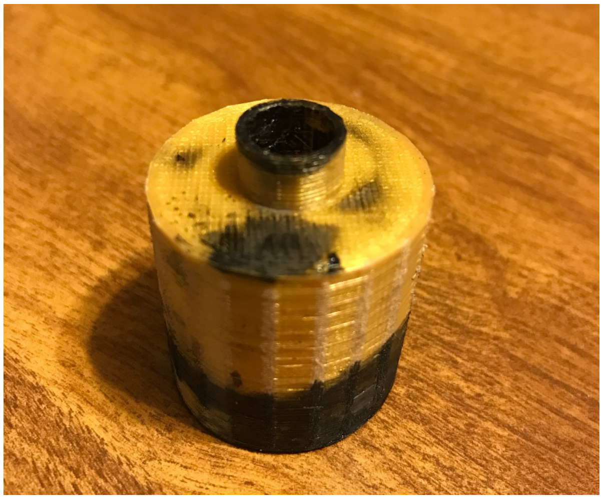


Fig. 9. Webcam nosepiece.

A motor mount was remixed and printed to mount the motor to the equatorial mount of the telescope as shown in Fig. 10.

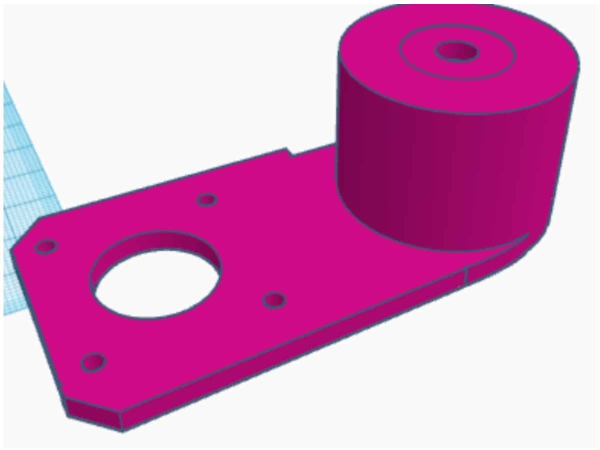


Fig. 10. Remixed motor mount.

A case for the Raspberry Pi was created using the two RPi designs from Thingiverse.com: the Raspberry Pi camera module to eye piece mount and the case for the Raspberry Pi. The case was too thin as downloaded and was doubled in thickness [6]. The top was reconfigured to bear the team logo. The mount was added to the front of the case to allow the camera to sit just above its connection port on the Raspberry

Pi [7].

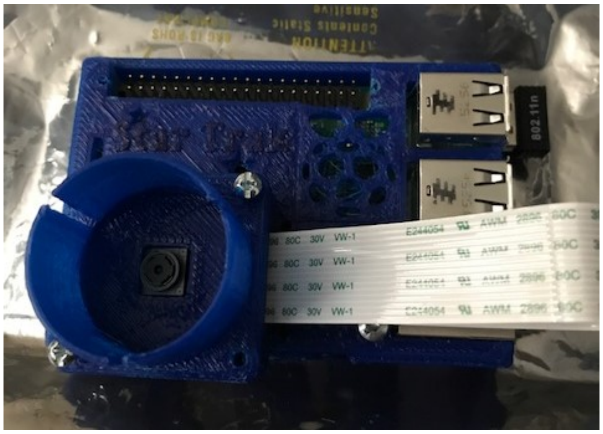


Fig. 11. RPi case and telescope camera mount combined.

*1) Raspberry Pi 2:* To improve upon the system, a Raspberry Pi 2 (RPi2) was obtained to make the system more compact, run independent of a console, and more affordable. Raspberry Pi computers natively run Python on the system. So the RPi2 was programmed with Python and OpenCV package libraries installed. OpenCV libraries contain the image processing protocols for taking an image and working convolution. It also contains code for motion detection which is useful for comparison against the VLSI chip system efficiency.

A Raspberry Pi camera module was also obtained to try to resolve the camera resolution of the webcam.

