

# BuddyCam

Joseph Cao, CSE, Steven Gurney, CSE, Saswati Swain, EE, and Kyle Wright, CSE

**Abstract — Unmanned Aircraft Systems (UAS) incorporate an Unmanned Aircraft Vehicle (UAV or drone) and a system controller while providing a means of communication between the two. UAS are rising in popularity and have helped to provide useful information in monitoring, security, and search and rescue. One main area of use is law enforcement, in which the UAS provides tactical surveillance, subject tracking, and assistance in investigation. BuddyCam is a deployable UAS capable of autonomously identifying, tracking and recording law enforcement officers in high stress situations. The system consists of a quadcopter equipped with two fixed cameras for video capture, an onboard Raspberry Pi 3 to perform real time image processing using computer vision, an IR tracker to assist with tracking accuracy, and a GPS client/server system to give coarse location. By combining the location data provided by each subsystem we attain the location of the subject and adjust the drone's position.**

## I. INTRODUCTION

THERE is growing concern surrounding the relationship between law enforcement officers and the public. Events such as riots, shooting of unarmed civilians, and violence against officers have created challenges in evaluating officer performances due to lack of information and reliable video evidence. A review of cases shows wrongful accusations against officers as well as citizens and officers alike often having trouble remembering important details after an adrenaline fueled event.

A case revolving around the topic of life and death for a police officer and the public involved a couple, Mendez and Garcia, that were expecting a child. They were just a step above homelessness, living in a rat infested shack. One day deputies came to the property searching for a man who had violated the terms of a parole. Not knowing it was police, Mendez picked up a BB gun and began to rise. The deputies opened fire, hitting Mendez 14 times and Garcia once in her back. The couple sued L.A. County for violation of their Fourth Amendment rights [1]. Legal practitioners are adopting a critical eye towards forensic evidence, and the role they play in convictions. Some shortfalls in the current system include operations problems, reliability tests, and bias of legal representatives [2]. One of the challenges in court cases include the reliability factor of the court admitted evidence. The lack of comparable data or ground truth available for the public results in the court having to drop cases, or side in favor of the party at fault. [2] This issue can be summed up by Edmond et al. [3]:

*The absence of a database or some other credible method of assigning significance to purported similarities means the*

*observer has no reasonable basis on which to draw conclusions about identity.*

Video surveillance has been used for the past couple years as a means of preventing crime. [4] Court cases including video surveillance, must ascertain how the video was recorded, whether the transporting of the video compromised its reliability. Body webcams (BWCs) are a new and growing addition for police-citizen encounters. Through the use of BWCs, there was a 68 percent decrease in use-of-force complaints. It is unknown whether the decline was due to the use of the recording technology, but such a drop is most likely due to the fact the false complaints were preempted due to the presence of live recording [5]. Not all BWC video footage will be used in every case, or determine the outcome of a case, although the video will provide circumstantial evidence. Despite the low-light, barely visible video, audio from the BWC provide courts with sufficient evidence.

Tackling this concern of evidence reliability and growing concern of false accusations against officers we provide a solution of BuddyCam. BuddyCam, is a Deployable Unmanned Aerial System (UAS) capable of autonomously identifying, tracking and recording officers. This quadcopter is equipped with a camera that provides an aerial video capture of the officer and transmits the video to an onboard Raspberry Pi. The Pi performs object identification and tracking through the use of computer vision. This tracking is supplemented by an Infrared (IR) Beacon to isolate the officer from other individuals, as well as GPS for coarse location data. Flight instructions are determined using these systems, and mapped to the flight controller to move the UAS. While this system provides valuable real-time information, the video footage is readily available for officers and law enforcement superiors, improving situational awareness.

To begin the design of the UAS systems, we analyzed current limitations which included: limited visibility, shaky footage, and biased footage due to first person perspective of BWCs. Our solution UAS will be fully autonomous after lift-off, enabling the officer to focus on the situation at hand. The UAS will be able to track and keep the subject in the middle of the video frame, with a maximum of 1.6 seconds out of frame, this was calculated from the latency of transferring the data back and forth through the Raspberry Pi that performs the real time image processing using computer vision. The UAS will maintain a minimum height of 10 feet, which prevents it from interfering with the officers on scene, and maintain a line of sight of subject within a radial distance of 15 feet. This allows for unbiased footage and ensuring all

interactions are captured in the recording. The last specification is the UAS being able to operate for at least 10 minutes, which was determined from the UAS abilities itself, and their changes based on funding and power added to the UAS. These specifications are summarized in Table 1.

