Mini AWACS Optical Project: Long Distance Stereo Depth Estimation

Leonard Mitchell

CREOL: The College of Optics and Photonics, University of Central Florida, Orlando FL, 32816-2450

ABSTRACT — The purpose of this project is to use optical technology to determine the depth/range of an object in relation to the system. The system will utilize stereo vision using two identical cameras and lens systems to capture stereo images of a scene. Using the stereo images and the extrinsic and intrinsic properties of the camera system image processing will be performed to rectify the images, perform disparity calculations and with the disparity calculations determine the distance from the camera system to the object in the scene captured by the images.

I. INTRODUCTION

Stereo vision is a widely used imaging technique for many applications including autonomous navigation, manufacturing, agriculture, medical imaging, and object detection and tracking. One of the main use cases for stereo vision is depth estimation. Depth estimation can be used to determine how far away objects within a scene are from the system. This technique is commonly used to perform 3D reconstruction and mapping which has a wide use case. For this project stereo vision will focus on depth estimation to determine the position of an object in relation to the system. Additionally, the system will be designed so that the scenes being captured are at a longer distance of roughly 200-300 ft to demonstrate the feasibility of stereo vision for long distance applications to mimic the intended purpose of the Mini-AWACS project. The long-distance imaging will be accomplished using a pair of matching custom designed telephoto lens systems connected to a pair of matching USB cameras. The cameras will be set at a baseline that will allow the cameras fields of view to overlap adequately to acquire stereo images. Although, it is beyond the scope of this project stereo imaging can also be used to track and identify objects in real time within the 3D mapping that depth estimation can provide.

II. SYSTEM COMPONENTS

Understanding of the system can be best understood by identifying each component of the system and its purpose towards achieving the goal of the project.

A. USB Cameras

The cameras selected for the project are ELP 2 Mega Pixel Global Shutter USB cameras with a 1/2.6" CMOS sensor. The global shutter was chosen as it will reduce the possibility of blurred images with moving objects. However, the sensor size of the camera limits the field of view (FOV) of the optical system which will significantly limit the size of the scene being captured which will limit the baseline of the system.

B. Telephoto Lens

The telephoto lenses consist of a positive 2" diameter positive achromatic doublet lens and a negative 20mm diameter lens. The telephoto design allows for long focal lengths while limiting the overall length of the system. Using these two lenses the system will have a 400mm focal length but a functional length of 360mm. The achromatic doublet is used to limit chromatic aberrations. Additionally, a field stop is incorporated in the system to assist with spherical and COMA aberrations.

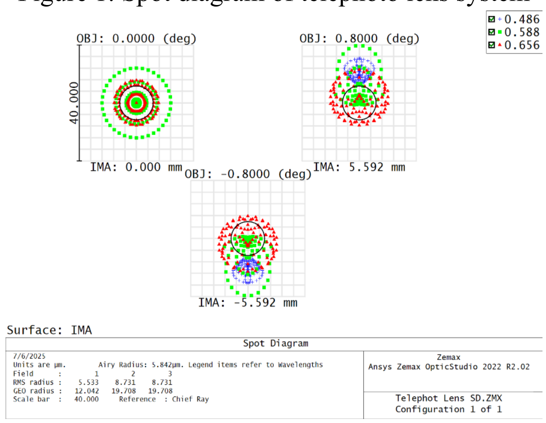
C. Image Processing

To perform the image processing required to provide depth estimation MATLAB was initially used as it has multiple add-ons to integrate USB cameras and has a stereo camera calibration app which can determine the extrinsic and intrinsic characteristics of the camera system. Although testing was initially successful, continued connectivity issues with the USB cameras occurred making the MATLAB software unusable. As a result, OpenCV is now being used to perform all image processing. OpenCV is python based and opensource. It is specifically designed for image processing and computer vision artificial intelligence and can perform all the desired image processing needed for the system to work properly.

III. HARDWARE PERFORMANCE

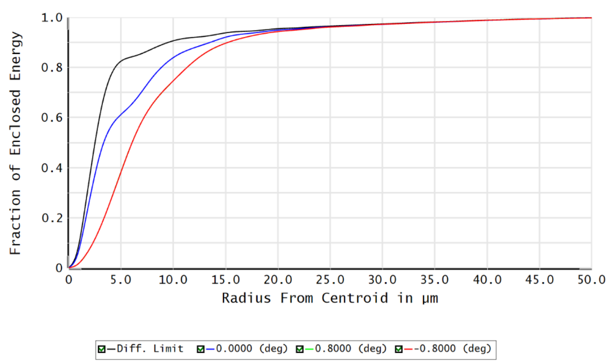
The performance of the system is highly dependent on the telephoto lenses and their integration with the USB cameras for image acquisition. Using Zemax to model the lens systems, the estimated performance can be seen in the figures below.

