Object Spectrometer

Design and Fabrication of a Transient Object Spectrometer
Submitted To: Professor Nick Glumac
University of Illinois at Urbana-Champaign
ME 470 Fall 2019
Final Report

Prepared By:

Ayano Hiranaka	ayanoh2@illinois.edu	(567) 207-6931
Wali Rizvi	smrizvi2@illinois.edu	(847) 641-7067
Christopher Wong	cw3@illinois.edu	(925) 548-4472

Department of Mechanical Science and Engineering
University of Illinois at Urbana-Champaign

December 11, 2019

Table of Contents

1. Executive Summary

Sponsored by Professor Nick Glumac, the goal of the Object Spectrometer project was to develop a device that detects, tracks, and collects spectral information of a moving, light-emitting object in the night sky to aid in identification. Efforts have been made in the past to develop similar devices, but the insufficient software tools available back then caused challenges, preventing successful, highly accurate devices from being developed. However, today, with extensive strides in computer vision and easy access to high-performance computers, it is possible to use state-of-the-art algorithms to solve this problem.

The developed system prototype includes a camera used for tracking and a lens used to collect the target's light mounted onto a servo-controlled pan-tilt stage. The input camera image is processed to detect a target and the pan-tilt stage is controlled to follow the target. The light collected from the lens is converted into spectral data using an accompanying spectrometer. Due to the nature of the project, the primary focus was placed on developing the software for object detection and pan-tilt stage control.

The system utilizes three controllers to run the necessary software. The Python code running on the PC controller uses OpenCV, an open-source computer vision library, to detect an object from the camera input. The same program also implements a PID controller program to direct the pan-tilt stage to follow the target object. An Arduino is incorporated into the system to obtain the absolute position of the pan-tilt stage servo motors, which serve as an input to the PID controller. Finally, a Maestro Servo Controller Board is used to signal the pan-tilt stage to the desired position calculated by the PID controller.

Although the consistency of data collection decreases with increases in object speed, the final prototype is able to detect and track a target object with high enough accuracy to obtain spectral data for ideal conditions where a single target (with angular velocity up to 5 degrees/sec) is observed in a dark background. However, the system struggles when the background contains distractions that can be mistaken for a target. Approximately $900 out of the $1500 budget was spent over the course of the project for prototype fabrication.

2. Introduction and Problem Statement

2.1. Introduction

Figure 1. The left image shows the Hessdalen lights in Norway [1] while the right image shows an optical phenomenon observed in the Bay Area, California in December 2018 [2]

Extensive use of filming devices such as surveillance and personal cameras has increased the number of incidents where unusual atmospheric phenomena are captured. The continuous media coverage of such incidents is immediately followed by concerns and speculations by the public. A famous example of these incidents is the Hessdalen lights, shown in Figure 1, observed periodically in Norway since the 1930s. This light-emitting atmospheric phenomena with an unknown cause has attracted tourists to the area for sightings as well as given rise to multiple theories on the cause of the lights. However, identification of these light sources is not only of interest for the public, but also holds potential value in defense and research applications. Despite the high interest, a video recording, in most cases, is insufficient to determine the source or cause of the light.

One way to identify the objects is by investigating the spectrum of the light. Each light source has a unique spectrum based on its chemical composition, as shown in Figure 2. Collecting spectral information can, therefore, contribute to its identification process. However, due to the transient nature of the lights, their spectral data is difficult to obtain. An autonomous system that waits for a transient object, tracks the light, and collects its spectral data will be of high value in a variety of applications, including research, defense, as well as civilian use.

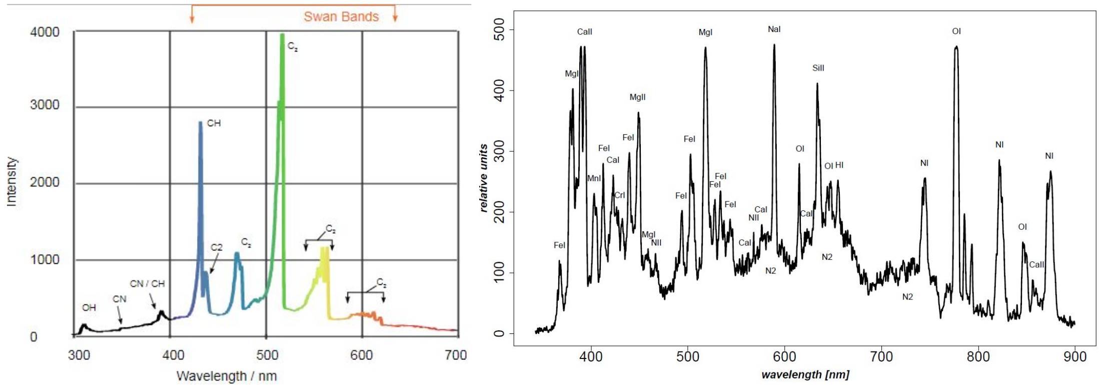


Figure 2. Example spectra of light-emitting phenomena. The left plot shows the spectra of a flame while the right plot is a spectrum of a meteorite.

2.2. Problem Statement

The goal of this project was to develop a prototype of a device that can detect and track observable objects or lights in the night sky, and obtain their spectra for further analysis. In real-life applications, a target object may exhibit inconsistent light-emission patterns or irregular motion. It is also possible that there are multiple objects either moving in unison or in disarray. Additionally, the environment may also contain distractions, such as the moon or stars, that can be mistaken for a target object. Due to the complexity of the problem, focus was directed towards developing a system that functions under ideal conditions. Henceforth, ideal conditions are defined to be a single light-emitting object present in a dark background absent of other objects with similar brightness to the target object. Under this ideal condition, the system must track a target with sufficient accuracy for the collected light to travel through the optical fibers into the spectrometer. Hardware components (excluding the PC controller) securely contained in a single package and a well-documented codebase are handed over to the sponsor at the conclusion of the project.

2.3 Literature Review

The Object Spectrometer project can be seen as an amalgamation of subprojects. These subprojects have been implemented before, but for different applications. Being aware of prior work not only ensures that verified and robust solutions are incorporated into the project, but also saves valuable time by not reinventing previously implemented solutions.

Variations of pan-tilt stages are usually used in surveillance cameras, making them active cameras. Objects that move across a wide area can be difficult to track using a stationary camera, so active cameras are useful for such applications. Researchers have previously created a robotic pan-tilt stage with two cameras that tracks and follows an object in 3D space [3]. Commercial products that track moving objects using a pan-tilt stage, such as the Real-Life Sentry Gun, are available on the market [4]. Another group in Japan has used a pan-tilt stage to keep a camera trained on an object by using background subtraction techniques [5].

The recent popularity of computer vision has produced many state-of-the-art algorithms that work on both recorded video and real-time feeds, with multiple techniques being developed for different applications. The book by Mitiche and Aggarwal [6] presents useful concepts in the analysis of image motion. The chapters on motion detection and tracking, presenting multiple algorithms for specific uses were of particular relevance to this project. Tracking a moving target using a moving camera, as in our case, poses new challenges as algorithms such as background subtraction do not work well with moving backgrounds. However, such an implementation has important applications like tracking pedestrians from a moving vehicle [8] or tracking moving objects in a video stream from a moving airborne platform [9]. A promising method includes compensating camera motion using Euler-Lagrange descent equations, which is shown to work on real image sequences [7].

