

# Precise Tracking of UAVs using LiDAR and Computer Vision



AE 8900 MS Special Problems Report  
Space Systems Design Lab (SSDL)  
Guggenheim School of Aerospace Engineering  
Georgia Institute of Technology  
Atlanta, GA

Author:  
Rachmat Subagia

Advisor:  
Prof. Brian C. Gunter

June 2019

# Precise Tracking of UAVs using LiDAR and Computer Vision

Rachmat Subagia\* and Brian C. Gunter†

*AE 8900 Special Problem Report*

## Abstract

The ubiquitous potential application of unmanned aerial vehicles (UAVs) has sparked many research topic areas for the past two decades, including remote sensing, dynamics system and control, and computer science. In this work, we establish a pan-tilt mechanism to support the integration of a ground-based UAV tracking system, which consists of a color-based machine-vision camera for object detection and LiDAR for object ranging. The support mechanism provides two degrees of freedom that allows the sensor to rotate in pitch and yaw direction, tracking the object of interest to a certain degree, that is, with side constraints in the two directions: -23.5 deg to 38.5 deg for tilt rotation and 25 to 151.5 deg for pan rotation. Furthermore, the refinement to improve the precision of the real-time measurement from LiDAR is also considered using an Extended Kalman Filter (EKF)-based algorithm. The experimental results have proved the effectiveness of this method. It is shown that the estimation of error sources inherent in this sensor improves the accuracy of these measurements. The error accuracy from the pan-tilt mechanism are 0.14 deg with standard deviation of 0.25 deg for pan angle and 0.22 deg with standard deviation of 0.38 deg for tilt angle. The measurement from lidar is in general overestimates the true value, for which the offset is linearly modeled. The measurement from UAV tracking provided approximately 15 cm accuracy in altitude, and about 5 cm accuracy in horizontal positions.

**Keywords:** computer vision; LiDAR; Extended Kalman filter; object detection and ranging

---

\* Master's Student, Department of Aerospace Engineering, Georgia Institute of Technology

† Associate Professor, Space System Design Laboratory, Georgia Institute of Technology

# 1. Introduction

Small unmanned aerial vehicles (UAVs) and micro-drones have rapidly proliferated over the past two decades, and they are used in an ever increasing number of activities including commercial (e.g., filming, agriculture, delivery package), military, and remote sensing applications to mention a few. As a result, improving the design, performance, and tracking capability of the UAVs is becoming more important for users, operators, and researchers, as it enables more advanced and complex applications. In remote sensing applications for example, several studies have focused on developing high resolution imagery [1-3] which is beyond that provided by satellites due to limited resolutions from current technologies for short- and close-range domain applications. Other studies have focused on improving the accuracy of topography [4, 5] and enhancing navigation system in hazardous environments [6, 7].

The use of light detection and ranging (LiDAR) to improve the accuracy of topographic mapping by UAVs and other aforementioned applications has also been implemented. Sankey et al. [8], for example, demonstrated a lidar-hyperspectral image fusion method in treated and control forests in northern Arizona, USA. The authors found that the LiDAR data produces highly accurate digital elevation model. In addition, Wallace et al. [9] found that using LiDAR improves the uncertainty of the measurement of tree locations. It is important to point out that the accuracy of LiDAR topography for example, is only as good as the position accuracy of the platform. In the case where centimeter-level topography is required such as in stereo imaging, Interferometric Synthetic Aperture Radar (InSAR), and gravity surveys, a very precise positioning knowledge of the sensors, which in this case is LiDAR will also be expected, that is to be in centimeter level accuracy.

However, in traditional real-time tracking that uses GPS receivers using civilian broadcast code for instance, the accuracy is typically at the decimeter or centimeter level, but degraded in the vertical direction as pointed out in recent studies conducted by Schipperjin et al. [10] and Ruffini et al. [11] to mention a few. This is the motivation of this work. The objective is that examining whether or not it is possible to design a low-cost method to provide complementary, high-accuracy ground-based ranging measurements that would allow for the precise positioning in the centimeter level in these more demanding applications.

On a related note, object detection is an active research topic in the field of computer vision that focuses on detecting, recognizing and tracking object through a series of image frames. Object tracking, on the other hand, is defined as the task of detecting an object every time, or every frame in the series of

images, and establishing the correspondence between the detected objects from one frame to another. Two major methods of object detection have been widely implemented in respective field: shape- and color-based object detection. The shape of an object can be utilized as a global feature to detect an object with a distinct shape, e.g., straight line, polygons, circles, or any other irregular shapes. Gavrila and Philomin [12] for instance, demonstrated shape-based object detection method, which uses a template hierarchy to capture the variety of object shapes, and implemented it on the real-time detection of traffic signs and pedestrians from a moving vehicle. Color-based object detection, on the other hand, is useful when an object to be detected is of or can be assigned to a specific color and this color is different from the color of the background or other objects in the environments. One interesting application of this method is conducted by Ahmad et al. [13] who developed effective target tracking system using Pixy CMUcam5 sensor that is color-based and applied this to luggage carrying following cart that could lighten the burden for wheelchair users. The latter method will be used for our study and leave the other for fruitful venue in future for interested readers.

In this work, the CMUcam5 sensor is involved to maintain active tracking of the UAV so that the laser pulses by LiDAR continue to hit the UAV to provide the complementary positioning data. At a certain distance of, say, 1 km or more, depending on the specification of the camera, the UAV will be nothing more than a point mass in the sky. Therefore, in order for the camera sensor to be able to continue tracking the object, some sort of blinking diode will be needed to be attached to support for object detection. Furthermore, computer vision is also used to provide independent validation of the positioning data we obtain from the GPS/lidar system. An illustration of how this is implemented in this work is shown in Figure 1.

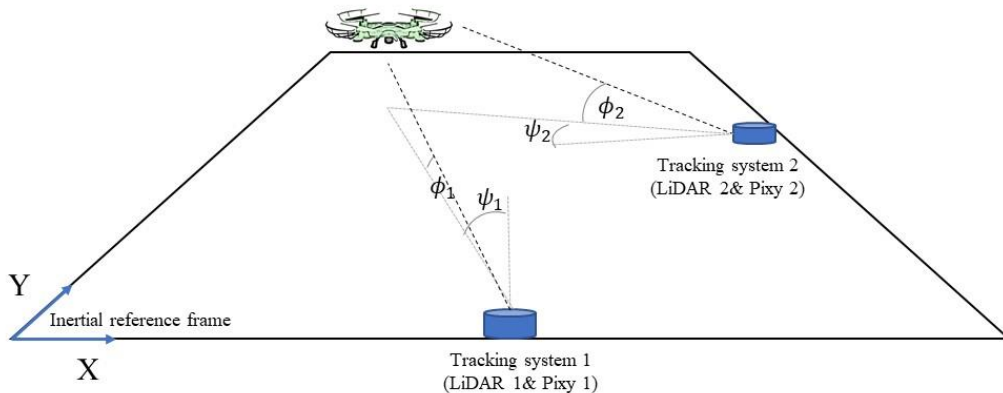


Figure 1. Schematic experimental setup on how LiDAR and CMUcam5 sensors are used in this work

One example work that uses LiDAR and computer vision for object detection was done by Huang and Barth [14], which proposed tightly-coupled LiDAR/computer vision integrated system for vehicle

detection. The authors presented that the system is reliable and can be used in applications such as traffic surveillance and roadway navigation tasks. In addition, two other works use these two technologies; one for automated navigation [15], and the other for pedestrian detection system [16]. Connected to these studies, the objective of this work is to demonstrate low-cost vision-based object detection and ranging with ground-based tracking system. More specifically, we integrate these two technologies to detect flying object such as small UAV and examine its accuracy for object detection and ranging. The key idea of the method is to detect the target and show its position relative to the sensor. The remainder of this work is organized as follows. In section 2, we provide introduction to computer vision and LiDAR that are relevant to this work. In section 3, we discuss experimental setup and extended Kalman filter (EKF) for refinement of measurement. In section 4, we provide experimental results of tracking of UAVs and discussion. Finally, we conclude this work in section 5.

## **2. Methodology**

In this section, we discuss essential background that are relevant to present work. We begin by discussing brief overview about the notion of computer vision and its role on object detection. The discussion is then continued with light detection and ranging before closing it with microcontroller.

### **2.1. Computer vision: object recognition**

Computer vision is an interdisciplinary scientific field that deals with how computers are programmed to gain high-level understanding obtained from digital images or videos. From engineering perspective, it aims to build autonomous systems which could perform some of the tasks that the human visual system performs, of which will be interest of this work [17]. The notion of computer vision includes methods for acquiring, processing, analyzing and understanding digital images in order to produce numerical or symbolic information in the forms of decisions, for object detection and navigation, as an example. For Morris [18], the domain of computer vision include scene reconstruction, event detection, video tracking, object recognition, 3D pose estimation, learning, indexing, motion estimation, and image restoration. In this work, we will focus on the role of computer vision in object recognition to detect color-based object such as a UAV.

In autonomous vehicles such as submersibles, land-based vehicles, and aerial vehicles, the application of computer vision ranges from fully unmanned vehicles to vehicles where computer-vision-based systems support a driver or a pilot in various situations. The former is typically used for navigation purposes, such as determining its location, producing a map of its environment which is commonly referred to as simultaneous localization and mapping (SLAM), detecting obstacles, or

detecting certain task in specific events such as UAV for forest fires or remote-sensing related application to mention a few. The latter is commonly used to obstacle warning systems or autonomous landing of aircraft. NASA's Mars Exploration Rover and ESA's ExoMars Rover are other examples of the application of autonomous vehicles using computer vision for space exploration.