Figure 1: Spot diagram of telephoto lens system



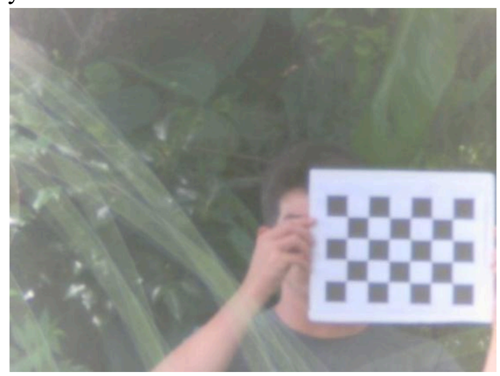
The spot diagram shows a near diffraction limited spot size for both on axis and off axis rays. However, some Chromatic aberration is still present.

Figure 2: Enclosed energy of the system



The enclosed energy of the system shows how much of the energy of the optical system stays within the desired spot size. We can see in the graph above that ~ 80% of the energy entering the system from on and off axis fields is contained within a 10 µm spot.

Figure 3: Sample image captured with lens and camera system used for initial stereo camera calibration



The sample image was taken with the assistant standing approximately 250ft away. This image was used to calibrate the stereo camera system using the checkerboard seen in the image. The checkerboard pattern consisted of a 7x5 pattern with squares of 35mm in length.

One of the primary drawbacks of the system is the limited FOV which is a result of the long focal length of 400 mm and the small format size of the sensor. The sensor diagonal is ~ 6.3mm. Using the FOV calculation:

$$FOV = 2 * arctan(\frac{d}{2*f})$$

$$2 * \text{arctan } arctan\left(\frac{6.3}{2*400}\right) = 0.902°$$

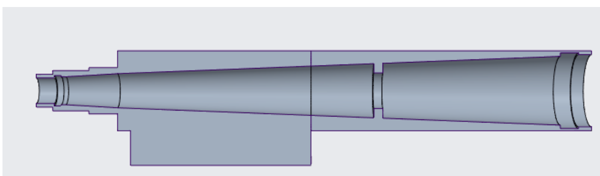
The limited FOV of 0.902° results in limitations with setting the baseline of the system which is the spacing between the stereo cameras. Because adequate overlap of the images is needed the baseline width was limited to 300mm initially during prototyping. This allowed for adequate overlap of the stereo images at distances between 70m-100m. However, after 3D printing the final platform there was an inadequate overlap of the stereo images for final stereo calibration. As a result, the baseline was decreased to 280mm which allowed for adequate image overlap.

In addition to the baseline distance between the cameras as listed above, the design of the lens tubes is a vital component of the system since successful image acquisition is dependent upon the exact spacing of the lenses, position of the stop shift and a manual focuser to adjust the position of the image plane in relation to the lenses to image at different distances.

Figure 4: Zemax Telephoto Lens data used for 3D CAD design

	Surface Type		Comment	Radius	Thickness		Material	Coating	Clear Semi-Dia	
0	OBJECT	Standard		Infinity	1.000E+05				1398.667	
1	(aper)	Standard	49289	227.160	8.000		N-BA...		24.500	U
2	(aper)	Standard		-174.8...	4.000		N-SF10		24.500	U
3	(aper)	Standard		-571.4...	105.888	V			24.500	U
4	STOP	Standard		Infinity	225.877	V			18.000	U
5	(aper)	Standard		-51.600	3.500		N-BK7		10.000	U
6	(aper)	Standard		Infinity	0.000	V			10.000	U
7		Standard		Infinity	13.561				5.618	
8	IMAGE	Standard		Infinity	-				5.731	

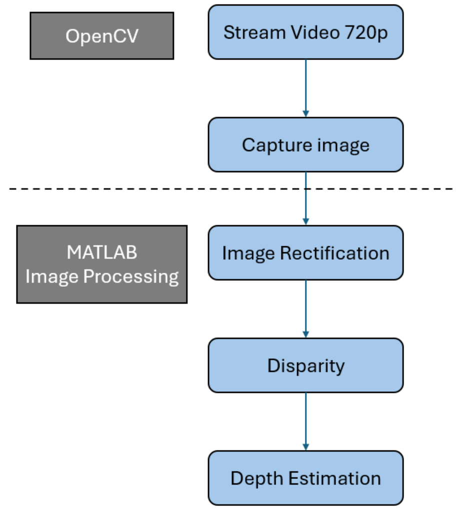
Figure 5: 3D CAD Design of lens tube/housing with rear focuser mount