The UAS will be deployed by the officer, and through the use of system will be able to keep track of the officer as they respond to various situations. Once the officer has completed addressing the situation, the UAS will be turned off and allowed to return to its starting location. Currently, the UAS lifts off from its initial location and lands at the location where it is turned off, which can be different from the initial starting point.

TABLE 1. Specifications

Requirement	Specification
System should be simple and easy to deploy	Fully autonomous after initial lift-off
Operate as to not interfere with officer's duties.	Minimum height of 10 feet
Maintain a view of the officer during deployment.	Radial line of sight, 15 feet from subject Track and keep subject in frame, no less than 1.6 seconds out of frame
Record for as long as any conflict or response would take to be resolved.	Operational time, more than 10 minutes

II. DESIGN

A. Overview

The current BuddyCam system consists of four major subsystems: the UAV, OpenCV image processing, an IR LED beacon, and a GPS client/server application. The UAV subsystem can be further subdivided into the Base UAV and Added Sensing Array systems. The Base UAV consists of bare features included with any UAV. This includes a flight controller, a power distribution and battery system, rotors, and manual controls (pilot remote). On top of these essential UAV features, we will be building an Added Sensing Array that will be housed on the Base UAV. This includes a wide-angle infrared camera and Raspberry Pi 3. These components were chosen as they meet the system requirements of being able to track and record a police officer in an emergency response situation. The wide-angle camera will be able to capture the officer to provide tracking and accommodate a high field of view of the situation. The Raspberry Pi will interface with the cameras, perform the necessary image processing tasks, and

act as a data passthrough for wireless communications.

The wearable tracker subsystem consists of two main components, an array of high-power IR LEDs and a IR capable camera on the sensing array of the drone.

The onboard processing will be performed on the Raspberry Pi which will take in the video feed and sensor data from the wearable tracker and calculate the necessary flight commands to keep the police officer in the center of the frame. Following this brief overview will be a more in-depth discussion of BuddyCam's subsystems.

B. Block 1: IR Beacon

This part of the system consists of an Infrared (IR) Beacon, a portable device that transmits a unique signal to help the UAS with tracking of the officer (Fig. 1A). The IR Beacon will be used simultaneously with the OpenCV identification and tracking. In order to complete this part of the system it is important to understand how IR communication works.

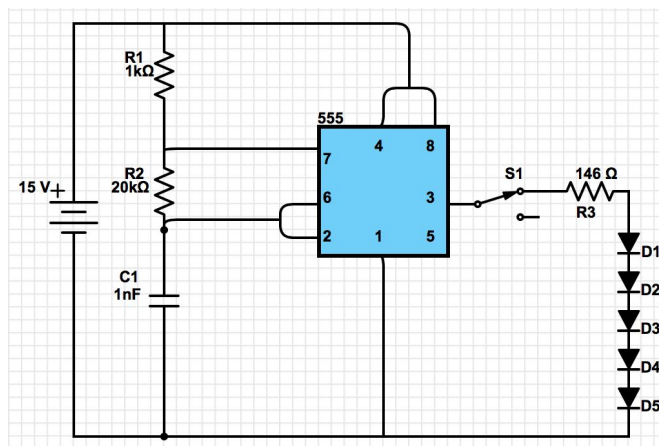


Fig. 1A Flashing LED Schematic. This IR beacon operates at 15 volts using a 555 Timer (IC) Chip, to create a blinking series of IR LEDs when switch is turned on.

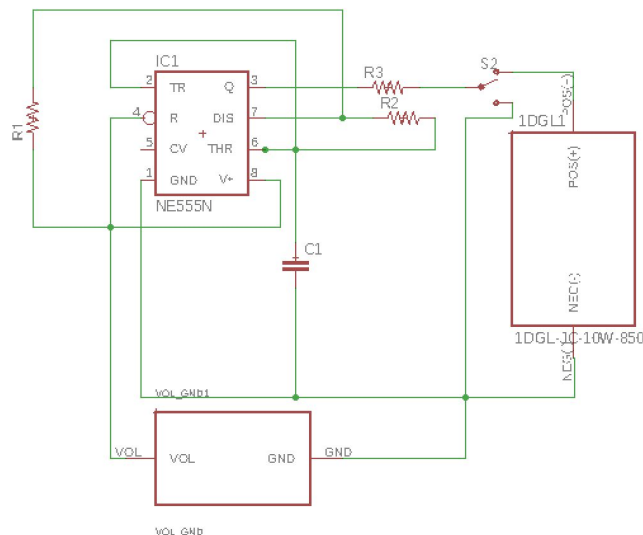


Fig. 1B Flashing LED PCB Schematic on Eagle. This uses a 10W High

power LED unlike the previous design which uses a series of blinking LEDs.