Spectroscopy, the end goal of our project, is a fundamental tool in investigating objects at the molecular or atomic level. It is an important tool used in optical astronomy allowing astronomers to learn a broad range of attributes of an object [10]. Astronomers have also used spectroscopic observations in distinguishing incoming meteorite material from the multiple types that exist [11]. Because different elements have distinct spectra, it is possible to identify the makeup of an object, which can aid in making inferences on object classification.

3. Solution Procedure

With the information gained from the literature review, the team then defined system specifications, selecting both hardware components and software tools. The following sections outline the system specifications and the overall setup of the system. The functions, selection processes, and the outcomes of important hardware and software components are also discussed in detail.

3.1 System Specifications

The lens and the tracking camera were selected first as they have the greatest influences on the selection of the remaining components. In order to allow for real-time tracking, the camera requires both a high resolution and a framerate of at least 30fps. The light-collecting lens must have a large aperture to collect the maximum amount of light, while also having a focal distance long enough to focus on a distant object. Once these two components were selected, their specifications informed the remaining system specifications.

When making hardware selections for the final design, one of the most important aspects to consider was the required accuracy of tracking for the system. In order for spectral data to be obtained, the light from the target object must be accurately centered in the lens so that the light is focused into the optical fiber attached to the back of the lens. Equation (1), the plate scale equation, was used to calculate the required tracking accuracy from lens focal length f, object angular separation θ in arcseconds, and object radius on the focal plane s.

$$\frac{206265}{f} = \frac{\theta}{s} \tag{1}$$

Using equation (1), the diameter of the optical fiber bundle and the tracking camera pixel distances are related to determine the required tracking accuracy.

Finally, with the required precision known, the pan-tilt stage must have a high control resolution to keep the target object within this light-collecting region, while also being capable of carrying the weight of the lens and camera. The control program and controller for the stage must also be able to provide commands with enough precision to keep the stage on target.

Due to the research-oriented nature of the project, there were no relevant standards the prototype must comply with. Similarly, constraints relating to public health, safety and welfare, as well as global, cultural, societal, environmental, and economic factors were not of high concern in the Object Spectrometer project.

3.2 Proposed Solution

In order to identify, track, and collect spectral data from objects in the night sky, the following components have been combined into a single portable package: (1) A tracking camera, (2) a computer (PC controller), (3) an Arduino and motor encoders, (6) Maestro Board (servo controller), (5) a pan-tilt stage, (6) a lens, and (7) a spectrometer and an optical fiber bundle. The spectrometer and the optical fiber were provided by the sponsor, and the team members' personal laptops were used as the PC controller during development and testing.

The process through which all the components come together is quite intricate. First, the tracking camera monitors the environment and this video stream is continuously sent to the computer. Each frame of the video is then processed by the object detection software which identifies a target object within the video frame and continuously outputs its center coordinates. At the same time, the Arduino and encoders provide instantaneous pan-tilt servo positions to the computer. The PID controller running on the computer takes the object coordinates from the detection software and the servo positions from the Arduino to calculate the target servo positions to keep the target object centered in the camera frame. These target positions are routed through the Maestro controller, which moves the stage. The process repeats with each incoming frame from the camera feed, allowing for the target position to be adjusted at a rate of up to 30 updates per second. As the system follows the object, light is collected through the lens and focused into the optical fiber bundle attached to the back of the lens. The object's light then travels through the optical fibers into a spectrometer. Additionally, when tracking is in progress, the computer records the image seen by the CMOS camera built-in to the spectrometer as a video file. This video file can be later used to extract the tracked object's spectral data. The system data flow is illustrated in Figure 5.