To narrow down our discussion related to present work, we begin with how object detection in computer vision works, which is highly related to image processing. Computer recognizes image as a signal with pixel values at any point in it. Pixel, a basic unit of digital image, represents values for its brightness, and color, hence called intensity. The signal is the continuous voltage signal given by visual sensor that will be converted to digital form with some processes such as sampling. The digital form of these data can be treated as two-dimensional array or matrix able to construct the digital image. Gaining information about the image recorded by the computer is the task of computer vision. Analogous to human vision that has the ability to learn and providing data from visual inputs, computer vision is a way of understanding digital images to collect some information from it such as its shape, color, or size and classify it based on certain rules. In robotic applications, two ways of image processing, i.e., using usual camera module that provide the image without any prior processing on it, or using machine-vision camera that have built-in image processing for faster and practical use. The latter will be used in present work since we are interested in real-time detection for moving objects, i.e. Pixy, which has been widely used in robotic applications.

Pixy is one of the camera modules specified for image processing for which the recognition algorithm is color-based filtering. The main purpose of this camera is to recognize objects by colors and label them as a familiar object. Specifically, Pixy2, as shown in Figure 2 is used in this experiment because it is faster and have more resolution in terms of image recognition of 60 frames-per-second compared to its earlier version, which is required for this work. The discussion of the technical specification in Table 1 on the performance of object tracking is addressed in the next section when necessary. In addition to color-based filtering machine-vision camera Pixy, two other similar types of cameras are publicly available, OpenMV and Jevois. A comparative analysis of the performances of Pixy, OpenMV, and Jevois is beyond the scope of this work, and is left as a fruitful venue for future work.



Figure 2. Pixy2 CMUcam5 vision-sensor camera used in this work

Table 1. Technical specifications of Pixy2 CMUcam5 vision sensor

Technical specification	Description
Processor	NXP LPC4330, 204 MHz, dual core
Image sensor	Aptina MT9M114, 1296×976 resolution with integrated image flow processor
Lens field-of-view	60 degrees horizontal, 40 degrees vertical
Power consumption	140 mA typical
RAM	264 KB
Flash	2 MB

## 2.2. Light detection and ranging

LiDAR, light detection and ranging, is a surveying method that measures distance to a target by irradiating the target with pulsed laser light and measuring the reflected pulses with a sensor. Differences in laser return times or delay time and wavelengths are then used to create digital 3-D representations of the target by having knowledge of its distance and orientation with respect to the sensor. Broadly speaking, LiDAR is commonly used to create high-resolution maps with applications in geodesy, geomatics, forestry, atmospheric physics, laser guidance, airborne laser swath mapping (ALS), and laser altimetry, to mention a few. Recently, LiDAR has also been used in autonomous vehicle for continuous mapping and obstacle detection.

LiDAR works with ultraviolet, visible, or near infrared light to image object of interests. It has the capability to target a wide range of materials, including non-metallic objects, rocks, rain, chemical compounds, aerosols, even clouds. Furthermore, a narrow laser beam can map physical features with very high resolutions. For example, the National Oceanic and Atmospheric Administration (NOAA) [19] demonstrated that an aircraft can map terrain with 30-centimeter resolution or better. Another

application was performed by NASA, identifying LiDAR as a key technology for enabling autonomous precision safe landing of future robotic and crewed lunar-landing vehicles [20]. In addition to these applications, LiDAR has also been used to measure distances to objects such as small UAVs. Such applications are referred to as laser guidance.

In general, laser guidance aims a robotic system to a target position or object of interests by means of a laser beam. Paromtchik [21] demonstrated that the laser guidance of a robot can be accomplished by projecting a laser light, image processing and communication in order to improve the accuracy of guidance along a desired route in an environment that is unknown a priori. The key purpose is to show target positions to the robot by laser light projection instead of communicating them numerically, which simplifies mechanism in directing the robot while improving the positioning accuracy and allows for implicit localization with the visual feedback. In military application, for example, laser guidance is specifically referred to as a technique of guiding weapon such as a missile, or other projectiles to a target using a laser beam or spot, e.g., beam-riding guidance (also known as Line-of-Sight Beam Riding – LOSBR) and semi-active radar homing (SARH). Glaros et al. [22], for instance, proposed a novel approach of striking a moving target to determine its position and velocity by laser lead designator with laser guided weapon (LGW) and a controller. In their work, the targeting sensor of the LGW detects laser energy from the off-target spot position, not energy from a laser marked/designated target. The signal processing of LGW provides guidance commands to its control system which directs the LGW to the off-target spot position, leading to a trajectory that results in the LGW impacting the moving target. However, as opposed to also provide guidance to the UAV, this work is limited to merely provide precise laser ranges from a known base station.

The technique that is sometimes called semi-active laser homing (SALH) requires a laser to be kept pointed at the target or object of interest. The laser radiation bounces off and is scattered in all directions, which is known as laser painting. When the projectile is in the vicinity of the target for some of the reflected laser energy to reach it, a laser seeker will then acquire information about which direction this energy is originated and adjusts the projectile trajectory towards the source. While the projectile is in the general area or some positions at some time, the laser is kept aimed at the target, allowing the projectile to be accurately guided to the target. It is important to note that SALH is limited to targets that are able to reflect much laser energy. Stated differently, the technique is not practical against the targets that do not have laser reflective materials, such as those coated with paints that absorbs the laser energy.

In the present work, we will use LiDAR-Lite V3\* , shown in Figure 3, on the ground to measure the position of detected object. This LiDAR is a compact optical distance measurement sensor from Garmin™ which is ideal for unmanned vehicle applications because it is lightweight. LiDAR is commonly mounted in the UAVs and used for 3D mapping. In this work, however, we will use this as ground-based LiDAR that is projected to the object of interest detected by the machine-vision camera. LiDAR Lite works with near-infrared laser signal with wavelength 905 nm, calculating the time delay between the transmission of the laser signal and its reception after reflecting off of a target to measure the distance by using the known speed of light. The unique signal processing approach developed in LiDAR Lite transmits a coded signature and looks for that signature in the return, allowing for highly effective detection with eye-safe laser power levels. In addition, proprietary signal processing techniques are used to achieve high sensitivity, speed, and accuracy in a small, low-power, and low-cost system. We will further discuss in the next section the specific capability and theory of operation of LiDAR Lite, along with the constraints that limit its applicability in present work.



Figure 3. LiDAR Lite V3 distance sensor from Garmin™ (Size L×W×H: 20×48×40 mm)

In terms of the performance, LiDAR Lite V3 has range up to 40 m with 70% reflective target and  $\pm 1$  cm of resolution. In addition, LiDAR Lite V3 has typically  $\pm 2.5$  cm of accuracy when the distance is less than 5 meter, and  $\pm 10$  cm otherwise. However, it is important to note that nonlinearity may be present for the distance below 1 meter, which will also be assessed later in this work. The update rate of the sensor is typically 270 Hz but can also perform at 670 Hz in fast mode with consequently reduced sensitivity. This is an important factor in real-time tracking for fast moving object, which will inevitably provide limitations at which the moving object cannot be well measured due to its fast movement beyond the lidar sensor capability, regardless the tracking performance of the vision-sensor camera.

---

\* Interested readers can find technical specification of this sensor at

[https://static.garmin.com/pumac/LIDAR\\_Lite\\_v3\\_Operation\\_Manual\\_and\\_Technical\\_Specifications.pdf](https://static.garmin.com/pumac/LIDAR_Lite_v3_Operation_Manual_and_Technical_Specifications.pdf)

The successful reception of a reflected signal is heavily influenced by several factors. These factors include target distance, target size, aspect, and reflectivity [23]. First, in target distance, the relationship of distance ( $D$ ) to returned signal strength is an inverse square. Therefore, an increase in distance resulted in the decrease in returned signal strength by  $1/D^2$  or the inverse of the square of the distance. Second, in target size, the relationship of Cross Section ( $C$ ) of a target to returned signal strength is an inverse power of four, that is, when the beam overfits (is larger than) the target, the signal returned decreases by  $1/C^4$  or the fourth power of the target's cross section. Third, the aspect of the target, or its orientation to the sensor, affects the observable cross section and, therefore, decreases the amount of returned signal as the aspect of the target varies from the normal. Lastly, the reflectivity characteristics of the surface of a target also affect the amount of returned signal received by the sensor. It is important to note that in this case, we concern ourselves with reflectivity of near infrared wavelength, which is addressed shortly.

Different surfaces have different behavior of reflecting signals that affects the quality of returned signal to the sensor. Figure 4a illustrates purely diffuse surfaces founded on materials with textured surface, causing the reflected energy to disperse uniformly. This tendency results in a relatively predictable percentage of the dispersed laser energy in finding its way back to the device. As a result, these materials tend to reflect signal well. Materials included into this category are paper, matte walls, and granite. It is important to note that materials that fit into this category due to observed reflection at visible light wavelengths may exhibit unexpected results in other wavelengths. The near infrared range used by this device, of which we are working at, may detect them as nearly identical. Figure 4b illustrates that for specular surfaces that are found on materials that have a smooth quality that reflect energy instead of dispersing it. In this situation, it is difficult if not impossible for the lidar to recognize or gain information about the distance of many specular surfaces. The reflections off of specular surfaces are prone to reflect with low dispersion which causes the reflected beam to remain small and, because of that, no signal will be directly back to the receiver. This indicates that the lidar may fail to detect a specular object in front of it unless viewed from the normal.