This wireless communication technology is very similar to visible light, except it has slightly longer wavelength [6]. IR radiation is undetectable to the human eye, making it great for wireless communication. IR signals are modulated, patterned, data so it is unique to the receiver. Most IR communication work under 38kHz modulation, but other frequencies can be used as well. When the switch is on, the transmitting IR LED blinks quickly for a fraction of a second to transmit data to the receiving device. The pulse width modulated signal (Fig 2.) can be controlled through a microcontroller, which allows for the waveform to be read by an input pin and decoded as a serial bit stream.

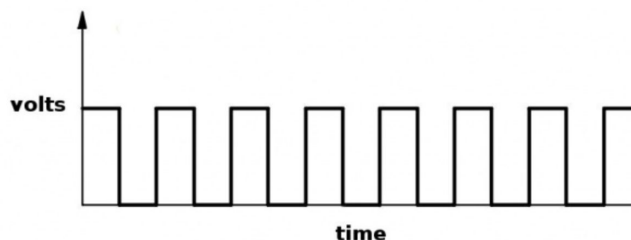


Fig. 2: Pulse width modulated signal (square wave). The output of the IR Beacon continually switches state from high to low without interference from the user. Gives beacon intermittent motion, by switching the IR LEDs between on and off.

The IR beacon will be set up with a 555 timer chip that sends a pulse modulated IR signal at  $\sim 1$  Hz frequency [7]. The IR Beacon will be connected to a 9 volts battery and operate through an on and off switch. The benefits of using an IR Beacon include, detection in the light, signal is unique so multiple officers can use the UAS on scene. The IR Beacon with the 5 series of LEDs (Fig 1A) was not detectable at a distance greater than 10 feet therefore the LEDs were replaced with a 10W LED, shown in (Fig 1B). This new design can be detected with lenses without IR filters from about 10 feet due to high power LED. It allows for a stronger signal, due to the high current in the system,  $\sim 900$  mA, that can be detected at a greater distance. The IR Beacon is portable, attached to the subject during testing and analysis to determine its reliability. The IR beacon was modified several times to determine a design that was the most efficient. The final design of the IR beacon was routed into a PCB design on Eagle (Fig 3A), two of the parts were custom made and added to the directory. The design was routed using two layers, red and blue, to ensure no routes were overlapping and interfering with other connections. The final design is shown on the PCB (Fig 3B), the next steps were to enclose it with plastic, or some material covering which would allow to physical attachment to the subject.

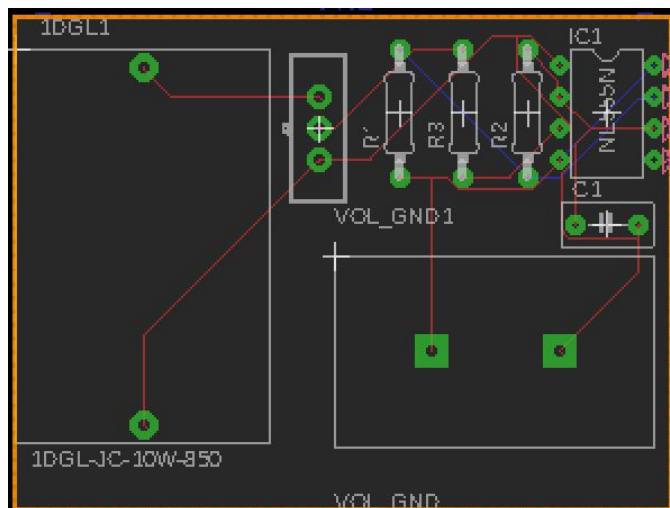


Fig. 3A: IR LED design on Eagle, software for routing circuit designs. Some parts are custom made to fit the IR beacon parts. Design consists of 2 layers, red and blue.