IV.SOFTWARE/ IMAGE PROCESSING

The intended purpose of the system is to perform depth measurements of objects captured in the stereo images to determine the range of the object in meters. To accomplish this, two different image processing languages are used. Although MATLAB has add-on software to integrate with the stereo pair of USB cameras used in the project, the connectivity was unreliable and higher resolution images could not be captured. As a result, OpenCV, an opensource Python based computer vision programing language was used to stream video to allow the system to be target and acquire the desired stereo images. Although OpenCV has the capabilities to do all the camera calibration and image processing there was no additional time to learn OpenCV adequately to use its full capabilities. As a result, the camera calibration and determination of the cameras intrinsic and extrinsic parameters were done via MATLABS stereo camera calibration app. Additionally, stereo image processing includes image rectification and depth mapping which is all performed in MATLAB.

Figure 6: Software workflow diagram



V. SYSTEM PERFORMANCE

The optical systems including the cameras and telephoto lenses performed as expected noting good visualization of a sighting target at slightly over 100m away. Below is a stereo image pair noting good visualization of the intended target.

Figure 7: Stereo image pair of sighting target at slightly over 100m

Although, the MATLAB stereo camera calibration app demonstrated promise with properly functioning calibration files during initial research at short distances with the stock lenses, we were not able to produce a stereo camera calibration file that accurately represented the environment that the calibration images were captured in. Additionally capturing images that had enough of the frame covered with the calibration checkerboard proved very difficult. As a result, the original software workflow could not produce the desired depth sensing. The best calibration file that was obtained introduced significant distortion which can be seen below.

Figure 8: Rectified test image showing severe distortion introduced by the stereo camera calibration file

Because of the poor calibration the stereo camera computer vision algorithms were not effective in producing any usable depth measurements noting error of over 1000% when testing at ~ 100m.

Consequently, basic depth calculations were attempted using the equation below using the focal length of the lens (*f*), the baseline which is the distance between the center of the cameras (*b*), and disparity (*d*) which is the relative difference in the location of an object in the stereo image pair:

$$z = \frac{f*b}{d}$$

The disparity was calculated by finding the pixel coordinate of the same feature in the stereo image pairs and calculating the difference in the x coordinate location using MATLAB. Thus, *d* is defined as:

$$d = x_l - x_r$$

A MATLAB script was created allowing us to pick an easily identifiable feature in the stereo images to find the x- coordinate in each image and complete the depth calculation to determine the value of *z*. However, at the testing distance of just over 100m return values of ~ 27m noting a 73% depth estimation which is a significant improvement from the output of the computer vision stereo camera algorithms but does not demonstrate the desired performance of the system.

ACKNOWLEDGEMENT

The author of this paper would like to acknowledge and thank Dr. Stephen Eikenberry for assisting with the successful design and implementation of the lens systems used on the system. Additionally, we would also like to thank him for volunteering his time to be the reviewer of this project

Additionally, the author would like to thank Kiva McCracken and Neil Singh who were instrumental in the design of the lens housing and physical structure of the system

REFERENCES

[1] https://www.mathworks.com/help/vision/ug/using-the-stereo-camera-calibrator-app.html

[2] Qingsheng Mu,, Jun Weib, Zhugang Yuanand, Yuheng Yin. Research on Target Ranging Method Based on Binocular Stereo Vision. 2021 International Conference on Intelligent Computing Automation and Applications (ICAA)

Mini AWACS Electrical and Computer Engineering Project

Ido Yaari
Phillip Jones

Department of Electrical Engineering and Computer Science, University of Central Florida, Orlando, Florida 32816-2450

ABSTRACT — The purpose of this project is to sweep a Simrad Halo marine radar 60° beneath the belly of a small plane to detect Unmanned Aerial Systems in an area and send the radar data received to Samsung smartphones in the UAS. The system will use a rack and pinion design to sweep the radar, and an MSP430 microcontroller to send the radar angle data to the centralized computer via UART. The central computer will take the radar angle data and combine it with the parsed radar packets and send the data out to all recipients in the UAS.