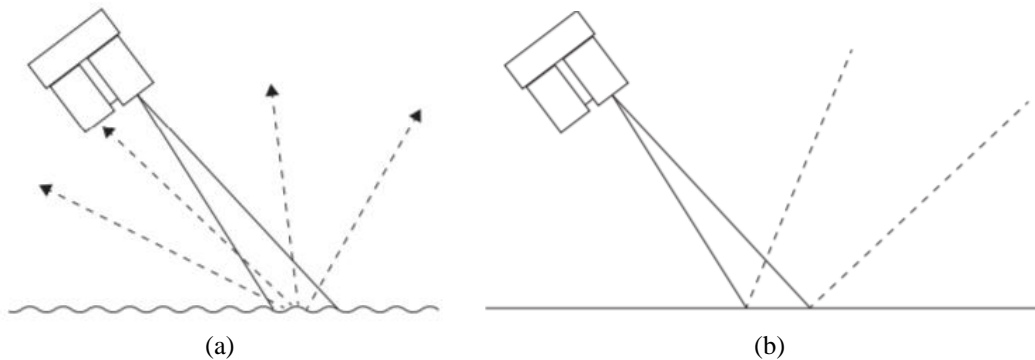


Figure 4. The reflective characteristic of an object (a) diffuse reflective surface (b) specular reflective surface [23]

### **2.3. Microcontroller**

Tracking devices on robotic systems in general involve microcontroller as a main source of command or control. A microcontroller is a self-contained system with a processor, memory, and peripherals, and it can be used as an embedded system [24]. In contrast to the microprocessors that are used in personal computers or other purpose applications requiring various discrete chips, microcontrollers are designed for embedded applications such as automobile engine control systems, implantable medical devices, and remote controls, to name a few. Furthermore, microcontrollers are often more economical and digitally able to control more devices and processes compared to using separate microprocessors, memory, and input/output (I/O) devices. Most microcontrollers available today are already embedded in other machinery, robotic systems, and computer systems, for instance. In addition, in the context of internet of things, microcontrollers are also considered as economical and popular in terms of data collection, sensing, and actuating the physical world as edge devices. In the present work, a microcontroller plays the role of integrating laser guidance and machine-vision camera for object detection and ranging.

One of the most widely used open-source single-board microcontrollers is Arduino. It is based on a simple I/O board and a development environment that implements the Processing programming Integrated Development Environment (IDE), which works essentially the same program as C++ [25]. This software enables the user to write and edit code, then convert this code into instructions that Arduino Hardware understands and executes. Several kinds of engineering problems can be solved by applying the Arduino program, especially in robotic, mechanical, aerospace, and electrical areas which requires an automatic command. For instance, in aerospace engineering, the Arduino microcontroller is used to control the command of Unmanned Aerial Vehicle so that the UAV is autonomous and move as it is programmed in the microcontroller. In present work, to move the pan-tilt system in order to follow along the object of interests, the Arduino gives commands to the motor driver, thus the motor driver control the motor (such as stepper motor, DC motor, and servo motor). Arduino Uno is the first generation of the Arduino board. It is based on the ATmega328 and has 14 digital input/output pins, 6 analog inputs, a 16 MHz ceramic resonator, a USB connection, a power jack, an ICSP header, and a reset button. Figure 5 illustrates the general component of Arduino Board.

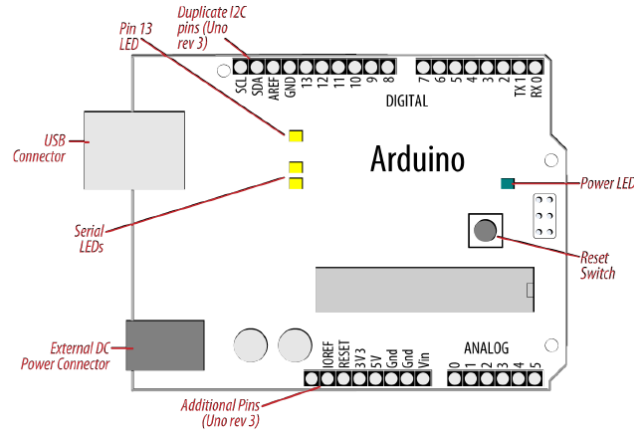


Figure 5. Basic Arduino Board - Duemilanove and Uno [26]

### 3. Experimental Setup

In this section, we discuss the experimental setup and the integration made in order to build the ground-based object tracking using LiDAR Lite V3. We begin by illustrating how it is mounted and integrated with Pixy2. Then, we continue with methods we use to reduce noise of measurement from lidar as well as refinement to improve measurement accuracy in terms of position of the object of interest.

#### 3.1. Pan-tilt mechanism and calibration

To take a measurement, LiDAR Lite first performs a receiver bias correction routine, correcting for changing ambient light levels and allowing maximum sensitivity. Then it sends a reference signal directly from the transmitter to the receiver. It stores the transmit signature, sets the time delay for “zero” distance, and recalculates this delay periodically after several measurements. Next, it initiates a measurement by performing a series of acquisitions. Each acquisition is a transmission of the main laser signal while recording the return signal at the receiver. If there is a signal match, the result is stored in memory as a correlation record. The next acquisition is summed with previous result. When an object at a certain distance reflects the laser signal back to the device, these repeated acquisitions cause a peak to emerge, out of the noise, at the corresponding distance location in the correlation record.

The lidar sensor integrates acquisitions until the signal peak in the correlation record reaches a maximum value. If the returned signal is not strong enough for this to occur, the device stops at a predetermined maximum acquisition count. Signal strength is calculated from the magnitude of the signal record peak and a valid signal threshold is calculated from the noise floor. If the peak is above this threshold, the measurement is considered valid and the device will calculate the distance. When beginning the next measurement, the device clears the signal record and starts the sequence again. In

implementing this, Figure 6a illustrates how I2C wiring from LiDAR Lite to Arduino where 680 $\mu$ F electrolytic capacitor is used to correct polarity. The additional note on the supporting label is provided in [23].

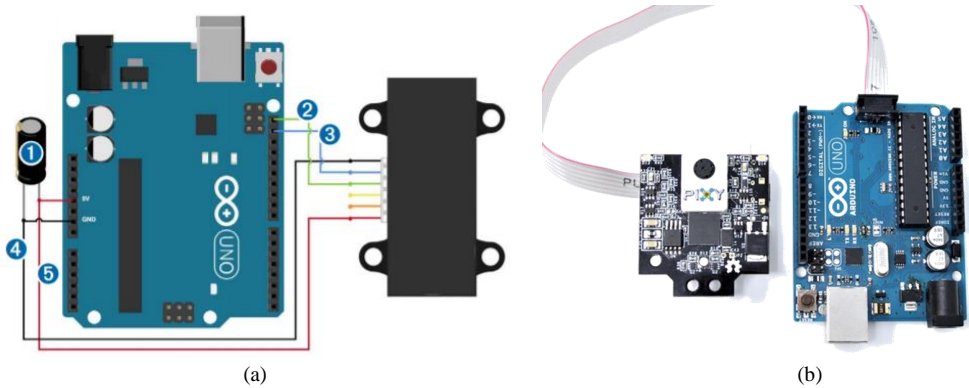


Figure 6. (a) Arduino I2C Wiring (b) Connection with Pixy Camera

On power-up or reset, LiDAR Lite performs a self-test sequence and initializes all registers with default values. After roughly 22ms distance measurements can be taken with the I2C interface. The enable pin uses an internal pullup resistor and can be driven low to shut off power to the device. This device has a 2-wire, I2C-compatible serial interface (refer to I2C-Bus Specification). It can be connected to an I2C bus as a slave device under the control of an I2C master device. It supports 400 kHz Fast Mode data transfer.

The I2C bus operates internally at 3.3V. An internal level shifter allows the bus to run at a maximum of 5 V. Internal 3k $\Omega$  pullup resistors ensure this functionality and allow for a simple connection to the I2C host. The device has a 7-bit slave address with a default value of 0x62. The effective 8-bit I2C address 0xC5 read. The sensor will not respond to a general call. Support is not provided for 10-bit addressing. Setting the most significant bit of the I2C address byte to one trigger automatic incrementing of the register address with successive reads or writes within an I2C block transfer. This is commonly used to read the two bytes of a 16-bit value with one transfer.