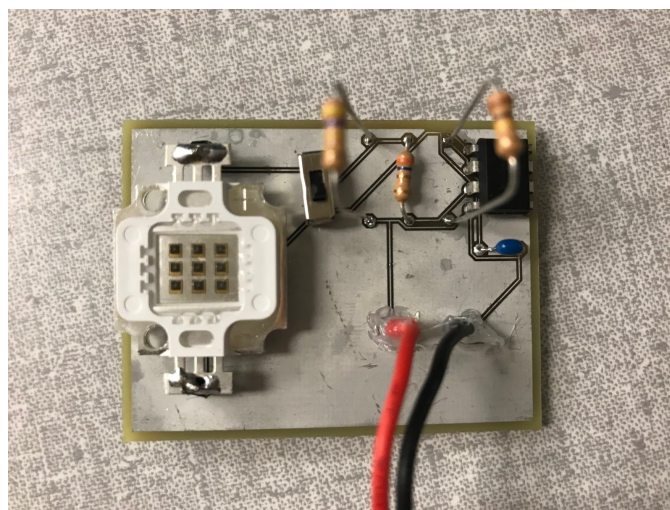


Fig. 3B: IR 10W High Power LED Beacon demoed at FPR. This Beacon is currently running at 9 volts using external battery source (portable), and operates through an on and off switch allowing for  $\sim 900$ mA of current to run through when in use.

### C. Block 2: Raspberry Pi/4G Interface

In order to deliver a high performance system that will meet the necessary requirements of law enforcement personnel, a strong processing component must be utilized for the image processing. Initially we were concerned with regard to the use of a Raspberry Pi to perform the image processing. The VideoCore IV GPU lacks the raw graphics processing power that some systems may require [8]. The worry was that it could severely limit the viewable framerates and analysis time, which is unacceptable as the effectiveness of the application demands on-time data delivery. If the video or flight instructions are not sent on time it could result in a loss of evidence, which defeats the purpose of the system. However, through testing we found that the Raspberry Pi would serve us well as both a means to process the computer vision aspects of the project as well as handling the data to be passed between subsystems.

In earlier stages of the project, we looked to harness the power of Google Compute Engine, a cloud computing solution developed to meet the requirements of the user. Utilizing Google's cloud platform would provide several benefits: the platform is comprised of a large collection of scalable virtual machines that can be reconfigured to meet the individual needs of each client, [9] compared to other cloud implementations Google's price-performance ratio is top-tier and allows for a powerful implementation while ensuring it is still suitable for our limited budget, and having a scalable performance solution will allow us to have the graphics processing capability that is needed for our OpenCV implementation to work efficiently.

However, the largest issue we found that would have occurred with the GCE was the latency in sending video data, not to mention the complexity that goes into doing so over a non-local network. The utilization of a 4G LTE modem raised a lot of concern. Initially our goal was to limit this latency via the use of newer video codec standards. One such is H.264/MPEG4-AVC, a common video compression codec that it utilized in many industries [10]. It is meant to transmit video at a higher quality while maintaining a lower bitrate and minimal latency. We also looked into several manufactured solutions that were meant to combine the video capture and data transmission together. For instance, The Sky Drone FPV 2 utilizes a custom UART protocol and 4G LTE technology to stream video footage at latencies under 150 ms [11].

These solutions, however, were not necessary, as we could rely on the Raspberry Pi to be strong enough, and thus shifted entirely to using it for our processing.



Fig.4: Sky Drone FPV 2 4G Camera/Transmitter. The device provides a solution which uses a custom video codec to provide quality footage at some of the lowest latencies for video over 4G networks.

The Raspberry Pi runs Raspbian as an OS. C/C++ based libraries are compiled and installed to it with Python bindings and all of the necessary image processing code is written in Python to minimize the complexities associated with the program while maintaining most of the underlying performance that C++ has. The details of the image processing will be discussed in the next section.

#### D. Block 3 - Object Tracking

At the heart of the BuddyCam system is an artificial intelligence system capable of detecting and tracking an object as it moves in its environment. Elements of this system will be completely software based, and interact directly with the *sensing array* system block — live video feed captured directly from the drone mounted camera will be processed by the object tracking system and logic will be returned to the drone to dictate which direction it is to move.