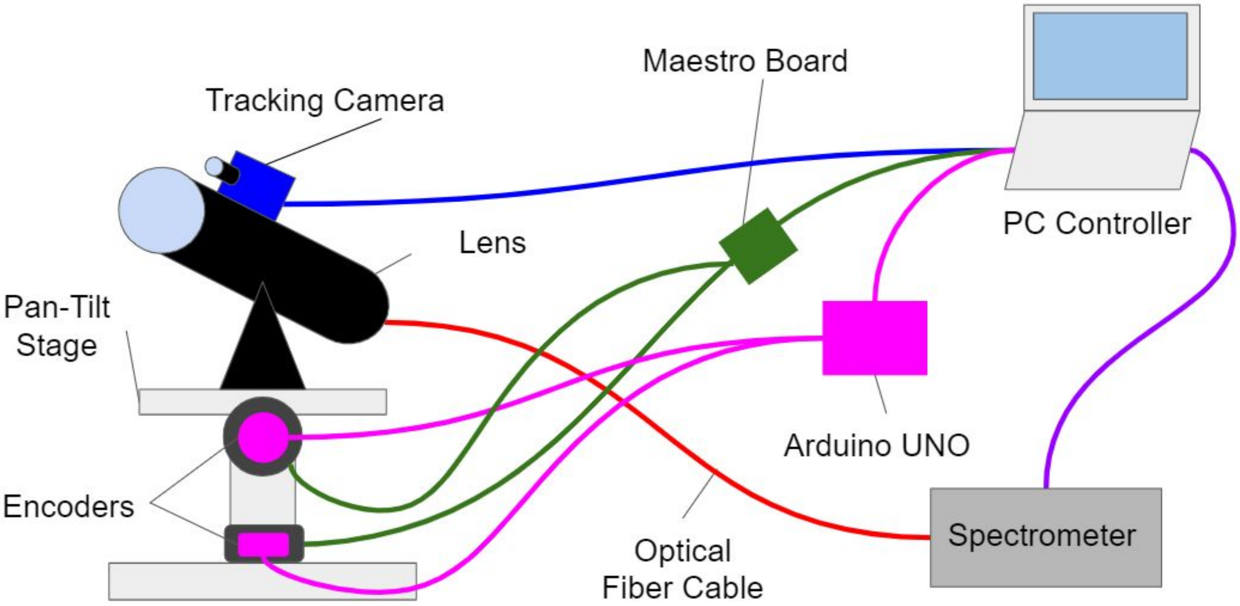


Figure 3. Final System Setup Diagram

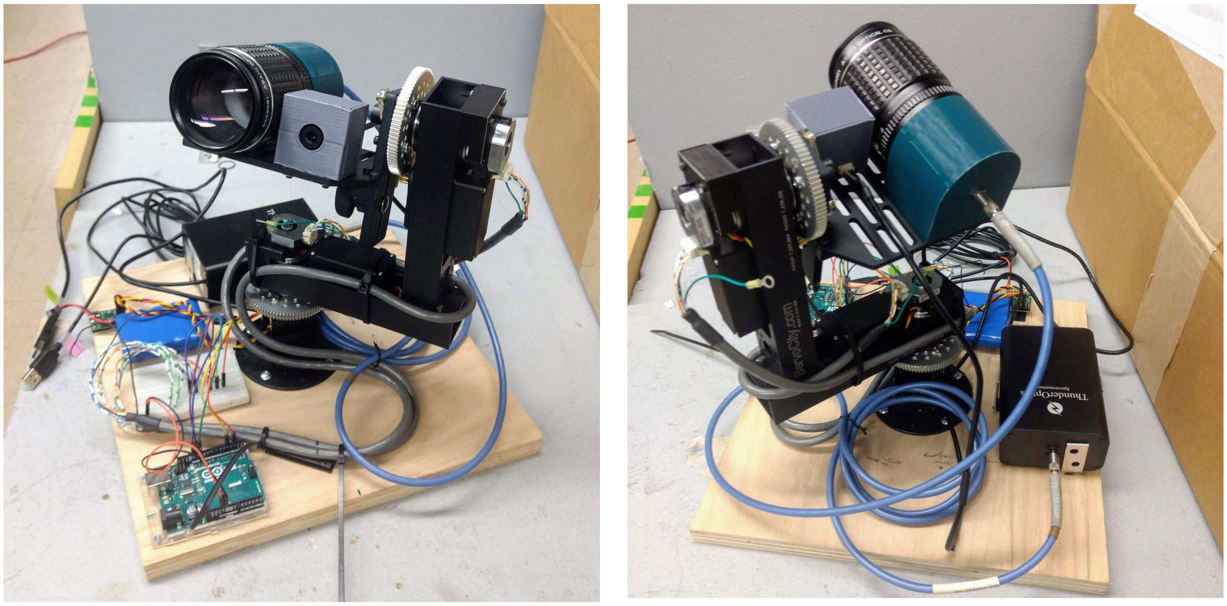


Figure 4. Final System Setup Photo

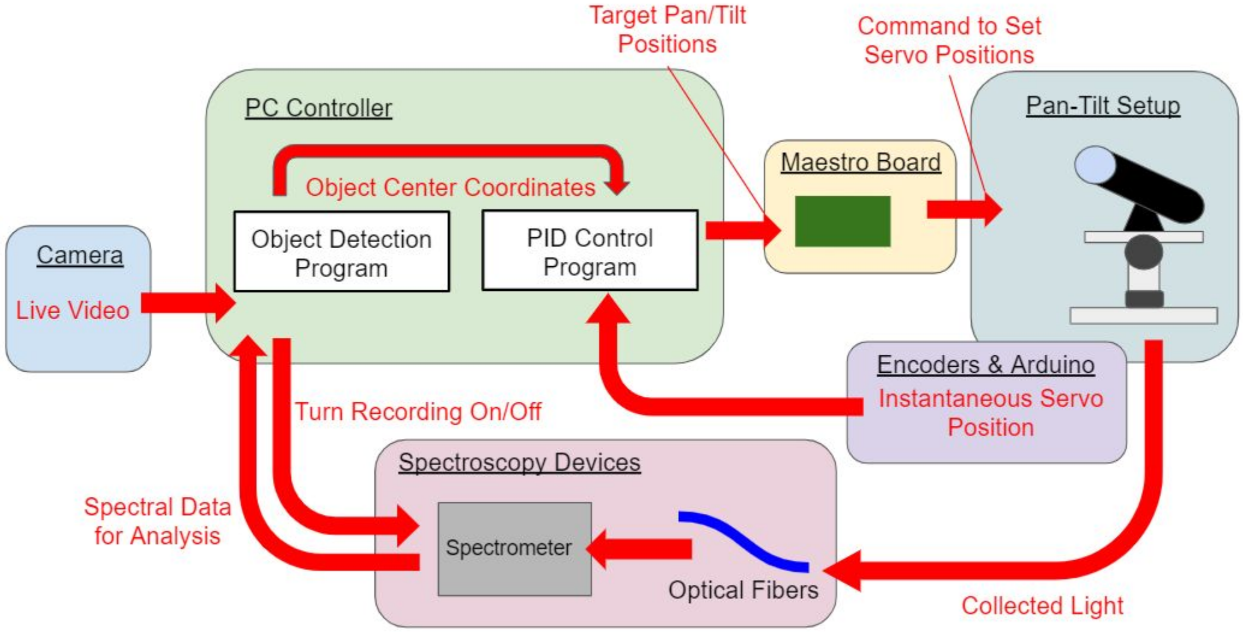


Figure 5. Flow of Data in the Final System

3.3 Hardware Components

Selection of hardware components was one of the major aspects of the project. The following sections discuss the system's hardware components, as well as the reasoning behind these selections.

3.3.1 Lens

In order to gather the desired data, the lens must be able to focus light from the target object into the fiber bundle connected to the spectrometer. When selecting the lens, the primary consideration was to minimize the f-number, or the ratio of the focal distance to the aperture diameter. A larger focal distance narrows the field of view of the lens, allowing for the target object to appear larger in the frame. On the other hand, a larger aperture allows more light to enter the lens. Therefore, it is important to find a reasonable balance between these two values when comparing f-numbers. The team originally targeted an f-number of 2.8 and a focal distance of 135 mm during lens selection in order to remain within budget. The Takumar 135 mm f/2.5 bayonet lens, which exceeds the target specifications while still remaining under budget, was selected for use in the final prototype. In the final prototype, the lens is mounted using a single

3D printed part. The use of a standardized Pentax K type mount allows for the lens to be easily replaced while also retaining the ability to manually focus the system. Additionally, the SMA 905 connector on the back face of the mount centers the optical fiber bundle on the lens to ensure consistent data collection while centered on a target.

Figure 6. Takumar 135mm f2.5 Bayonet Lens

3.3.2 Spectrometer and Optical Fiber Bundle

The spectrometer's role in the system is to decompose the light collected from a target object into spectral data. A ThunderOptics Mini USB Spectrometer was provided by the sponsor for this purpose. The provided spectrometer consists of a USB webcam behind a diffraction grating, with light entering through an attached 600 µm optical fiber bundle. The video feed from the spectrometer's camera can be processed using Theremino Spectrometer, the official software package for ThunderOptics spectrometers, which converts the band of colors in the webcam frame into usable spectral data. Additionally, since the data collection is done by a USB webcam, the incoming feed can be manipulated just like any other webcam feed or saved for later use. Due to Theremino Spectrometer's lack of options for interacting with other programs, the final prototype creates a time stamped video file every time a new object is detected and outputs the recording of the spectrometer's raw video feed.

Figure 7. Thunder Optics Mini USB Spectrometer

3.3.3 Camera

Since the object spectrometer will primarily be used at night, one of the main concerns during camera selection was performance in low-light environments. Additionally, since the lens and guide camera are mounted together on the pan-tilt stage, the camera's physical footprint had to be small enough to allow for a secure mount without jeopardizing the stability of the lens. For these reasons, the Spinel UC20MPG L60 was chosen for use in the final prototype. The Spinel's 2.9 µm pixel size allows it to take high-resolution images at 30 fps with a minimum ambient light level of only 0.002 lux for fast and effective object identification. The Spinel also came with a 6 mm lens, eliminating the need for an additional lens for the tracking camera. Moreover, the wide operating temperature range of -20 °C to 75 °C ensures smooth operation in any weather conditions. Finally, the 12 g weight and 38 mm square footprint allow the camera to be mounted easily to the pan-tilt stage. With the camera selection finalized, the required accuracy of the object tracking program can be determined. Since the tracking camera and lens are both watching the same object and the angular separation of the object is unknown, the plate scale equation (Equation 1) can be solved for θ. The θ values from the camera and the lens can then be equated, resulting in the following equation relating the lens specifications to those of the camera.

$$\frac{s_{lens}}{f_{lens}} = \frac{s_{camera}}{f_{camera}} \tag{2}$$

Substituting the diameter of the optical fiber bundle 600 µm for s_{lens}, 135 mm for f_{lens}, and 6 mm for f_{camera}, in equation 2, s_{camera}, is calculated to be 26.6 µm, or 9 pixels based on the Spinel's pixel size. This 9-pixel diameter circle at the center of the frame corresponds to the area within the tracking camera frame in which the target object must be maintained such that the target light is focused into the optical fiber bundle.