## III. Testing

Testing was accomplished in phases. The first set of tests completed were the camera interface and image processing. Off scope, the camera and software worked. When mounted to the scope problems arose. The camera was not sensitive enough to find even bright stars. Only near earth objects were detectable by the webcam. This turned out to be a limitation of the webcam and can be fixed by purchasing a more sensitive camera. The next set of testing was the embedded programming to test the VLSI chip or simulating the chip. The convolution method worked, but was found to be unreliable if the image shifted too far in between image updates. Convolution has to be used for the VLSI chip due to the chip design. Correlation matrices are much larger in size and are not compatible with the chips interface. Testing showed that correlation was much more successful at tracking than convolution, but both performed well.

The off-axis optics did not work out with the camera. The amount of light from a tiny little prism would never be spotted by the camera that couldn't detect it straight from the telescope. One solution was to use the splitter which divides the incoming light in half. It obstructs the view more but allows more light into the webcam. This method was tested and proved to be successful, but camera limitations were still a problem on low light objects.

Motor control was calibrated during testing. Only one motor was needed as the equitorial mount that was used was designed to only need one axis of rotation to follow an object moving in the same general speed and direction as the earth's rotation. The equitorial mount requires alignment with the north star in order to properly remove the z axis drift. Once aligned, the mount proved reliable in removing the need for a z axis motor.

### *A. Image Processing*

The camera interface was tested by acquiring images from the webcam via MATLAB and displaying the images on the screen of a tablet Personal Computer (PC). Two algorithms were written into MATLAB and used to find image movements. One algorithm uses convolution as the basis of operation [10] and the other uses normalized correlation.

The image processing algorithms were tested by using an image of a red blob on a black background as in Fig. 12. Multiple copies of the image were made, but with the blobs at different locations. The algorithms were used to calculate where the blob had moved and to report that location. Since the new blob position was always known the algorithms had to report the new location correctly. Both algorithms performed flawlessly in these test conditions.

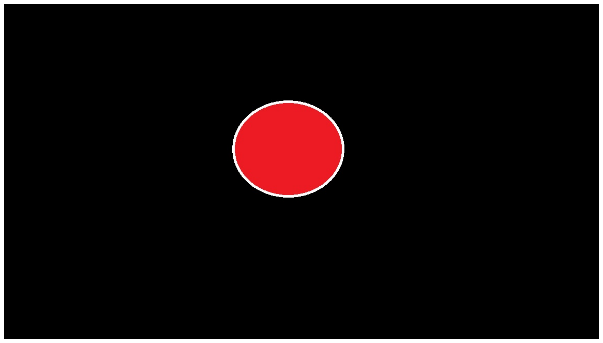


Fig. 12. Moving Blob Image

The second set of tests was done by taking an image of a room and then slightly moving the camera and snapping another image. The algorithms had to report the camera movement in order to be considered a successful test. As long as the camera movement was small these test performed quite well.

The final test was mounting the camera to the telescope and testing its ability to correctly determine shifts. During this testing it became apparent that convolution was not as effective as correlation. Convolution tended to get confused and report false directions if the image drifted at all in the z direction. The equitorial mount minimized this, but it still caused issues at times with dimmer objects. The correlation algortihm performed correctly with dim objects as well as with any z axis shifts. It was because of these issues that correlation was chosen over convolution as the preffered method of image processing.

The VLSI chip will be tested by attaching it to an embedded system programmed specifically for transmission of peak correlation matrices. The embedded system will send the matrices to the chip and then receive motor directions back via external interrupts. The VLSI chip will be synched to the embedded system via a shared clock.

MATLAB serial communications were tested by having a known message sent out to the board and then echoing it back to MATLAB. Once communications were established a simple transfer protocol was developed. The first letters of the words needed were used to abbreviate the whole word. Up became u, down became d, quit became q, left became l and finally right became r. In this manner the non VLSI version communicated directions. The VLSI version requires an entire matrix at a resolution of 800X600 to be sent. The communication for this was done by transmitting unsigned 8 bit data. One number per pixel.

Motor testing was accomplished by hooking the drivers to the embedded board. Putty was then used to send movement commands to the board which caused the motors to move. In this manner the motors could be given direction pulses or tested with continuous movements by holding the direction button down. The motors responded properly to directional commands indicating that the interface was written correctly.

The off-axis optics were tested by connecting them to a telescope and then hooking the camera to the guider. Problems with the setup were rapidly apparent. The camera was not sensitive enough to pick up light from the tiny prism in the guider unless the guider was picking up high intensity light from a bright source. The camera did successfully track the moon in off-axis mode, but the image produced by the camera was not focused. It turned out that it was tracking by aligning unfocused images. This worked, but it was readily apparent that in order for off-axis guiding to be successful a much better camera would be required. The discovered limitation was caused by the camera and not the off axis equipment.