Once the live video stream has been sent from the *Sensing Array* block to the *Object Tracking* block, software processing of the video feed begins. To start, the video feed is broken down into 30 frames per second, and each frame is processed individually in real time. Two main processes are conducted during the processing of the frame. The first is to detect all persons in frame using deep learning, and the second is used to detect the IR LED beacon via color detection.

In the first portion of the image processing, the neural net library that is contained within the OpenCV library is utilized. Our goal with this portion is to detect all individuals within the frame, both officer and other pedestrians that may be near them. Utilizing a trained “caffe” model, a data representation produced by Berkeley, this can be done. The caffe model is comprised of data extracted from a large amount of images, taking more data in with each addition.

The model is utilized to form a deep neural net within OpenCV. This model is loaded into the program and read via OpenCV functions. Each time a frame is captured, it is then processed by this network. The regions where a pedestrian is detected within the frame are converted to blob images and location information is pushed into a data structure within Python. Using this information, we can extract the boundaries within the frame where the pedestrians are and utilize these to display where each individual is. The officer will be isolated from others in the following image processing step.





The initial portion of the image processing will detect a region containing individuals with high accuracy. The boundary box is displayed to showcase this processing in effect.

In the second portion of the image processing, the frame is first segmented using a process called image thresholding. This creates a mask on top of the frame, segmented only one specific shade of color in the frame. This shade correlates with the low-wavelength violet that is read in by most IR capable cameras. If the officer is wearing the IR beacon, this allows the system to differentiate from civilians and officers. To perform the segmentation, each pixel value in the image is recalculated according to a mask matrix, and the IR color in a HSV color space is segmented for, meaning all colors that do not fall within a certain boundary are turned to black and colors that do are turned to white. An example is shown in Fig. 5.

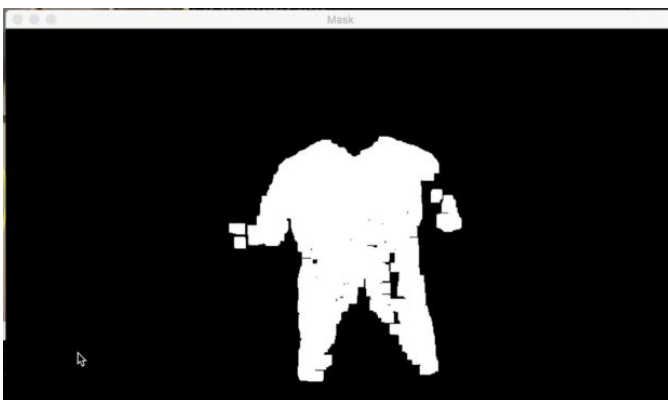


Fig. 5: Image mask in HSV colorspace using image thresholding. The largest pixel region within the image that correlates to the specified HSV boundaries is selected via BLOB analysis.

Next, the distance between the center of this segmented

subject and the center of the frame is calculated with a trivial distance formula displayed in Fig. 6. This is used to determine whether the subject is in the left, right, top or bottom of the frame. Necessary logic is derived from this information to tell the drone in which direction to move to maintain the center of the subject as close to the center of the frame as possible.

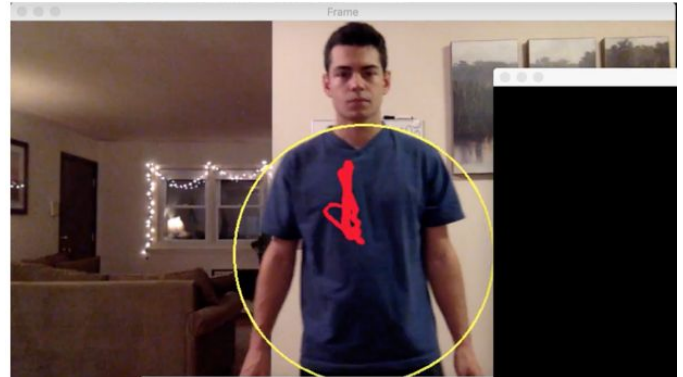


Fig. 6: Overlaid relative movement from center. The circle depicts the result of the BLOB analysis after applying the HSV mask. The red is tracking the approximate center of that region.

Testing of the *Object Tracking* block will take place through analyzing the results of this image processing in a variety of environments. Variables to consider include lighting, background of the environment, amount of people in the frame, and how fast the people are moving. These variables can be considered individually by manually controlling the others in a controlled testing environment.