Figure 8. The UC20MPG L60 camera

3.3.4 Pan-Tilt Stage

An off-the-shelf pan-tilt stage from ServoCity, the PT785-S, was selected for the project. The stage includes preinstalled HiTec HS-785HB Servos for both the pan and tilt axis. This particular pan-tilt stage was chosen for its large bedspace and maximum payload capacity of 6 lbs. Its bed provides more than enough room for secure hardware mounts as well as a multitude of possible connection points. Additionally, each axis is geared down with a 1:7 gear ratio for increased torque and precision during movement. Due to the tracking accuracy requirement, the PT785-S stage's gear ratio was one of the most important deciding factors.

Figure 9. The PT785-S Pan-Tilt stage

3.3.5 Encoders

The primary weakness of the HS-785HB servos on the pan-tilt stage is the lack of instantaneous feedback for positional adjustments. With stock hardware, querying the position of the servos will only return their last target position, regardless of whether or not the motor has actually reached that position. By adding motor encoders to each of the pan-tilt stage servos, instantaneous servo motor positions become available, allowing for more precise adjustments in the stage position. When selecting an encoder, the team opted for an absolute encoder that can be programmed to have a direct mapping to the angular position of the servo. An incremental encoder would have required calibration before startup each time. Absolute encoders, on the other hand, remember their positions after they have been calibrated once. CUI Devices' AMT22B encoders were selected for this purpose. The CUI encoders are single rotation and programmable, allowing for a resolution of up to 14 bits, which translates roughly to 0.022º per increment. As shown in figure 10, the encoders were mounted on each axis of the stage using 3D printed, laser cut, and stock components. Additionally, as single rotation encoders, the position value resets with every rotation. Mounting the encoders on the stage axes keeps the movement within the one-rotation range, maintaining the linear relationship between encoder and servo position. Since the encoders use the Serial Peripheral Interface (SPI) protocol to communicate, a microcontroller with SPI receiving capability is required to read any useful information. An Arduino UNO board was added to the system to establish SPI communication with the encoder. The function of the Arduino in the prototype is detailed in the following sections.

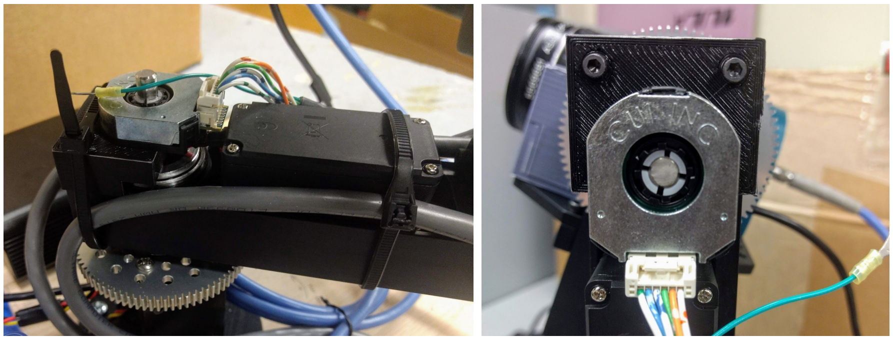


Figure 10. Left: Encoder mounted on the pan axis, Right: Encoder mounted on the tilt axis

3.3.6 Microcontroller

Since the pan-tilt servos cannot be directly controlled by the computer, an additional piece of hardware must be used to facilitate communications. Initially, a Maestro board was selected for PC-servo communications. The Maestro's small size allowed for extremely easy mounting, and its quarter-microsecond resolution and Python support allowed for precise stage control and easy integration into the system. However, the Maestro's lack of input pins for encoders and no protocol for SPI communication led the team to search for alternatives, ultimately landing on an Arduino UNO. The Arduino's small size, versatility, and power consumption made it ideal for the team's needs, and the abundance of support and reference materials available online greatly simplified the transition process from the Maestro to the Arduino. CUI Devices also provides extensive documentation on integrating their modular encoders into an Arduino system, further simplifying the implementation of the Arduino controller.

However, the primary limitation of the Arduino is its servo control library. Since the required precision is much greater than the standard `servo.write(deg)` function, the team instead opted to directly control the PWM signal using `servo.writeMicroseconds(ms).` The resolution of the Arduino's PWM control is limited to increments of full microseconds, so a range of 800 ms corresponds to a 180° sweep of the pan-tilt stage. Based on these values, 1 ms was found to correspond to 0.225° or 810 arcsec. Substituting this angle value and the camera specifications into equation 1 results in a 1 microsecond pulse corresponding to a 23.5 μm distance on the sensor, or an 8 pixel distance in the tracking camera image. Therefore, a 2 μs error in the PWM control could result in passing over the entirety of the fiber bundle diameter. Comparatively, the 0.25 μs resolution of the Maestro board translates to 0.056º stage movement per step after the 7:1 gearing of the stage. This moves the frame of the guide camera by roughly 2 pixels per step, making precise tracking much more feasible. Based on these calculations, the team decided to move back to the Maestro board for more precise servo control, while also retaining the Arduino for encoder communication.

Figure 11. Maestro Servo Controller Board (left) and Arduino UNO (right)

3.4 Software Components

The software component of the project can be divided into three main parts: (1) Object detection, (2) PID control, and (3) encoder data incorporation and servo motor control. The first two components, written in Python, are combined into a single program and run on the PC controller. The third component is divided among the PC controller, Arduino, and the Maestro Board. The following sections discuss each software component in further detail.

3.4.1 Object Detection Program

The role of the object detection program is to identify an object from the live video feed from the tracking camera and continuously return the center coordinates of the objects for use by the PID controller. The developed prototype uses Python's OpenCV library. While other options available in the market, such as TrackR and a control program by Real-Life Sentry Guns, were considered, Python OpenCV was selected due to its ability to take live video stream as input, high customizability, and abundance of support resources.

The detection program processes each incoming frame in the following way. First, a user-defined brightness threshold is applied to convert the image into a binary image. All pixels with brightness values within the selected range become ones, while all pixels with brightness values outside of the range become zeroes. The result is a black-and-white image showing the target object in white and all other areas in black. However, this binary image is generally too noisy to be used as-is, with small white particles appearing as a result of both hardware and environmental factors. To combat this, a series of erosions and dilations are applied to the frame to eliminate any noise. Finally, SimpleBlobDetector, an OpenCV class, uses the resulting binary image to determine the center coordinate of the detected object (SimpleBlobDetector detects groups of white pixels and returns the center coordinates of the identified objects). A visualization of this process is shown in Figure 12.

The initial detection program applied a Gaussian blur to each frame prior to binary conversion. However, testing suggested that the performance of the detection program did not change for the idealized test conditions (laser pointer light in a dark room) regardless of the presence of the blur. Additionally, when a photo of a star was taken with the tracking camera used in the final prototype, the Gaussian blur eliminated the star completely due to its small size. With these results, the blur process was eliminated to increase the detection program's sensitivity.

This project focuses on ideal conditions, and the detector can therefore be assumed to return a single center coordinate. The coordinate then serves as an input to the PID controller, which moves the stage to keep the object in the center of the frame.

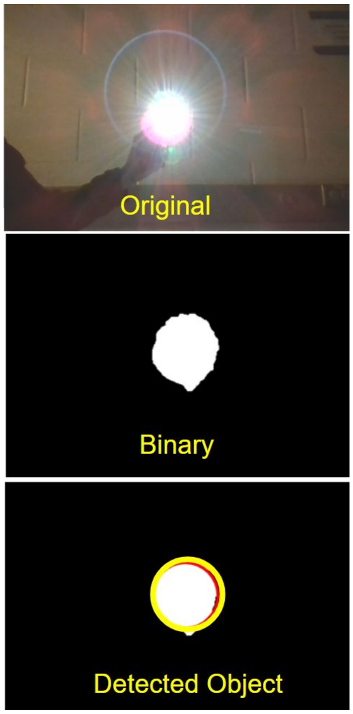


Figure 12: Example of Target Detection.