The final operating test was putting everything together. The equatorial mount was aligned to the north star. The moon was tracked, Saturn was tracked, and Jupiter was tracked. Attempts at tracking other celestial objects were not successful because the camera could not detect the faint light. The eyepiece produced images, but the camera could not detect them. It was at this point that a splitter was purchased to help mitigate the off-axis lighting problems. The splitter was able to allow the camera to see fainter objects than the off-axis guider allowed for. By using both adapters the ability of the project to track objects was increased.

The final tracking test performed was that of tracking the Sun. To track the Sun, no off-axis or splitters were used. The camera mounted directly to the telescope which was equipped with a solar filter. The project was able to successfully track the sun. The images from the camera in this setup were quite interesting. Sun spots could be clearly seen in the tracking images.

## IV. Baseline Requirements

The proposal laid out very specific requirements that this project was to meet in order to be called successful. The requirements were implemented as shown in Fig. 13. The requirements are listed below:

1) A camera will mount to a telescope and acquire images of the night sky.
2) The images will be mathematically processed to find convolution or correlation.
3) A motor directional signal will be sent out to drive motors.
4) Motors will be mounted and drive a telescope to maintain a desired image in the view port.

Every requirement was met. The project was able to get images, process them and produce motor control signals that allowed tracking of objects. Video proof of the requirements has been obtained and a live demonstration was also given.

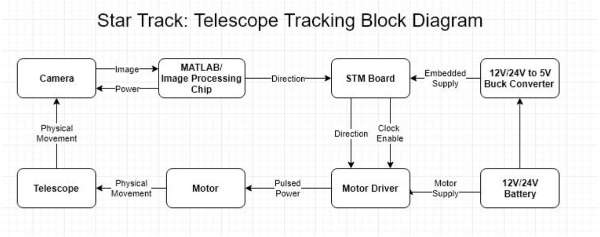


Fig. 13. Star Tracking Block Diagram

## V. Lessons Learned

Prototyping specific parts takes time, material, and ability to adapt to unforeseen circumstances. Many adapters and parts were designed, printed, redesigned, and reprinted.

Learning Python was a monumental task. Finally, merging an already existing image processing code with a motor command line and getting it to work with the system was priceless.

On occasion problems will develop that can be interpreted as project issues, but in reality are related to failed hardware. There were several occurrences of motor malfunction that were misinterpreted as project issues. After rechecking signals and testing everything in the loop it was determined that the motors were failing. Further analysis showed that the chosen motors had sensitive wiring that would break under minimal stresses.

## VI. Conclusion

The system solves a huge problem for amateur astronomers and backyard enthusiasts alike. Anyone who has ever used a telescope to view objects in the night sky will be giddy to know that a system like this exists.

The proposal requirements were met. Use of embedded systems knowledge, digital design, software programming and a cheap webcam resulted in an impressive control system. The normal view from the telescope eye piece was not compromised due to the use of off-axis optics and the optional splitter for lower light conditions. Viewing images from the night sky has become more fun and much less of a hassle.

## References

[1] A. AstroProducts, “Agena Deluxe Off-Axis Guider for Astrophotography,” 2017.

[2] Krishnavedala. commons.wikimedia.org/w/index.php?curid=34890843, Own work, CC BY-SA 4.0, for the Newtonian telescope image.

[3] MathWorks, “Acquire images from webcams.”

[4] C. Pageau, “Cam Track.” https://github.com/pageauc/cam-track.

[5] C. Queck, “Astro Tracker for EQ1 Mount.” Thing 1033336, www.thingiverse.com.

[6] A. Mary, “Raspberry Pi Case Model B+ 2/3.” Thing: 559858 www.thingiverse.com.

[7] S. Salt, “Raspberry Pi Camera Telescope Mount.” Thing 1812708, www.thingiverse.com.

[8] Y. Schefres, “Webcam Nosepiece Adapter for Telescope.” Thing 533599, Thingiverse.com.

[9] Y. Schefres, “Webcam Nosepiece Adapter for Telescope,” 2017. Thing:533599, www.thingiverse.com.

[10] S. de, “Finding the shift,” 2011. Thread 302016, Mathworks.com.