#### E. Block 4 - GPS Client/Server

Though the target can reliably be tracked using the image processing that is conducted, it is not enough. The UAS must first be moved into a position where the image processing can locate the subject in frame. In addition, if the drone were to move to a position through error where the officer is no longer in frame, we need a way to correct this.

Our solution to these issues was to implement a GPS subsystem. There are two main components to this system. The first is an Android application which serves only to pull GPS coordinates. The phone running the application will be located on the officer, or person looking to be isolated. From here the GPS coordinates are read in through access to the phone's GPS system, which combines the onboard GPS module as well as location data collected from network and cell access. These coordinates are packaged into a small amount of data which is periodically sent to our GPS server.

The GPS server is run on the Google Compute Engine. The same benefits detailed earlier still apply in this situation, though it is only being used to collect GPS data from our application. The server maintains a static IP and when started will continually listen on a specified TCP port for our application. The GPS latitude and longitude are received through a socket connection on this port and unpackaged to

usable data. The server will then relay this information to the Raspberry Pi via 4G. The libraries used to control the UAS can be utilized to adjust the drone to a specific latitude and longitude based off of the data received and current GPS location provided by the GPS module within the base UAS.

With this system in place, we are able to solve the aforementioned issues and have a way of moving the drone into position, recovering from image processing failure, as well as supplementing the isolation as only the officer would be tracked via GPS.

Once the target has been successfully identified and tracked within the frame of the video stream, the relative movements of the subject with respect to the center of the frame combined with the information from tracking the IR beacon will be translated into flight instructions that move the drone to keep the officer in frame. Once the Raspberry Pi has a new movement instruction, it will send it to the onboard flight controller, which then adjusts the voltage sent to the rotors that correspond to the command. Other commands such as manual control, emergency landing, or subject switching can also be sent through this data path. The operational flow of the system is summarized in Fig. 8.

### III. PROJECT MANAGEMENT

Upon presentation of our project at the Preliminary Design Review, our group laid out a series of tangible, deliverable goals for the future of the project. By Mid Design Review, our group’s goal was to have the *Object Tracking* block completely finished. This included a live demonstration of video processing and object tracking, using a fixed camera and the various methods of software tracking explored above.

Through much hard work and development of the project, we were successfully able to deliver upon this goal for Mid Design Review. While we did change a few specifications of the project relating to hardware and the location of processing, we were fully able to demonstrate a fixed camera obtaining a live feed and tracking a subject through the frame.

In our group’s preliminary design review, project management was broken up in the following way; Saswati would be tasked with implementing the wireless transmission of analog video, a communication link from a drone mounted camera to the raspberry pi using radio frequency. Joseph would be responsible for implementing object recognition in image processing software. This included researching possible methods, such as image segmentation, for determining which object in the video frame is the desired subject. Finally, Steven and Kyle were tasked with implementing object tracking through the frame. This included implementing software that both determined the relative location in frame of the subject, as well as implementing the logic that will tell the drone where to move in order to maintain the subject in the center of the frame.

Following our preliminary design review, several changes were made to the technical implementation of our project, explained above, that moved the location of video processing from a base station to the onboard Raspberry Pi (see differences between Fig 7. and Fig. 8). Further, it was determined that an infrared beacon was necessary to further and more accurately detect the subject of the video frame. Per these specification changes, Saswati’s responsibilities were changed from video transmission to the implementation of the infrared beacon.

### IV. CONCLUSION

BuddyCam proceeded on schedule and met our expectations and design criteria. We intended to have our object recognition and tracking program written and running on a stationary camera for MDR. Taking on this objective first would layout most of the groundwork needed to complete the project. This goal was achieved, as well as preliminary implementations of both the infrared beacon circuit and the interface for communication between the flight controller and the Raspberry Pi. With one of the most challenging aspects of the project completed, we have moved onto focusing on our CDR deliverables, for which we intend to complete the work