Original video frame of a flashlight in a dark room (top), frame after binary conversion, erosions, and dilations (middle), and the object identified by the SimpleBlobDetector (bottom)

3.4.2 PID Controller

Given the target's center coordinates and the instantaneous positions of the pan-tilt stage servo motors, the project can be visualized as a control problem where the center of the camera frame is taken as the reference point and the object coordinates are taken as the feedback signal. The objective becomes to drive the error between the two sets of coordinates to zero. A PID controller is used as the main control method, as it only requires the error between desired and actual signals. Figure 13. illustrates the control process.

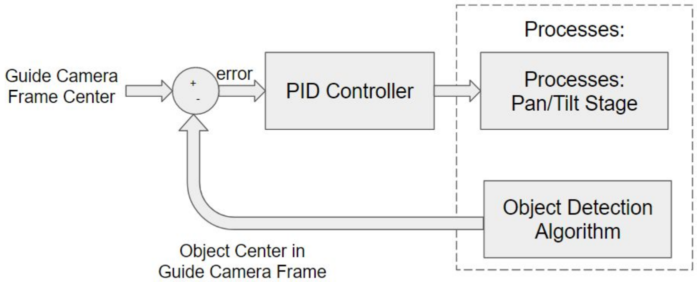


Figure 13. PID controller and the feedback signal

The PID controller requires precise tuning of gain values in order to prevent any unwanted oscillation while also providing fast enough responses to accurately track a moving object. Initially, the gains were set to constant values and adjusted manually until the camera center consistently approached the detected object. However, these fixed gains proved to be insufficient, resulting in either small overshoots but high rise times, or low rise times but large overshoots. To resolve this, the team opted to adjust gains in real-time. Taking the average distance between the detected object and center reference point over the previous three frames, the gains are scaled by a factor proportional to this distance value. This allows for large adjustments to be made extremely quickly, while also performing minute adjustments once the object is closer to the center of the frame.

Another important factor that had to be considered was the fact that the PID controller is purely reactionary. Unlike other mechatronic applications where the trajectory of the system is known, objects in the night sky have inconsistent trajectories. Due to the controller's reactionary nature, the stage will always lag behind the target, regardless of how precisely its position is controlled. Therefore, it was necessary to attempt to predict the future position of the target using its motion history. To make this prediction, the average angular velocity of the stage over the previous three frames are taken and converted into a pixel distance using the plate-scale equation. This approximated stage velocity is then combined with the velocity of the object relative to the stage, using the velocity of the object within the tracking camera frame. Adding this velocity vector to the position of the object in frame increases the error going into the PID controller, producing an overshoot in the direction of predicted object movement. Leading the target object ultimately allows for the stage to track objects with much more precision, resulting in more consistent data collection.

3.4.3 Encoder Incorporation and Servo Control Program

Inherently, servos are designed to take the target position as input and do not have any instantaneous position feedback. This caused complications in the PID controller, which expects to receive the instantaneous position of the pan-tilt stage. Non-elegant solutions include finding the potentiometer output from the servo to get the instantaneous position. In the prototype, capacitive absolute encoders are used to obtain the instantaneous position for both the pan and tilt axis. The encoders were calibrated such that their output and the angular position of the servos retain a linear relationship throughout the entire range of motion.

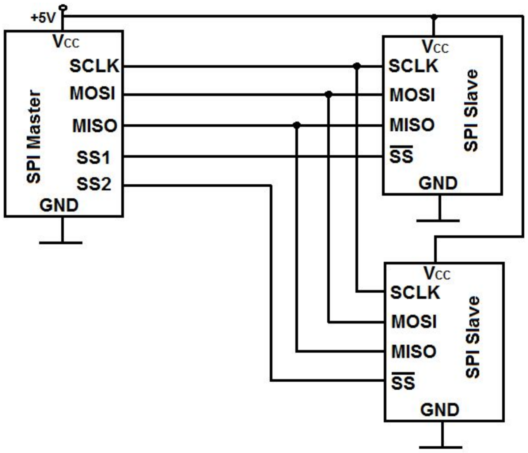


Figure 14. As implemented in the project, the Arduino is the SPI Master while each encoder is the SPI Slave. Image Source: [13]

Since the encoders use the SPI communication protocol, the team utilized the Arduino microcontroller to receive the encoder output. Fig 14. shows the hardware setup of the encoders and the Arduino board. The encoders continuously send their positional values to the Arduino which in turn waits for a signal from the computer over the serial port to broadcast the position. These values are broadcast as strings, so this call-and-response setup prevents data from being garbled or misread, as was the case when the position values were broadcast constantly. After the raw encoder values are fed into the computer, a simple linear relationship converts them into servo quarter-microsecond positions. This instantaneous position is then combined with the output of the PID controller to produce the target position of each motor. The Maestro controller takes in these target values and uses them to set the position of the pan-tilt stage to point the lens at the object.

4. System Performance

4.1 Indoor Testing

Since the occurrence of a potential target object is unpredictable, testing the system on real-life conditions is impractical. To test the system performance, an ideal real-life condition was simulated using a laser pointer in a dark room. The prototype was placed in a dark room, with no significant environmental distractions, and the laser pointer light pointed at a wall served as the target object.

While the indoor simulation allowed for system testing without waiting for a target object to appear in the sky, there was an additional calibration procedure required for the testing. Due to binocular disparity, the center of the camera frame does not correspond to the center of the lens. While this effect becomes negligible in real-life applications in which the distance between the device and the target object is very large, the offset is non-negligible during indoor testing, where the system-target distance is small. Therefore, in order to compensate for this offset, the target position for the PID must be shifted slightly off center. The necessary offset is a function of the distance between the lens and the object, and this relationship was obtained by connecting the optical fiber to a photodiode and moving a light source on the wall. A digital multimeter was used to read the voltage from the photodiode. In this setup, a maximum voltage reading is observed on the multimeter when the light is centered into the optical fibers. A photo was taken with the tracking camera at the maximum voltage point, and the offset was then determined by finding the pixel distance between the laser light and the center of the frame from the photo. Determining the offset for multiple distances revealed that, for a range of distances relevant for indoor testing, the relationship between lens-target distance and center offset can be approximated by Equation (3), where the offset is measured from the center of the camera frame and positive offset is to the right of the frame center.

$$offset\ in\ pixels = -0.07 \cdot (object - lens\ distance\ in\ meters) + 61.95 \tag{3}$$

Even with this calibration, keeping the laser within the target region did not guarantee that the spectrometer would be able to gather spectral information. This inconsistency can most likely be attributed to the distance at which the system was tested. While the vertical offset remained constant, the horizontal offset was found to be highly sensitive to the distance from the lens to the target. Even the change in perpendicular distance from the lens to the wall due to stage motion prevented effective data collection. As the stage rotated further in either direction, the targeted position and the area corresponding to the fiber bundle will eventually no longer overlap, preventing data collection. To minimize the effects of disparity as a result of stage rotation, all tests were performed with the system approximately 4 m away from the wall.