I. INTRODUCTION

Current training methods require extremely expensive aerial radar systems in order to monitor the sky for all aircrafts in an area at a given time. Airborne Warning and Control System, or the AWACS costs tens of thousands of dollars an hour to operate, and hundreds of millions of dollars to build. Our sponsor wants to be able to deploy UAS as functional drone swarms and to test other applications for these platforms. However, due to the size and cost of these smaller UAS platforms a full sensor and radar package is too costly and not feasible. Our sponsor, United States Southern Command, or SOUTHCOM, is looking to build a system that functions the same way as the AWACS does, at a fraction of the price. SouthCOM has provided our group with a Simrad Halo Marine Radar, a central computer, a laptop, a starlink terminal, a samsung smartphone, and a power supply with the hopes of creating something that will change the future of airborne radar systems and unmanned aerial systems.

II. SYSTEM COMPONENTS

Although our sponsor has provided us with most of the components needed to get this project moving, some more, less costly, items were needed to complete this project. The components shown below are a mix of those provided to us by our sponsor, those provided to us by the University of Central Florida, and those that we purchased ourselves to ensure the full implementation of our project.

A. Simrad Halo Marine Radar

The radar chosen for our project is the Simrad Halo Marine Radar. This radar was chosen as it is open source and allowed our computer science team to edit and manipulate the code to fulfill our project's needs. Additionally, this radar is much smaller than other radars we have dealt with, and very lightweight as well, making it much easier to mount on the bottom of a plane or balloon when that time comes. Initially, our sponsor supplied us with a Raymarine 6' Cyclone Radar, which proved to be far too heavy and very difficult to work with, as raymarine keeps all its information as proprietary information, making the modification of its data packets very difficult.

B. *Starlink Mini Terminal*

The Starlink Mini Terminal provided to us by our sponsor is a major part in allowing our system to work. In real use, our system will be used in locations with no networks available, the Starlink mini terminal allows us to receive a network anywhere we are, which allows our radar, microcontroller, and computer all work as required.

C. *Intel NUC*

The Intel NUC is the centralized computer on board the plane which plays a critical role in the design of our project. The radar angle data is passed directly to the Intel NUC, and along with the radar data that it parses, it is able to create a packet that includes the range, azimuth and package it into a JSON and send it out to the UAS' smartphones.

E. *NEMA Stepper Motor*

The NEMA Stepper Motor we have chosen is the HT23-601D NEMA 23 High Torque Stepper Motor. We chose this motor for its High Torque which is required in order to sweep the radar, top down, in a 60° angle under the belly of the plane. We initially went with the HT08-230D NEMA 8 High Torque Stepper Motor but realized it would not be powerful enough to sweep our system.

F. *Power Supply*

The Westinghouse Portable Power Supply was provided to us by our sponsor. It is used to power the Intel NUC along with the Starlink Mini Terminal aboard the plane as it provides a 120V AC output and a 12V DC output. We also used a Talentcell DC Battery which offered a 12V DC output as well as a 24V DC output.

G. *STR 2 Motor Driver*

The motor driver we used was the STR 2 Motor Driver which was used to program our NEMA Stepper Motor to sweep the rack. We chose the STR 2 for its compatibility with both the stepper motors that we are using in this project.

H. *MSP430G2553*

The microcontroller that we used was the MSP430G2553. It was used to enable the UART communication which provided the sweeping radar's angle to the central computer.

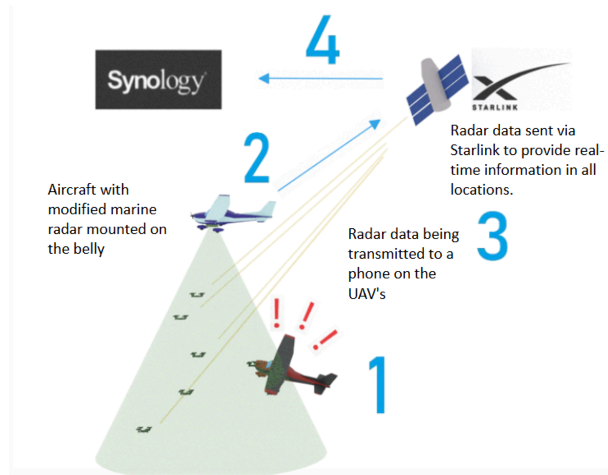
III. PROJECT SYSTEM FLOW

This project requires multiple sub-projects to ensure the full functionality of the system. The first portion of the project comes from the centralized computer. An algorithm, written by our computer science teammates, takes in the radar's data and parses the packets, ensuring that what we are sending out is information necessary for the UAS.