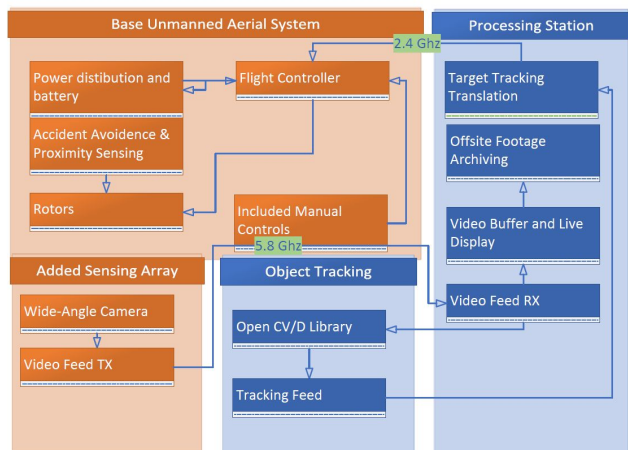


Fig. 7: Original block diagram presented at PDR

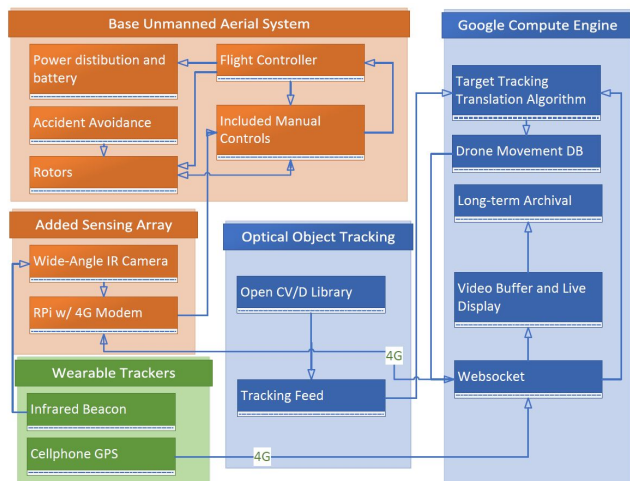


Fig. 8: Updated block diagram presented MDR. Note the addition of wearable trackers and a control server hosted on GCE instead of a nearby base station.

on the IR beacon, configuring the Raspberry Pi to work with our tracking algorithms, initialize the remote connection between the Raspberry Pi and GPS server via 4G, and begin formatting flight controls based on the outputs of the program. The schedule for our intended goals is seen in the Gantt chart (Fig. 9), indicating the aforementioned tasks be complete by the CDR presentation.

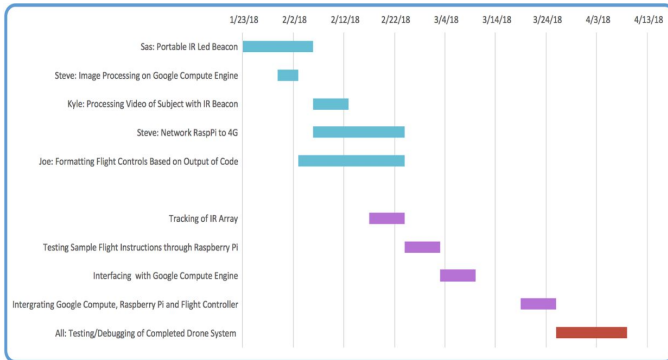


Fig. 9: Gantt Chart through FDR

Once our CDR objectives were complete, the majority of work beyond that point was integration of subsystems. The IR beacon array was combined with the OpenCV tracking, the flight controller interfaced with the Raspberry Pi and connected to the server via 4G. Finally testing and debugging took up the last portion of the schedule. By continuing to hold our group and advisor meetings, staying on schedule, and remaining dedicated to finishing what we've started, BuddyCam was an overall success.

V. APPENDIX

A. APPLICATION OF ENGINEERING

There are many areas of math, and engineering that apply to BuddyCam, most notably: probability, networks, electronics and circuit analysis. For the software development portion of BuddyCam the server, databases, image processing was all done through Python. The image processing used the Python libraries, specifically deep learning, in computer vision to help with detection of subject sin the video frame. We had exposure to these programming languages and libraries through ECE courses such as ECE242 Data Structures, ECE373 Software Intensive Engineering, ECE374 Compute Networks, and ECE597IP Image Processing. The IR beacon design was an essential subsystem that could not have been implemented correctly had we not been exposed to circuit design, that included awareness to power specifications in courses like ECE 211, ECE 212, ECE 323 and ECE 324. To achieve cohesiveness between all three subsystems and ensure they worked simultaneously used the concepts of probability, especially the Bayesian Rule from ECE 314. All these courses helped for the successful implementation and completion of this project.