4.2 Results

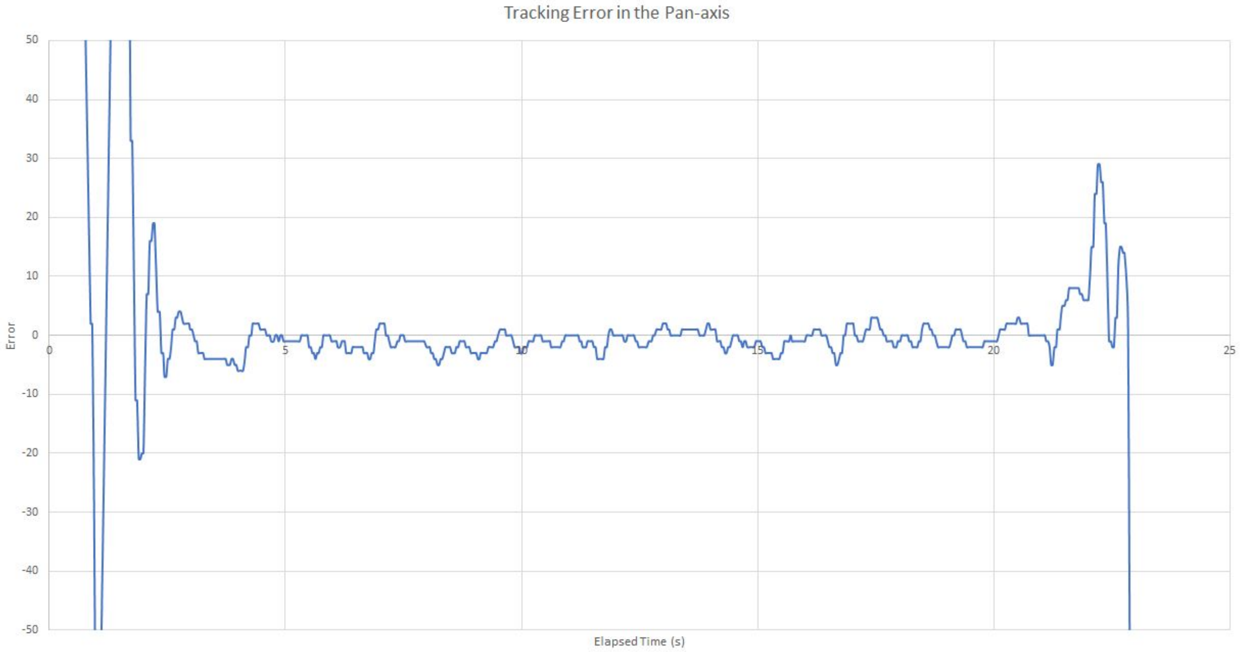


Figure 15. Tracking Error in the Pan Axis for Target Angular Velocity 2.5 deg/sec

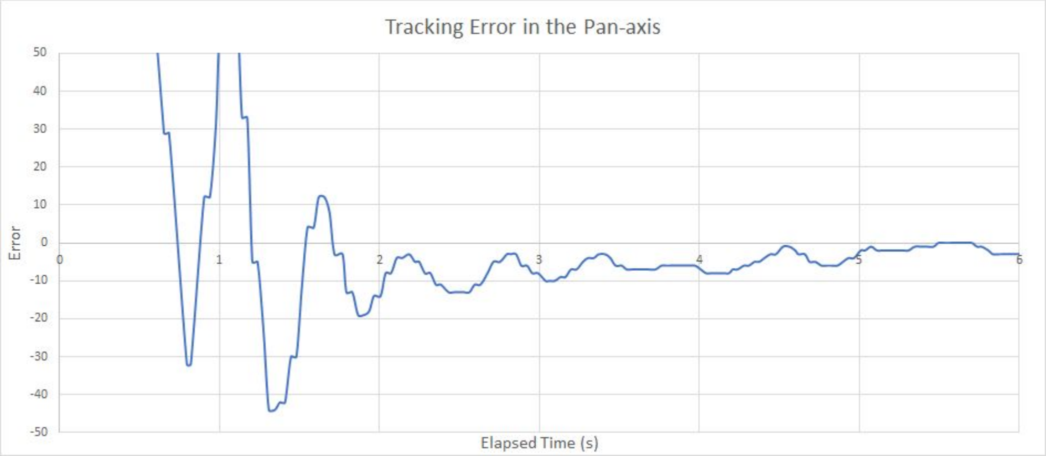


Figure 16. Tracking Error in the Pan Axis for Target Angular Velocity 5.0 deg/sec

Figures 15 and 16 show the tracking error in the pan axis for target object velocities 2.5 and 5.0 deg/sec, respectively. As seen in the error plots, the system is able to track an object under ideal conditions with an error within 5 pixels. While the error occasionally spikes due to changes in the direction of movement of the object, the target object is generally within the acceptable region. However, despite the light being within the area of the camera frame corresponding to the fiber bundle, the data output from the spectrometer remains intermittent at best. When the angular velocity of the target object is increased to the maximum expected value of 5° per second, the average error increases, but still remains within approximately 10 pixels of the target. The consistency of data collection decreases accordingly, but the spectrometer is still able to record meaningful information.

When testing indoors with the laser pointer, the software consistently detected the target, with issues only arising when the reflection of the laser pointer also appeared in frame. However, under ideal conditions, reflections are unlikely, making this issue a minor concern. A screenshot from one of the tests is shown in Figure 17.

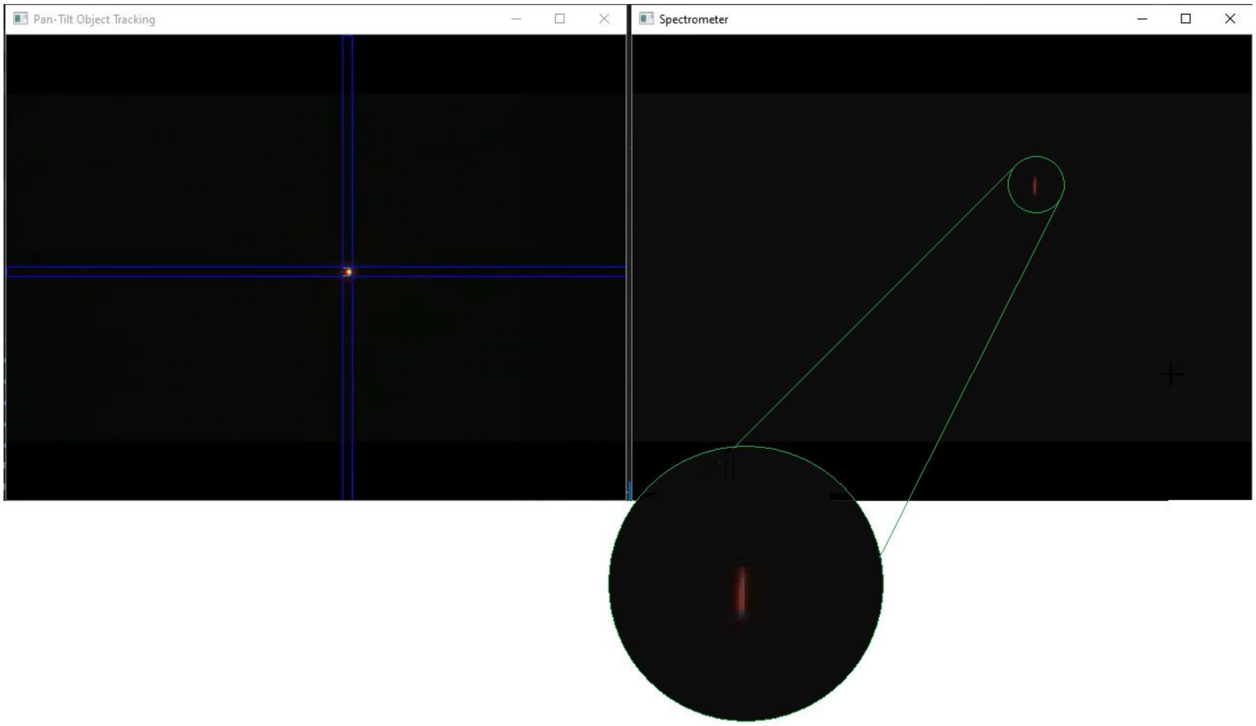


Figure 17. The left frame is outputted from the guide camera. The red laser light can be seen at the center of the frame due to successful tracking. At the same time, the right frame shows a spectrometer camera frame, outputting the corresponding spectrum footage. The sliver of red can be seen (a zoomed-in view is shown for better visibility.)