The second portion of the project is the sweeping rack and gear, which has many components within itself. In order to sweep the radar, a 3D printed rack and pinion design is needed. Once printed, the gear is mounted onto the stepper motor, which is programmed by the motor driver to deliver the 60° sweep.

Third, the sweeping radar's angle needs to be recorded every step in order for the computer to know where the radar is looking at any given moment. This is done through UART communication. The microcontroller, which is housed right beside the sweeping radar, tracks its sweeping motion using two buttons. Once the rack gets to the end of its sweep, a button is pushed, signaling for the count to go the opposite way, and the rack to move the opposite way as well. On the other end, another button is set up, which works the exact same way. After the first few sweeps, the maximum sweep is recorded and then the count becomes very accurate and the computer knows exactly where the radar is at all times.

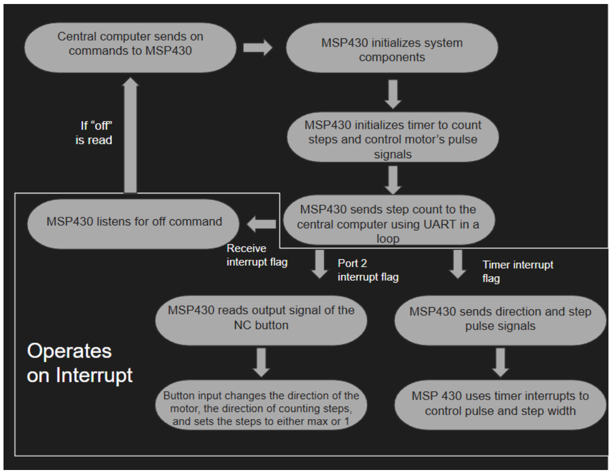
After this is all done, all the information recorded is sent through starlink to the UAS' samsung smartphones which are on board every UAS. Shown below is a photo of the system from afar.



IV. SOFTWARE IMPLEMENTATION

This project requires software implementation in every aspect of it, from the stepper motor, to the microcontroller, to the radar's raw data, software was needed. Below, we will go over the Software that was necessary to implement the ECE portions of this project.

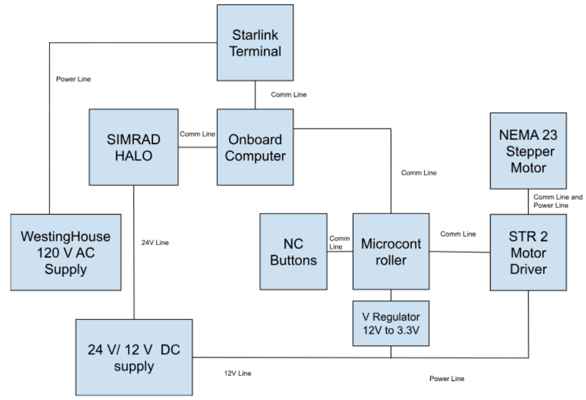
Initially, the central computer sends an "ON" command to the MSP430 microcontroller. The microcontroller then initializes its system components. Once it initializes everything, the microcontroller send the step count to the central computer using UART in a loop. From there, it operates in a loop on interrupts. There are three interrupt flags. The first one is the timer which calls the microcontroller to send direction and step pulse signals. It uses timer interrupts to control pulse and step width. The next is the interrupt that reads the output signal of the NC button. The button input changes the direction of the motor, the direction of the counting steps, and sets the steps to either max or one, depending on which button was hit. Lastly, the microcontroller listens for an "OFF" command from the central computer, if "OFF" is read, it shuts down and begins listening for the "ON" command once again. Shown below is the software design diagram of the implementation.



V. HARDWARE IMPLEMENTATION

The hardware design of the components has had many changes whether that be changing which power source is powering which component, or even changing the entire radar system of the project.

The project's main component is the onboard centralized computer. The computer is connected to three different components using communication lines, the Starlink Terminal, the SIMRAD HALO, and the Microcontroller. It is connected to the Starlink Terminal in order to receive a network connection. The Starlink Terminal is connected to the Westinghouse 120V AC Supply through a powerline. The SIMRAD HALO is connected to the centralized computer in order to pass the raw data it takes in. It is connected to the Talentcell 24V DC Output using a 24V line. The microcontroller is connected to the onboard centralized computer using a communication line as well. It is connected to the normal closed buttons, the STR 2 Motor Driver, and the voltage regulator. The microcontroller is connected to the NC buttons in order to change motor direction and step count, it is connected to the regulator to receive the right amount of voltage, and it is connected to the motor driver to send its signals through it. The regulator is connected to the 24V/12V Talentcell DC output using a 12V line, it takes in 12V and delivers 3.3V to the microcontroller. Lastly, the motor driver is connected to the NEMA 23 Stepper Motor using both a communication line, along with a power line. Shown below the block diagram of our hardware design.