B. BUDDYCAM COST

Below is our cost analysis for the project. The drone was given to us for use by our advisor, which allowed us to use the budget towards devices such as PCB, extra parts, and cameras. The chart below covers the total cost of implementing the entire system, assuming the the user has an android phone that is capable of downloading the GPS tracker app.

Development		Production	
Part	Price (\$)	Part	Price (\$)
<i>3dr Iris+ drone</i>	600	<i>3dr Iris+ drone</i>	600
<i>Raspberry Pi 3B</i>	35	<i>Raspberry Pi 3B</i>	21
<i>Rpi Battery</i>	20	<i>Rpi Battery</i>	14
<i>Logitech C920 Camera</i>	99	<i>Logitech C920 Camera</i>	85
<i>IR Beacon PCB</i>	90	<i>IR Beacon PCB</i>	8
<i>Data Cables</i>	10	<i>Data Cables</i>	1
<b>Total</b>	<b>854</b>	<b>Total</b>	<b>729</b>

VI. ACKNOWLEDGMENT

We would like to extend our appreciation and gratitude to Professor Pishro-Nik who throughout this process has helped keep us motivated and on track, as well as provided us with the resources and tools necessary to complete each iteration of the project. We would also like to thank Professors Polizzi and Koren for taking the time to meet with us and give us feedback on the current achievements of the project. Their comments and recommendations have helped shape the project to its current state and greatly influenced our way of thinking about how we look to finish out the year.

VII. REFERENCES

- [1] Bains, Chiraag. "Can Cops Use Force With Impunity When They've Created an Unsafe Situation?" *Slate Magazine*, The Slate Group, a Graham Holdings Company, 15 June 2017, www.slate.com/articles/news\_and\_politics/jurisprudence/2017/06/the\_supreme\_court\_suggests\_cops\_use\_of\_force\_is\_always\_justified.html.
- [2] O'Brien, Éadaoin, et al. "Science in the Court: Pitfalls, Challenges and Solutions." *Philosophical Transactions of the Royal Society B: Biological Sciences*, The Royal Society, 5 Aug. 2015, www.ncbi.nlm.nih.gov/pmc/articles/PMC4581010/.
- [3] Edmond G, Kemp R, Porter G, Hamer D, Burton M, Biber K, San Roque M. 2010. *Atkins v The Emperor: the 'cautious' use of unreliable 'expert' opinion*. Int. J. Evid. Proof 14, 146–166

- [4] "Using Video Surveillance as Evidence in Court." *SecurityBros*, [securitybros.com/using-video-surveillance-as-evidence-in-court/](http://securitybros.com/using-video-surveillance-as-evidence-in-court/).
- [5] "When Body-Worn Cameras Become a Matter of the Courts." *PoliceOne*, 23 Mar. 2017, [www.policeone.com/policing-in-the-video-age/articles/320408006-When-body-worn-cameras-become-a-matter-of-the-courts/](http://www.policeone.com/policing-in-the-video-age/articles/320408006-When-body-worn-cameras-become-a-matter-of-the-courts/).
- [6] A1RONZO. "IR Communication." *IR Communication*, [learn.sparkfun.com/tutorials/ir-communication#res](http://learn.sparkfun.com/tutorials/ir-communication#res).
- [7] Jayant. "IR Transmitter and Receiver." *IR Transmitter and Receiver Circuit Diagram*, [circuitdigest.com/electronic-circuits/ir-transmitter-and-receiver-circuit](http://circuitdigest.com/electronic-circuits/ir-transmitter-and-receiver-circuit).
- [8] "Raspberry Pi 3 Benchmarks." *MagPi*, RaspberryPi.org, Dec. 2016, [www.raspberrypi.org/magpi/raspberry-pi-3-specs-benchmarks/](http://www.raspberrypi.org/magpi/raspberry-pi-3-specs-benchmarks/).
- [9] "Compute Engine." Google Cloud Platform, Google, Inc., [cloud.google.com/compute/](http://cloud.google.com/compute/).
- [10] "An Overview of H.264 Advance Video Coding." *VCodex*, [vcodex.com/an-overview-of-h264-advanced-video-coding/](http://vcodex.com/an-overview-of-h264-advanced-video-coding/).
- [11] "Sky Drone FPV 2." *Sky Drone*, [www.skydrone.aero/products/sky-drone-fpv](http://www.skydrone.aero/products/sky-drone-fpv).