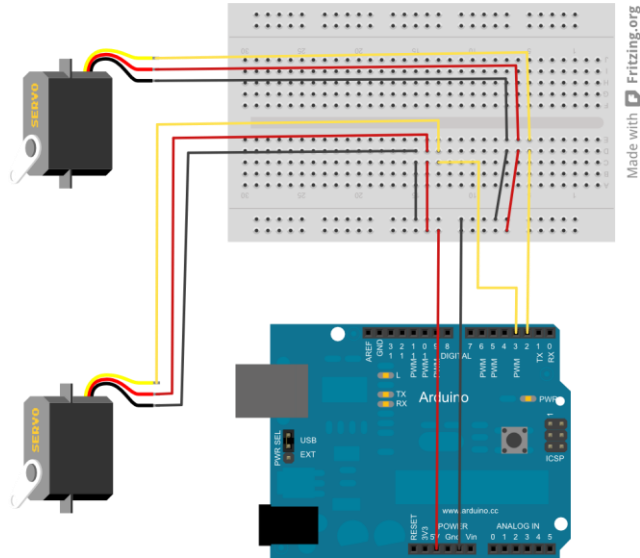


Figure 7. Schematic of servo connection for pan-tilt mechanism

Next, the connection of Pixy camera to Arduino microcontroller is shown in Figure 6b, which is also connected to servo motor in Figure 7 for pan-tilt mechanism shown in Figure 2 so that the camera is able to track the object within its field-of-view and with two degrees of freedom movement. Suppose that the x-axis is pointed towards the direction of the camera, y-axis towards the right-side of the camera when we look at it from the back, and z-axis towards the center of the Earth, then the two directions for which the pixy camera can rotate is pitch and yaw direction. The Arduino API for pan-tilt mechanism developed by CMUcam5 Pixy is used in this experiment. It uses argument from 0 to 1000 for each servo motor as a rotation command, instead of actual angular quantity. During the calibration, we found that the mechanism shown produces side constraint for which the pitch movement is in the range between  $-47$  to  $38.5$  degrees, while yaw ranges from  $25$  to  $151.5$  degrees from its axes with the accuracy of  $\pm 1$  cm. However, when the lidar lite mounted on this mechanism, the side constraint changes. The discussion for this will be addressed shortly after the design on how the lidar lite and the pixy camera is joined as well as the calibration in terms of servo command from Arduino API and actual angular.

While the servo motors primarily serve to support the required movement of pixy camera to follow detected object, it also has to have sufficient power to support heavier hardware, that is, when the pixy camera is physically connected or integrated to LiDAR as shown in Figure 8, with maintaining the capability of the speed of the camera with the movement of the object. However, it is important to note that the choice of how it is mounted affects the robustness of object ranging. In a way that, due to the fact that the distance measurement from LiDAR is dependent on the pixy camera, the further away of the LiDAR position from the pixy camera, the less robust it can measure the intended object. Stated

differently, we want to place the LiDAR and camera as close as possible without reducing the field-of-view of the camera so that it can be applied to small objects.

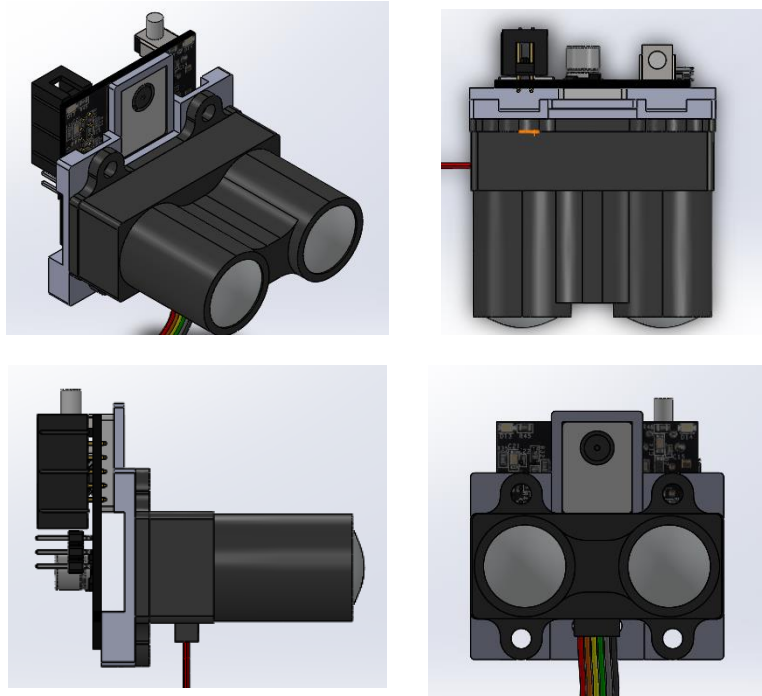


Figure 8. Connection for LiDAR and Pixy2 CMUcam5

One hypothetical example is that suppose we want to measure the distance and hence determine the position of a cube up in the air (we can also imagine a very small UAV with this shape) with dimension 5 unit-length using LiDAR. If the pixy camera detect this object and shown in the center of the image, and while the LiDAR center of mass is placed, say 5 unit-length or more from the center of the camera, then the LiDAR will not reflect its signal to the surface of the cube and instead will go through it until it is reflected by unwanted object, for example wall or trees. However, this will not likely the case for this experiment considering the LiDAR Lite beam divergence of 8 mili-radian and the CMUcam5 field-of-view, the two (solid) angles generated by these two devices will likely overlap in a distance of less than about 20 cm from its departing point.

Now, when the Lidar Lite attached to the pixy camera, additional constraints occur especially in tilt-mechanism due to additional weights that affects how servo rotates. Unlike stepper motor, servo motor does not work with a fixed step of angle but depends on the signal received. It is shown that the new range of pan-tilt angle is from 25 to 151.5 deg and from -23.5 to 38.5 deg with measurement accuracy of  $\pm 1$  deg. Moreover, the attachment of the lidar to the pan-tilt mechanism changes the angle produced by

the same servo command in Arduino API as shown in Figure 9 for pan angle and Figure 10 for tilt angle. The uncalibrated refers to the linear assumption between servo command versus the angular movement. In Figure 9, it appears that the linear assumption can be used. However, due to the fact that it is crucial for the code to read correct angular position, the calibration to fit the actual measurement is conducted by using surrogate model with radial basis function method. The training data points is taken from 70% of the actual measurements, while 30% of the observation data is used for validation to avoid overfitting. This calibration is inevitably important for tilt angle because the relationship between the servo command is not linear from the angle generated by the servo motor as depicted in Figure 10. It is important to also point out that the servo command 0 for pan angle begins from 25 deg, which is the minimum angle that the pixy cam can reach to detect the object, while the servo command 0 for the tilt is from its maximum angle, that is, 38.5 deg. Notice that in Figure 10, the maximum servo command is not 1000. Instead, the servo command 575 is used as a maximum value to produces the angle of -23.5 deg in tilt angle based on experimental study. The servo command for pitch direction greater that this number will result in the same angle due to mechanical constraint between the LiDAR orientation and the base of the pan-tilt mechanism.

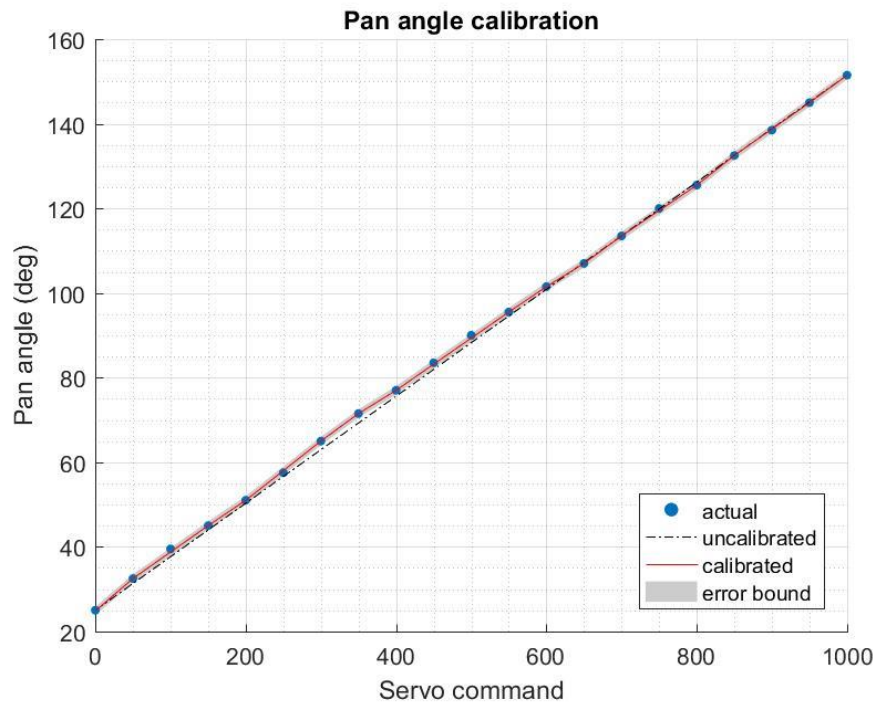


Figure 9. Pan angle calibration due to the attachment of LiDAR Lite

The resulting surrogate model constructed from the observation data is mathematically expressed as follows:

$$\hat{f}(x) = \sum_{i=1}^N w_i \times (\|x - x_{\text{ref},i}\|^2 + 1)^{1/2} \quad (1)$$

Where  $\hat{f}$  is the estimated angular position,  $i$  is index,  $w_i$  is the corresponding weights,  $x_{\text{ref},i}$  is the  $i$ -th reference point,  $N$  is the maximum index for each model, and  $x$  is the Arduino servo command. Table 2 and Table 3 provide the corresponding parameters.

Table 2. Model parameters for pan angle measurement

Index	1	2	3	4	5	6	7	8
ref. points	0	50	150	200	300	350	400	550
weights	0.1649	-0.0143	-0.0026	0.0102	-0.005	-0.0102	0.0069	-0.0017
Index	9	10	11	12	13	14	15	
ref. points	600	650	700	800	850	950	1000	
weights	-0.0052	0.0103	-0.0052	0.0103	-0.0077	0.0023	0.0234	

The surrogate model is provided in Figure 9 and Figure 10 for which is considered as calibrated measurement with error bound  $\pm 1$  cm. Another important implication in this is that because the servo command is from 0 to 1000 that produces angle from 25 to 151.5 deg in yaw direction for example, the resolution of the pan angle can be considered to be 0.13 deg. In similar fashion, the resolution for tilt angle is approximately 0.11 deg. The mean errors of this surrogate model when evaluated at the observation points are 0.14 deg with standard deviation of 0.25 deg for pan angle and 0.22 deg with standard deviation of 0.38 deg for tilt angle.