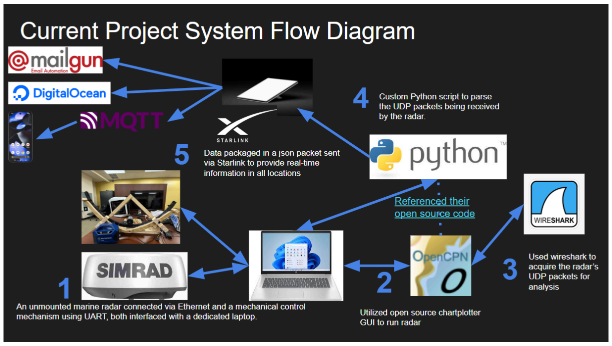


VI . COMPUTER SCIENCE SOFTWARE

The computer science side of this project was required in order to make the project work. As a multidisciplinary team we worked together to ensure the full functionality of the project that U.S. SOUTHCOM provided us. The computer science team used many different softwares to allow this project to meet its full potential.

Visual Studio Code is the main software application that the computer science team uses to write and test their code. Code Composer Studio is used heavily on this project as well however it is for the computer and electrical engineering side of the project. Wireshark is a tool that allowed the computer science team to capture the raw UDP packets from the SIMRAD HALO Marine Radar over ethernet. The language they used to write their code which did this was Python. They wrote custom scripts to receive and decode radar spoke data. They also transformed range, azimuth, and elevation data into structured messages.

PySerial is the serial communication that receives the elevation angle via UART from the microcontroller. The OpenCPN plugin, radar, reverse engineered the C++ program to understand the range and doppler format. Node.js is the backend server that handles radar data and websocket broadcasting, while socket IO handles real time communication between the radar server and the client dashboard. Then, PM2 is used as the process manager for the running node server on the remote cloud. The remote host we used was DigitalOcean. Mailgun API is used for email alerts and lastly MQTT is used to send real time data to lightweight clients, which were the Samsung smartphones that will be placed in the unmanned aerial systems. Shown below is a diagram which shows how this all ties into each other.



VII . CONCLUSION

In conclusion, as a multidisciplinary project, our project had plenty of portions which came together to form our final product. Our software and hardware components all worked together to ensure the full final functionality of our project. Although our project is over, there are still ways to enhance it and ensure it is capable of being mounted on a plane and used in real time. First of all, an aerodynamic encasing is necessary to hold the mechanism and make sure that no wind resistance affects the data sent out. Second, as a team with no mechanical engineers, a better mechanical design for the project would be necessary for it to be able to handle the weight of the entire radar, along with normal plane movements. We made due of what we had and have created a fully functioning prototype of what may change the future for aerial radar systems.

VIII . ACKNOWLEDGEMENT

The authors of this paper would like to acknowledge the hard work and dedication of multiple professors, sponsors, and mentors throughout the entire project.

First and foremost, our professors Dr. Wei, Dr. Chan, and Dr. Weeks who spent unbelievable amounts of time perfecting our projects to minimize hiccups that would occur along the way, as well as ensuring our finished products would work as intended. Dr. Weeks spent hundreds of hours with us in the ECE Lab throughout this project to ensure our success.

Secondly, our sponsor Roy Wubker from United States Southern Command. Roy helped us tremendously throughout this project. He shared his expertise he has gained throughout the years with us to ensure the project's success. He gave us many ideas, and helped us solve many complex problems which we were unable to do on our own.

Lastly, we would like to acknowledge the amazing efforts of Tony Levesque, the certified raymarine dealer who had no obligation to assist and ensure this project's success, however assisted us plenty of times whether it be when our Raymarine Radar was not working, to ensuring we get replacement pieces from Raymarine. He also helped us with design ideas and implementation.

IX. REFERENCES

[1] *E-3 Sentry (AWACS)*. Air Force. (n.d.-b). https://www.af.mil/About-Us/Fact-Sheets/Display/Article/104504/e-3-sentry-awacs/