In terms of runtimes, the entire tracking program runs through a single detection-stage control cycle in approximately 35 ms, with approximately 85% of this time used for object detection and the remaining 15% going towards target position calculations and command issuing. Since the tracking camera records at 30 fps, reducing overall runtimes to 33 ms or less would be ideal in order to take full advantage of the available hardware and to improve tracking performance, but the effects of the 2 ms delay seen here should be negligible.

5. Project Deliverables

Table 2 in the appendix summarizes all deliverables completed throughout the project. The team developed a prototype of the system, and the following hardware components were included in a single package and submitted to the sponsor: (1) pan-tilt stage assembly (with the tracking camera, lens, and encoders attached securely with custom mounts), (2) Arduino UNO, Maestro board, breadboard, and batteries attached to a stable base, (3) custom adapters for the optical fibers, and (4) spectrometer and optical fiber (returned to the sponsor). The PC controller will not be included in the package, and the user is required to provide his or her own computer.

6. Recommendations

6.1 Processor

The team members' personal Windows 10 laptops were used to run the device during prototype development and testing. This means that the runtime of the code was highly dependent on a multitude of factors, including, but not limited to, the influence of background processes in the host computer, type of processor, and the number of multiprocessing threads. A dedicated microprocessor that is optimized to the operational needs of the project can be incorporated to solve this problem. A microprocessor with vision capability and the ability to communicate through common protocols would be a suitable replacement. A dedicated high-performance computer is another option.

6.2 Pan-Tilt Stage

For similar projects in the future, use of motors or servos with speed control to move the pan-tilt stage is recommended. While use of such motors require modifications to the PID controller, the ability for speed control eliminates the need for instantaneous motor position feedback, eliminating the need for encoders and an Arduino board. Using speed control rather than positional control also expands the options for control methods. For example, a predictive open-loop car-following control (POL-CFC)[12], which would result in smoother stage motion compared to a PID controller, becomes an option with speed control. The POL-CFC method also uses Kalman Filters to predict the next position of the target, which can lead to reduced lag, increasing system tracking accuracy. An example of an alternative pan-tilt stage is included in the appendix.

6.3 Guide Camera

While the color camera chosen by the team successfully tracked the laser pointer in a darkened room, it may struggle to identify real-life targets in the actual night sky. In future attempts, it may be beneficial to use a monochrome camera instead of a color camera. In a color camera, each sensor is filtered such that it can only register one color of light (red, green, or blue), and the other colors in each pixel are calculated based on those of its neighbors. However, in a monochrome camera, these filters are not present, allowing each sensor to receive up to three times the light compared to a color camera, resulting in an overall higher sensitivity. Additionally, monochrome cameras boast a higher signal-to-noise ratio compared to color cameras, further reducing the need for a noise reduction step during object detection, leading to shorter per-cycle run times.

6.4 Object Detection Algorithm

During testing, the simple blob detector algorithm was sufficient to identify a single object of interest against a plain background. However, this idealized scenario may not always hold during actual use. A finalized program would ideally be much more robust, and be able to track a single point of interest among multiple stationary spots of light. For example, the program should ideally be able to isolate and track a single moving target within a group of stars of similar brightness. One potential approach to this issue would be to combine the current algorithm with a background subtraction program to eliminate stationary lights. However, such a program would also require adjustments of its own to accommodate the moving camera frame.

The team previously experimented with such an approach but was forced to sideline it in order to meet other needs.

Additionally, improving the runtime of the object detection algorithm would improve the accuracy of the stage tracking. Currently, the object detection portion of the code runs in approximately 30 ms, with the control and tracking segments taking another 5 to 10 ms. However, the object detection processing time increases with more objects in the frame. While the situation is outside of the idealized scenario the team tested for, the processing time can be expected to increase during actual use. The camera updates the image every 33 ms, so the current program cannot take full advantage of the camera's frame rate in its current state. One option for improvement would be running the object detection program in parallel with the control program. This would allow for the target position to be updated while the next frame is being processed, leading to more frequent updates to the stage's target position, and in turn, more precise tracking. Another alternative would be to rewrite the entire software package in C or C++. Since Python is a higher-level language than the aforementioned languages, its runtimes tend to be higher than the same program written in a different, lower-level language. Depending on the scale of these runtime improvements, the 30 fps camera could potentially be replaced with a 60 fps camera for an even more responsive system to further increase the tracking accuracy.

7. Budget

The budget for this project was provided by Professor Nick Glumac, sponsor and advisor of the project. The team spent roughly $900 out of the provided $1500, with the funds primarily being used to purchase hardware from external sources. Some funds were used for rapid prototyping purposes in the Innovation Studio, with the majority of these costs going towards hardware for mounting the encoders. A detailed breakdown of the budget is attached in the appendix.

8. Conclusions

The primary goal of the transient object spectrometer project was to detect and track a moving, light-emitting object in the night sky, while collecting its spectral data to aid in the identification of the light source. The developed prototype combines a servo-driven pan-tilt stage, a tracking camera, a lens, encoders, a spectrometer with an optical fiber bundle, a PC controller, an Arduino microcontroller, and a Maestro control board into a single package. The video stream from the guide camera is used by the object detection program to output the coordinates of an object in the frame. The coordinates and the instantaneous position readings from the encoder are used by a PID controller to direct the pan-tilt stage to move in a way such that the target light is centered in the lens. The final prototype was able to demonstrate all of the desired functionality, tracking a laser pointer on the wall with enough precision to collect information via the spectrometer. However, there are some improvements that can be made for better system response.

9. Acknowledgements

The team would like to express great appreciation to Professor Nick Glumac for sponsoring this project and for his valuable suggestions and constructive feedback as an adviser. We would also like to thank our TA, Sharon Edward for the time and effort she has dedicated in supporting and guiding the team throughout the project. We would also like to acknowledge the technical advice provided to the team by Professor Dan Block. Finally, we would like to thank the MechSE Business Office for their expeditious services.

10. References

[1] Castillo, Amanda del. “Mysterious Light Seen in Night Sky over California.” *ABC7 San Francisco*, 20 Dec. 2018, abc7news.com/science/mysterious-light-seen-in-night-sky-over-california/4932239/.

[2] Griffiths, Sarah. “The Mystery of Norway's Hessdalen Lights May Be Solved as Expert Claims Natural 'Battery' Creates the Light Show.” *Daily Mail Online*, Associated Newspapers, 19 May 2014, www.dailymail.co.uk/sciencetech/article-2632650/Has-mystery-glowing-Norwegian-orbs-solved-Expert-claims-underground-battery-creates-amazing-light-show.html.