Table 3. Model parameters for tilt angle measurement

Index	1	2	3	4	5	6	7	8	9
ref. points	0	50	75	100	125	175	200	250	325
weights	-0.0322	-0.0156	0.0216	-0.0217	0.021	0	-0.0103	-0.0035	0.0351
Index	10	11	12	13	14	15	16	17	18
ref. points	350	375	397	400	425	500	525	550	575
weights	-0.0422	-0.0084	0.0292	0.0163	-0.0536	0.0175	0.0003	-0.0334	0.1057

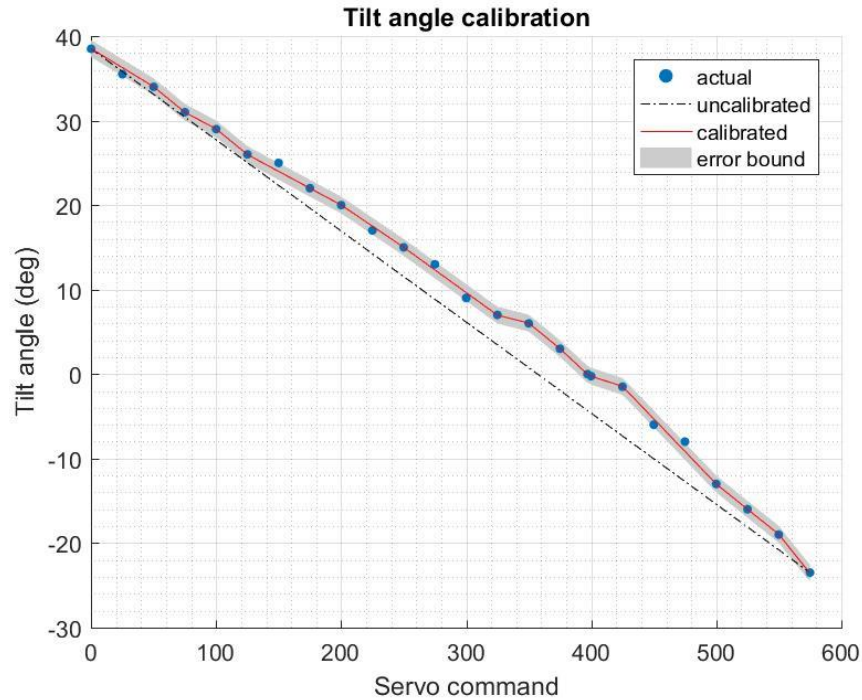


Figure 10. Tilt angle calibration due to the attachment of LiDAR Lite

In practice, using publicly available Arduino API developed by CMUcam5 developers, the pan-tilt mechanism measures the servo command given a detected object to keep track that the detected object is located near the center of the camera view, if not exactly at the center. The surrogate model will then translate this servo command for each direction into angular positions with measured errors. Providing that the detected object is in the direction of the lidar laser beam, the distance of the detected object is then measured by LiDAR Lite V3 so that we finally obtain the position of the object in 3D, by transforming these three measurements (pan angle, yaw angle, and distance) to Cartesian coordinates.

In addition to calibrating the measurement from the servo motor, we also need to examine the distance measurement from the lidar. The result of this examination is shown in Figure 11. It is shown that the nonlinearity is present when the distance is less than 50 cm, which is depicted by enormously wide confidence interval as well as instability in the mean offset. In general, the measurement from lidar overestimates the actual distances, and it gradually increases as the distance increases in approximately linear manner in the mean offset. This also applies to the corresponding confidence interval for distance greater than 50 cm. Therefore, the calibration for LiDAR measurement is based on offset-modeling and its 95% confidence interval in Figure 11. The offset model is obtained from constructing linear surrogate model that learns the offset nature from the observation data. The modeling is based on splitting the observation data into training and validation set by 70% and 30% respectively. The model

produces the coefficient of determination,  $R^2$ , of 0.9107 and 0.9166 in the training and validation points, which indicates that the model fit the observation data well.

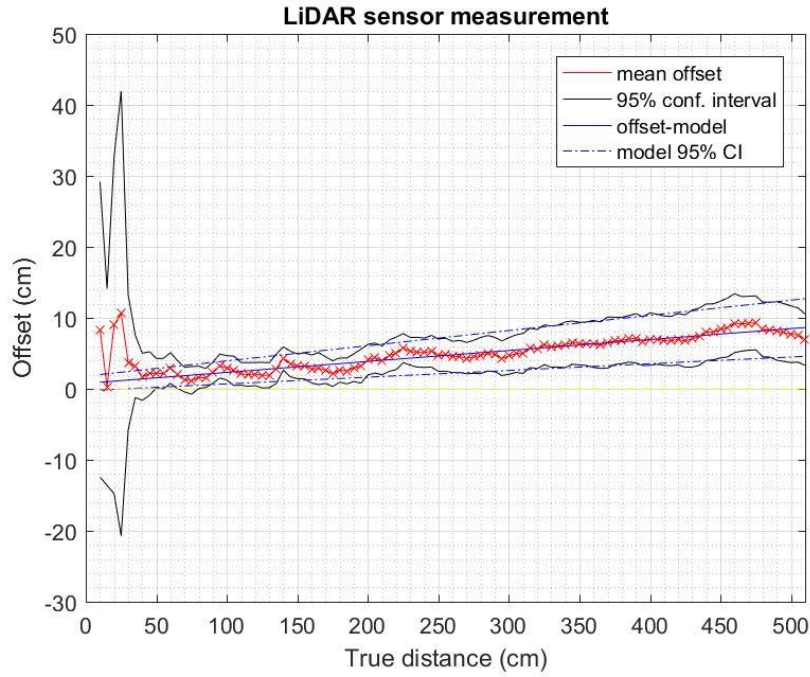


Figure 11. LiDAR sensor measurement performance

In addition to the offset model that basically takes the data from the mean offset, the standard error of the lidar measurement is also calibrated, as it linearly increases as the distance increase. The 95% confidence interval model in Figure 10 has  $R^2$  of 0.9288 and 0.9536 in training and validation dataset which implies a good fit. An important note in this process is that, the model does not consider the linearity in the lidar measurement behavior for the distance less than 50 cm because it may affect the goodness-of-fit for other distances, as its confidence interval is significantly large compared to the other distances. However, in practice, this nonlinearity can be neglected since in most condition, the distance of the detected object from the lidar is much greater than this threshold of nonlinearity. The parameters of the offset model as well as the confidence interval are shown as follows:

$$\widehat{\text{Offset}}(\text{distance}) = 0.7693 + 0.0154 \times \text{distance} \quad (2)$$

and the corresponding standard error is computed as follows

$$\widehat{SE}_{\text{offset}}(\text{distance}) = 0.5255 + 0.003 \times \text{distance} \quad (3)$$

### 3.2. Experimental setup

Another term of the application of LiDAR for distance measurement is called laser rangefinder. It is a rangefinder that uses a laser beam to determine the distance to an object. The most common form of laser rangefinder operates on the time of flight principle by sending a laser pulse in a narrow beam towards the object and measuring the time taken by the pulse to be reflected off the target and returned to the sender. Despite the high speed of light, this technique is not appropriate for high precision sub-millimeter measurements, where triangulation and other techniques are often used. The pulse may be coded to reduce the chance that the rangefinder can be jammed. It is possible to use Doppler effect techniques to judge whether the object is moving towards or away from the rangefinder. The precision of the instrument is determined by the rise or fall time of the laser pulse and the speed of the receiver. One that uses very sharp laser pulses and has a very fast detector can range an object to within a few millimeters.

The beam will eventually spread over long distances due to scintillation and the beam wander effects, caused by the presence of air bubbles in the air acting as lenses ranging in size from microscopic to roughly half the height of the laser beam's path above the earth. This atmospheric distortion coupled with the divergence of the laser itself and with transverse winds that serve to push the atmospheric heat bubbles laterally can combine to make it difficult to get an accurate reading of the distance of an object, say, beneath some trees or behind bushes. Some of the laser light might reflect off leaves or branches which are closer than the object, giving an early return and a reading which is too low.

The experiment is conducted in indoor environment with the size of  $4.5 \times 4.3 \times 2.7$  meters. Figure 12 shows the experimental setup environment to tracking the UAV. The black line in the figure represents the limit for which the UAV will be flown with the size of  $3.6 \times 2.7$  meters. The Two identical tracking systems are used to measure the distance of the UAV and its orientation relative to each system, allowing us to conduct result comparisons between the two after they are expressed in the same reference frame. Different scenarios will be conducted to simulate the robustness of the algorithm as well as examine the capability of the pixy camera to follow through the UAV. In this experiment, the UAV is treated as point mass because the information about its orientation is not available from both accelerometer and gyroscope.

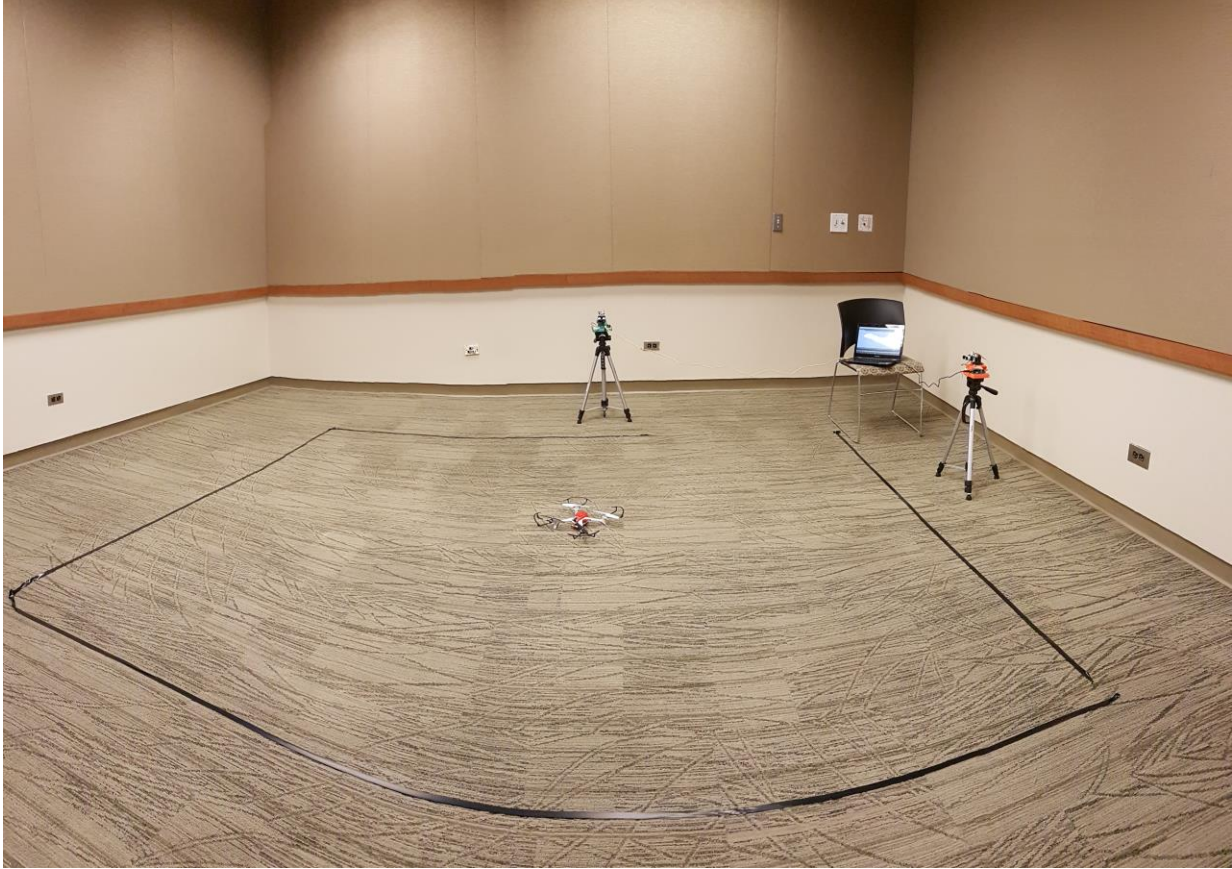


Figure 12. Experimental setup environment for this work

The UAV we use for this experiment is Sky Viper Journey GPS Drone and shown in Figure 13. The choice is based on its dimension that fits with the integration we chose for the LiDAR and pixy camera as well as the room size of the experiment. In addition, in order to provide both color support and good reflection for the tracking system, a red-color retroreflector material is coated on the UAV.



Figure 13. The UAV used in the experiment (dimension: 13.28 inches (L)  $\times$  13.28 inches (W)  $\times$  2.21 inches (H))

Furthermore, the housing for Arduino, breadboard, and the pan-tilt mechanism is designed so that it acts as tracking system station. Figure 14 shows the housing design and its implementation on the experiment. The housing is attached to a tripod, and in this experiment, we use two identical tripods with two tracking system. The height of the tracking station from the ground is 63 cm. The use of tripod is useful to compensate the maximum tilt angle the pixy camera can achieve, that is 38.5 deg, because the most important application of this work is that tracking flying objects, which in general operates in high altitude.

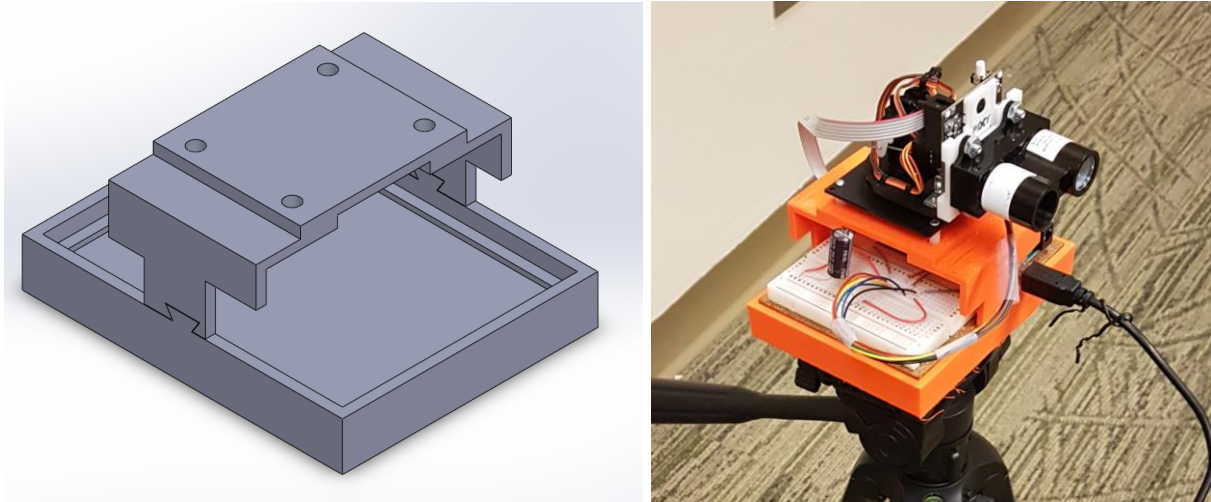


Figure 14. Tracking system housing and station

### 3.3. Kalman Filter and error analysis

Most of the times we have to use a processing unit such as an Arduino board, Raspberry Pi, or Beaglebone, a microcontroller chip plays a role to process the sensor data and produce the corresponding outputs. The sensor data contains noise. This results in considerable error for calculations and outputs when these noises are used. For robotic systems, noisy readings are not suitable as they will not work well or will not produce the desired task. The need to filter the noise and stabilize the readings of every sensor before using it for applications is therefore a crucial task. One common approach to achieve this is by using Kalman Filter.

Kalman Filter first introduced by Rudolph E. Kalman in 1960. An attempt at a clear and simple explanation is made here. Readers looking for more information are encouraged to see Simon's *Optimal State Estimation* [27]. The Kalman Filter can be broken down into two main portions: prediction and update. In the prediction phase, a model of the state dynamics is used to propagate the state forward through time. In the update phase, the KF receives a measurement that is in some way related to the state; this allows the KF to calculate the state that would have made the measurement possible. Based on

the covariances of the propagated state estimate and the measurement, the KF assigns weights to each and calculates the most probable state, a weighted average of the two.

The prediction phase is driven by the dynamics of the model, or how the state changes with time. This relationship is used to solve for the state transition matrix,  $\Phi(t, t_o)$ , which in turn is used to propagate both the state and the state covariance matrices forward. The dynamics model used in the author's code, shown below, closely follows the equations found in Li [28] with the exception of a few changes. First, we define our state vector. The measurements from the sensors provide distance and two angular positions. Therefore, the state vector used can be these three variables, or the 3D positions in the Cartesian coordinates so that the dynamics equations in the process is in this coordinate system, whichever is practical. In this calculation, we will use Cartesian coordinates so that in the Kalman filtering, the predict and updated is readily in position term. The dynamics of the state vector can be written as

$$\frac{d}{dt}X(t) = F(t) \cdot X(t) + G(t) \cdot \bar{\epsilon}(t) \quad (4)$$

Where this equation, first-order system of differential equations has a solution of the form

$$x(t) = \Phi(t, t_o) \cdot x(t_o) + \int_{t_o}^t \Phi(t, t') \cdot G(t') \cdot x(t') dt' \quad (5)$$

where  $\Phi$  is the state transition matrix and satisfies

$$\frac{d}{dt}\Phi(t, t') = F(t) \cdot \Phi(t, t') \quad (6)$$

which has the solution

$$\begin{aligned} \Phi(t, t') &= e^{F(t-t')} \\ &= I + F(t-t') + \frac{1}{2!}(F(t-t'))^2 + \frac{1}{3!}(F(t-t'))^3 + \dots \\ &= \sum_{n=0}^{\infty} \frac{(F(t-t'))^n}{n!} \end{aligned} \quad (7)$$