[3] Ukida, Hiroyuki. “SICE Annual Conference 2010.” IEEE, *Proceedings of SICE Annual Conference 2010*, 2010.

[4] “Life Sentry Guns for Sale - Home.” *Real*, realsentrygun.com/index.htm.

[5] Okumura, Kohei, et al. “1 ms Auto Pan-Tilt – Video Shooting Technology for Objects in Motion Based on Saccade Mirror with Background Subtraction.” *Advanced Robotics*, vol. 29, no. 7, 2015, pp. 457–468., doi:10.1080/01691864.2015.1011299.

[6] Mitiche, Amar. *Computer Vision Analysis of Image Motion by Variational Methods*. SPRINGER INTERNATIONAL PU, 2016.

[7] R. Feghali and A. Mitiche, "Spatiotemporal motion boundary detection and motion boundary velocity estimation for tracking moving objects with a moving camera: a level sets PDEs approach with concurrent camera motion compensation," in IEEE Transactions on Image Processing, vol. 13, no. 11, pp. 1473-1490, Nov. 2004. doi: 10.1109/TIP.2004.836158

[8] V. Philomin, R. Duraiswami and L. Davis, "Pedestrian tracking from a moving vehicle," *Proceedings of the IEEE Intelligent Vehicles Symposium 2000 (Cat. No.00TH8511)*, Dearborn, MI, USA, 2000, pp. 350-355. doi: 10.1109/IVS.2000.898368

[9] I. Cohen and G. Medioni, "Detecting and tracking moving objects for video surveillance," Proceedings. 1999 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (Cat. No PR00149), Fort Collins, CO, USA, 1999, pp. 319-325 Vol. 2. doi: 10.1109/CVPR.1999.784651

[10] Dooling, Dave. “Optical Astronomy.” Salem Press Encyclopedia of Science, 2018. EBSCOhost

[11] A. Drouard et al. “Probing the use of spectroscopy to determine the meteoritic analogues of meteors” A&A 613 A54 (2018) doi: 10.1051/0004-6361/201732225

[12] J. Ahmed, A. Ali, and A. Khan, “Stabilized active camera tracking system,” Journal of Real-Time Image Processing, vol. 11, no. 2, pp. 315–334, May 2012. doi: 10.1007/s11554-012-0251-z

[13] “How to Connect Multiple SPI Devices to an Arduino Microcontroller.” Learning about Electronics,

learningaboutelectronics.com/Articles/Multiple-SPI-devices-to-an-arduino-microcontroller.php.

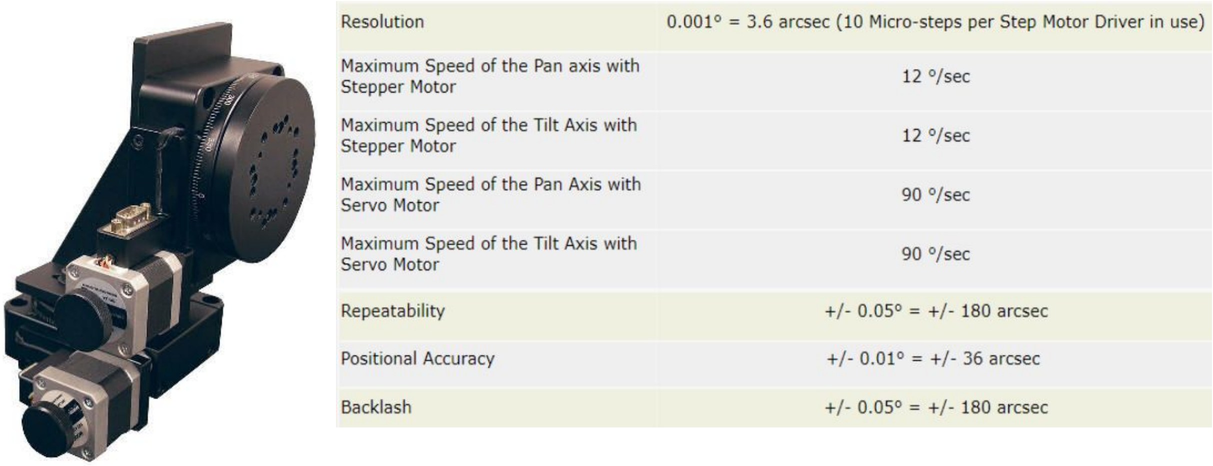
Appendix

Table 1. Budget

Item	Supplier	Unit Cost	Quantity	Total Cost
Micro Maestro 6-Channel USB Servo Controller	Pololu	19.95	1	19.95
Thin (2mm) USB Cable A to Mini-B, 5 ft.	Pololu	2.75	1	2.75
PT785-S Pan & Tilt System	ServoCity	349.99	1	349.99
Spinel UC20MPG-L60 Camera	Spinel	48	1	48
Takumar 135 mm f/2.5 Bayonet Lens	eBay	56	1	56
Blomiky 2 Pack 6.0 V 700 mAh Ni-CD AA Rechargeable Battery Pack	Amazon	13.98	1	13.98
Depets Rechargeable Red Laser Pointer	Amazon	8.99	1	8.99
PT-0613 6.00 mm f/1.2 Lens	M12 Lenses	15	1	15
PT-0613 6.00 mm f/1.6 Lens	M12 Lenses	6.5	1	6.5
HASMA - SMA Bulkhead Adapter with Lock Nut	ThorLabs	8.55	1	8.55
CUI Devices AMT223B-V Motor Encoder	Mouser	50.04	2	100.08
CUI Devices AMT-PGRM-06C Encoder Programming Cable	Mouser	17.68	1	17.68
HS-785HB Servo	ServoCity	49.99	2	99.98
32P, 24T C1 Spline Servo Mount Gear (Metal, 12T)	ServoCity	14.99	2	29.98
AMT-06C-1-036 Encoder Cable	DigiKey	35.82	2	71.64
Arduino Uno	ECE Supply Center	22.95	1	22.95
Breadboard	ECE Supply Center	7.21	1	7.21
USB A-Male to B-Male	ECE Supply Center	0.6	1	0.6
M/M Jumper Wires	ECE Supply Center	0.34	1	0.34
Miscellaneous Hardware/3D Printing	Innovation Studio	25	1	25
		Total		**$905.17**
		Remaining		**$594.83**

Table 2. Deliverables

Action Item
Proposal Report
Status Report
Hardware Items: 1. Single mount to secure camera and lens on pan-tilt stage 2. Connector to join lens to the optical fiber cable
Object Detection & Tracking Algorithm
PID Controller with Servo command software
Spectrometer Interaction Software
Code Documentation with Test Results
Final Report
Video



Resolution	0.001° = 3.6 arcsec (10 Micro-steps per Step Motor Driver in use)
Maximum Speed of the Pan axis with Stepper Motor	12 °/sec
Maximum Speed of the Tilt Axis with Stepper Motor	12 °/sec
Maximum Speed of the Pan Axis with Servo Motor	90 °/sec
Maximum Speed of the Tilt Axis with Servo Motor	90 °/sec
Repeatability	+/- 0.05° = +/- 180 arcsec
Positional Accuracy	+/- 0.01° = +/- 36 arcsec
Backlash	+/- 0.05° = +/- 180 arcsec

Figure 18. Example of Alternate Pan-Tilt System, the PT100 Pan-Tilt Stage by OES