$\Phi_{k,k-1}$  propagates the state forward according to the following:

$$X_k = \Phi_{k,k-1}X_{k-1} + w_k \quad (8)$$

where

$$w_k \sim (0, Q_k) \quad (9)$$

with the state transition matrix defined, we now move on to describe the update phase, in which a measurement is received. In this case, the measurement is defined as the difference between the position derived from the sensor measurements and actual position, for known determined trajectory. As an example of how this relationship is defined, Li [28] provides similar derivation process for which this experiment is based on. However, in the paper, the derivation is demonstrated in most applications of the Kalman filter that is the position and velocity derived from INS data versus what is received from the GPS data with known measurement error and thus, can estimate the process error in the Kalman filter. In this work, the measurement and process matrix are approximated based on the calibration conducted in Section 3.1. The covariance matrix of these initial estimates becomes  $\hat{P}(t_o)$ . Therefore, we can implement EKF according to the following algorithm:

### Predict

$$\begin{aligned}\hat{X}_k^- &= \Phi_{k,k-1} \hat{X}_{k-1}^+ \\ \hat{P}_k^- &= \Phi_{k,k-1} \hat{P}_{k-1}^+ \Phi_{k,k-1}^T + G_{k-1} Q_{k-1} G_{k-1}^T\end{aligned}\quad (10)$$

### Update

$$\begin{aligned}K_k &= \hat{P}_k^- H_k^T (H_k \hat{P}_k^- H_k^T + R_k)^{-1} \\ \hat{X}_k^+ &= \hat{X}_k^- + K_k (z_k - H_k \hat{X}_k^-) \\ P_k &= (I - K_k H_k) \hat{P}_k^-\end{aligned}\quad (11)$$

## 4. Results and Discussion

### 4.1. Real-time detection and ranging results

Several simulation tests have been carried out in order to verify the performance of the proposed algorithm. We here present results demonstrating the consistency of the multi-state constraint filter. In Figure 15a, the estimated trajectory of the UAV for a typical simulation run is plotted. For this particular experiment, the UAV is moving in a circular trajectory of radius 1 m. In Figure 15b-d, the estimation errors for the UAV position, orientation, and velocity are shown (blue lines), for 10 trials under this setup. These errors are compared against the corresponding  $\pm 3\sigma$  bounds computed using the estimated covariance (the red enveloping lines). From these plots it is clear that the errors are commensurate with the computed covariance, and thus the estimator is consistent. Although the results shown in Figure 15 pertain to this particular simulation setup, the results are typical; through numerous simulations we have verified that the estimates produced by the multiple-state constraint filter are

generally consistent. Given the severe nonlinearities that arise in vision-based estimation, this is a very significant property of the algorithm.

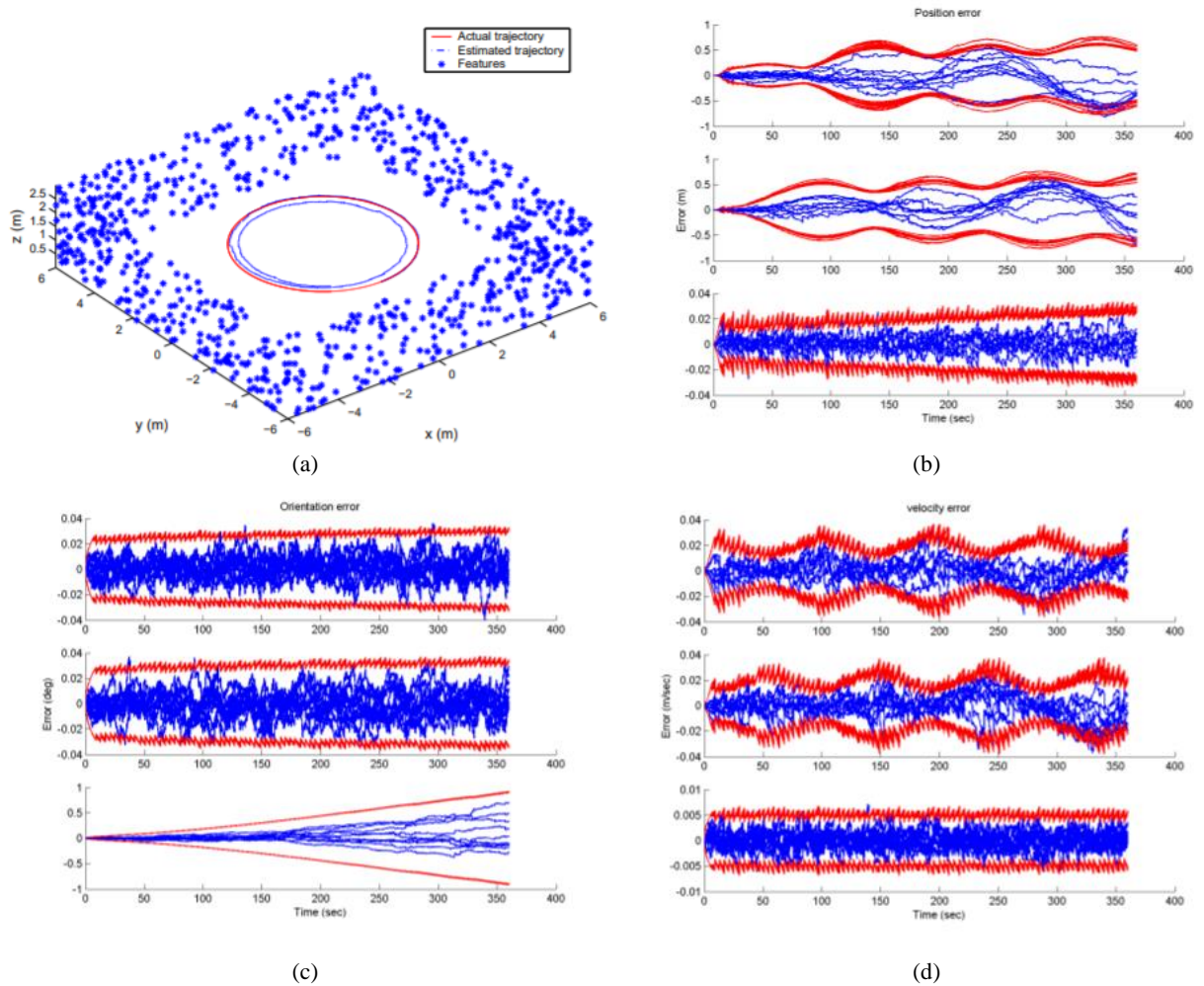


Figure 15. (a) The trajectory used for the experiments. The solid red line denotes the actual trajectory, the dash-dotted blue line represents the estimated trajectory for a single trial, while the feature positions are shown with asterisks. (b-d) The position, attitude, and velocity errors (blue lines), respectively, and the  $\pm 3\sigma$  bounds (red enveloping lines) for 10 trials.

In the next experiment, the UAV was manually operated with overall of 42m-long trajectory. The zero-altitude reference is taken from the position of tracking system housing base. The estimated trajectory of the UAV can be seen in Figure 16. In this plot, the initial position of the UAV is denoted by a red square, while the final position estimate is denoted by a star. Even though the ground truth for the entire duration of the experiment is not available, it is known that during this motion the system was returned to its initial position,  $\mathbf{p}_{init} = [0 \ 0 \ 0]^T$ , twice; once at  $t = 220$  sec, and once at the end of the

trajectory. At these two time instants the position estimates are equal to  $\hat{\mathbf{p}}_1 = [-0.12 \ 0.2 \ -0.2]^T$  m and  $\hat{\mathbf{p}}_{\text{final}} = [-0.10 \ 0.36 \ -0.3]^T$  m, respectively. It is important to note that these estimates correspond to errors smaller than 0.5% of the travelled distance.

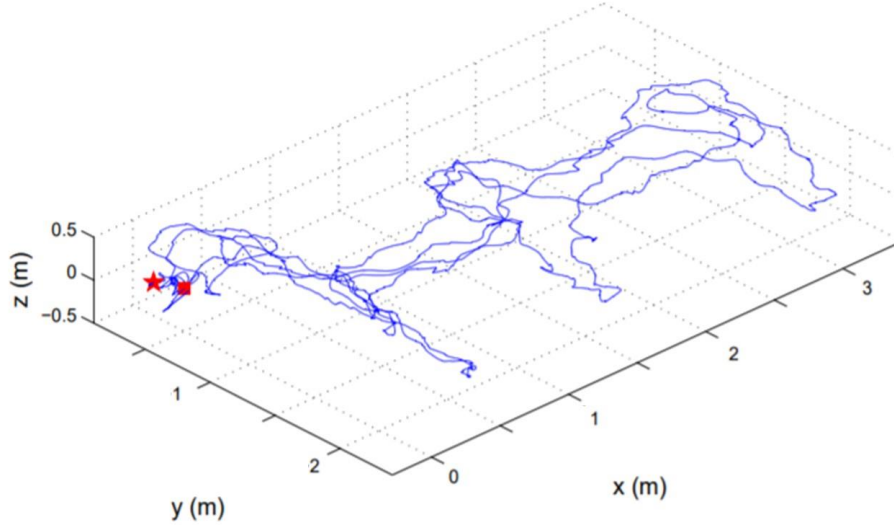


Figure 16. The estimated 3D trajectory of the UAV. The initial position is denoted by a red square, while the final one is denoted by a red star

## 4.2. Discussions and limitations

The pan-tilt mechanism design plays an important role in determining the accessibility of the machine-vision camera to track an object. In addition, it is important that the choice of motor type accounts for the required power to being able to carry out the sensor and motion of the camera as the object moves in certain speed and direction. In this work, the pan-tilt system designed was able to bring two degree of freedom, yaw and pitch direction. However, it limits to certain magnitude of rotation; pitch ranges from -23.5 to 38.5 degrees, while yaw ranges from 25 to 151.5 degrees from its reference axis. This situation narrows down the options of the placement of the sensor, and the nearest and maximum height of the UAV it is able to track.

Another limitation from present work came from the capability of the vision camera to recognize the object of interest, e.g., UAV. Due to the fact that color-based filtering vision camera is used, the UAVs have to be assigned distinctive colors that is dominant in the environment and easily detected by the camera. Since the experiment is limited to closed environment, that is, situated such that the color assigned in the UAV is distinctive, future work should examine the suitability of the current algorithm that the tracking device is readily application in an open environment, with different intensity source of light that will affect the reflection color of the object and with terrain such as building and trees.

Furthermore, the focal length of the pixy camera is 3mm, which affects the limited distance of the camera detects the object. The smaller the focal length, the more the constraint in capturing distant object. In this experiment, the pixy camera is only able to detect the UAV up to approximately 6 meters from the lidar sensor. Because of the short focal length, the object within this distance is shown as a small point mass on screen and consequently the pixy camera is hard to detect the object based on color. This issue can be addressed in future in several ways: one would consider to mount external device to enhance the camera performance in terms of capturing distant object; attach active LED light to the object to provide some sort of blinking diode so that it increase the color range in the object and thus, would increase the limit of the distance for which the pixy camera started to loose track. In the former, the pixy camera is widely used to track close-range object in robotics, and therefore the issue we have would not likely be experienced by such applications. This also would be a potential area for which the vision-camera developers could considers, that is, develop the low-cost camera with long focal length. For the latter, in this experiment, attaching LED light is not possible due to payload that the UAV used in this experiment can carry. For future application or in practice, however, this method is feasible to be implemented, since outdoor UAVs that are widely used for practical purposes such as topography mapping, surveillance, or agriculture, are most likely capable of carrying active LED light.

## **5. Conclusion**

The potential applications of UAVs have sparked many research topic areas for the past two decades, including remote sensing, dynamics system and control, and computer science. In this work, we establish pan-tilt mechanism to support the integration of ground-based tracking system, which consists of color-based machine-vision camera for object detection and LiDAR for object ranging. The support mechanism provides two degrees of freedom that allows the sensor to rotate in pitch and yaw directions, tracking the object of interest to a certain degree, that is, with side constraints in the two directions. Furthermore, the refinement to improve the precision of the real-time measurement from LiDAR is also considered using an Extended Kalman Filter (EKF)-based algorithm. The experimental results have proved the effectiveness of this method. It is shown that the estimation of error sources inherent in this sensor significantly improves the accuracy of these measurements.

The error accuracy from the pan-tilt mechanism are 0.14 deg with standard deviation of 0.25 deg for pan angle and 0.22 deg with standard deviation of 0.38 deg for tilt angle. The measurement from lidar is in general overestimates the true value, for which the offset is linearly modeled. The measurement from UAV tracking provided approximately 15 cm accuracy in altitude, and about 5 cm

accuracy in horizontal position. Some limitations apply at which this tracking system applies regarding its maximum range due to limitation on built-in focal length of the pixy camera.

## References

- [1] L. Johnson, S. Herwitz, S. Dunagan, B. Lobitz, D. Sullivan, and R. Slye, "Collection of ultra high spatial and spectral resolution image data over California vineyards with a small UAV," in *Proceedings of the 30th International Symposium on Remote Sensing of Environment*, 2003, vol. 20, no. 845-849: International Society of Photogrammetry and Remote Sensing.
- [2] S. Nebiker, A. Annen, M. Scherrer, and D. Oesch, "A light-weight multispectral sensor for micro UAV—Opportunities for very high resolution airborne remote sensing," *The International Archives of The Photogrammetry, Remote Sensing Spatial Information Sciences*, vol. 37, no. B1, pp. 1193-1199, 2008.
- [3] M. Uysal, A. Toprak, and N. Polat, "DEM generation with UAV Photogrammetry and accuracy analysis in Sahitler hill," *Journal of Measurement*, vol. 73, pp. 539-543, 2015.
- [4] U. Niethammer, M. James, S. Rothmund, J. Travelletti, and M. Joswig, "UAV-based remote sensing of the Super-Sauze landslide: Evaluation and results," *Journal of Engineering Geology*, vol. 128, pp. 2-11, 2012.
- [5] H. Xiang and L. J. B. e. Tian, "Development of a low-cost agricultural remote sensing system based on an autonomous unmanned aerial vehicle (UAV)," vol. 108, no. 2, pp. 174-190, 2011.
- [6] M. Bartholmai and P. Neumann, "Micro-drone for gas measurement in hazardous scenarios via remote sensing," in *Proceedings of the 6th WSEAS International conference on remote sensing*, 2010.
- [7] A. Cesetti, E. Frontoni, A. Mancini, P. Zingaretti, and S. Longhi, "A vision-based guidance system for UAV navigation and safe landing using natural landmarks," *Journal of Intelligent Robotic Systems*, vol. 57, no. 1-4, p. 233, 2010.
- [8] T. Sankey, J. Donager, J. McVay, and J. B. Sankey, "UAV lidar and hyperspectral fusion for forest monitoring in the southwestern USA," *Journal of Remote Sensing of Environment*, vol. 195, pp. 30-43, 2017.
- [9] L. Wallace, A. Lucieer, C. Watson, and D. Turner, "Development of a UAV-LiDAR system with application to forest inventory," *Journal of Remote Sensing*, vol. 4, no. 6, pp. 1519-1543, 2012.
- [10] J. Schipperijn, J. Kerr, S. Duncan, T. Madsen, C. D. Klinker, and J. Troelsen, "Dynamic accuracy of GPS receivers for use in health research: a novel method to assess GPS accuracy in real-world settings," *Journal Frontiers in public health*, vol. 2, p. 21, 2014.
- [11] n. G. Ruffini, F. Soulat, M. Caparrini, O. Germain, and M. Martin-Neira, "The Eddy Experiment: Accurate GNSS-R ocean altimetry from low altitude aircraft," *J Geophysical research letters*, vol. 31, no. 12, 2004.
- [12] D. M. Gavrila and V. Philomin, "Real-time object detection for" smart" vehicles," in *Proceedings of the Seventh IEEE International Conference on Computer Vision*, 1999, vol. 1, pp. 87-93: IEEE.
- [13] M. Ahmad, H. Rong, S. Alhady, W. Rahiman, and W. Othman, "Colour tracking technique by using Pixy CMUcam5 for wheelchair luggage follower," in *2017 7th IEEE International Conference on Control System, Computing and Engineering (ICCSC)*, 2017, pp. 186-191: IEEE.
- [14] L. Huang and M. Barth, "Tightly-coupled LIDAR and computer vision integration for vehicle detection," in *2009 IEEE Intelligent Vehicles Symposium*, 2009, pp. 604-609: IEEE.

- [15] L. Huang and M. Barth, "A novel multi-planar LIDAR and computer vision calibration procedure using 2D patterns for automated navigation," in *2009 IEEE Intelligent Vehicles Symposium*, 2009, pp. 117-122: IEEE.
- [16] C. Premebida, O. Ludwig, and U. J. J. o. F. R. Nunes, "LIDAR and vision-based pedestrian detection system," vol. 26, no. 9, pp. 696-711, 2009.
- [17] T. Huang, "Computer vision: Evolution and promise," 1996.
- [18] T. Morris, *Computer Vision and Image Processing (Cornerstones of Computing)*. Palgrave Macmillan Limited, 2004.
- [19] J. Carter *et al.*, "An introduction to LiDAR technology, data, and applications," p. 2, 2012.
- [20] F. Amzajerjian, D. Pierrottet, L. Petway, G. Hines, and V. Roback, "Lidar systems for precision navigation and safe landing on planetary bodies," in *International Symposium on Photoelectronic Detection and Imaging 2011: Laser Sensing and Imaging; and Biological and Medical Applications of Photonics Sensing and Imaging*, 2011, vol. 8192, p. 819202: International Society for Optics and Photonics.
- [21] I. E. Paromtchik, "Optical guidance method for robots capable of vision and communication," *Journal of Robotics and Autonomous Systems*, vol. 54, no. 6, pp. 461-471, 2006.
- [22] L. N. Glaros, W. B. Eddins, C. Gaylord, and M. Kincheloe, "Spot leading target laser guidance for engaging moving targets," ed: Google Patents, 2012.
- [23] G. Ltd., "Lidar Lite v3 Operation Manual and Technical Specifications," Available: [https://static.garmin.com/pumac/LIDAR\\_Lite\\_v3\\_Operation\\_Manual\\_and\\_Technical\\_Specifications.pdf](https://static.garmin.com/pumac/LIDAR_Lite_v3_Operation_Manual_and_Technical_Specifications.pdf)
- [24] S. Heath, *Embedded systems design*. Elsevier, 2002.
- [25] M. Banzi and M. Shiloh, *Getting started with Arduino: the open source electronics prototyping platform*. Maker Media, Inc., 2014.
- [26] M. Margolis, *Arduino Cookbook: Recipes to Begin, Expand, and Enhance Your Projects*. O'Reilly Media, Inc., 2011.
- [27] D. Simon, *Optimal state estimation: Kalman, H infinity, and nonlinear approaches*. John Wiley & Sons, 2006.
- [28] X. Li, "Moving base INS/GPS vector gravimetry on a land vehicle," The Ohio State University, 2007.