Coleman University 2016 Journal Paper

Team Captain: Nicholas Cantrell, President: Bradley Risse,
Chief Financial Officer: Mong Ng, OpenCV Programmers: Choi Ng & Mong Him Ng, NodeJS Server:Phelps Witter
Lead Mechanical Engineer: Adam Thorp, Unity-3D: Bradley Risse, Alex Gilbert DevOps Lead: JayTachibana
Computer Science Advisor: Delane Pickel Mathematics Advisor: Ken Kuniyuki Engineering Advisor: James Burns

Abstract—This paper will document the project management of the 2016 Coleman University and Electric Networked Vehicle Institute(ENVI) Robosub Team, and how an off-the-shelf Remotely Operated Vehicle(ROV) kit was equipped with extended functionality. This effort included the integration of sensors, actuators, energy storage, in addition to the development of custom vehicle autonomy code written for the AUVSI Foundation's 2016 Robosub Competition.

I. INTRODUCTION

ENVI Unmanned Underwater Vehicle Team is cooperative development effort between Coleman University and the Mesa College MATLAB Club. The purpose of this partnership is to build a platform and curriculum which will empower students pursuing STEM education paths to apply the knowledge they learn in the classroom to real world problems.

II. REQUIREMENTS ANALYSIS

The published evaluation rubric used for scoring teams in the AUVSI Foundation's Robosub competition provided our team with a means to quantify the relative importance of different areas of the competition. This enabled the team to focus its efforts on the components of the competition that were most important to the customer.

Like any project, we had a finite pool of available time, money, and labor to draw from. This reality impressed upon our team how important it was to be realistic about what could be accomplished before the competition. To make efficient use of the resources we had available, a risk-adjusted Return on Investment analysis was conducted on a per-task basis to cap expenditures and create a project budget and fundraising goal.

We were able to simplify this analysis to a single dimension by assigning a monetary value to both schedule and labor. These values were approximated from past experience(e.g. economic decisions such as *expediting shipping* or *machining your own fasteners*) to generate a realistic value.

Year	2015					2016						
Month	August	September	October	November	December	January	February	March	April	May	June	July
Mechanical / Electrical	Planning		Component Selection		CAD Assembly		Mechanical Drawings			Vehicle Assembly	Pool Testing	Competition
Business Considerations			Fundraising			Fundraising		Vehicle Manufacturing / Component Purchasing		Delivery Lead Time		
Team Organization			Planning			Team Formation / Preparation			Recruiting	Training		
Computer Science						Planning				OpenCV Filters		

Fig 1. An estimated project schedule if we had elected to build a custom vehicle within our resource constraints.

Since both "cost" and "risk" are estimated values, we found it beneficial to conduct this analysis on a recurring basis so that the availability of new evidence could be incorporated into the decision making process.

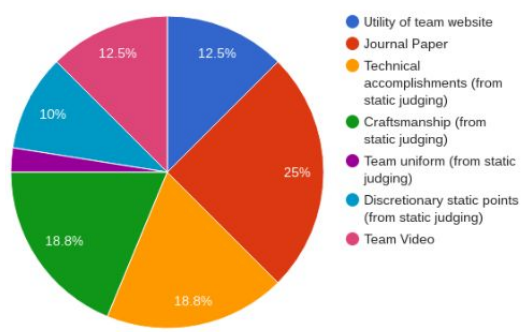

Fig 2. The limited number of available points for "Craftsmanship"(relative to their expense), deprioritized this component of the competition as a low ROI undertaking once a viable commercial alternative became available.

This analysis lead to an early decision that our team would purchase its vehicle off-the-shelf and instead focus the resulting free time, energy, and acceptable risk on topics such as as sensors, actuators, vehicle autonomy, and systems reliability testing.

We estimate this decision saved our team 5-7 months of mechanical build schedule(Figures 1,2, & 3), which we were then able to reinvest in component selection, testing, and conducting experiments with Artificial Neural Networks. These experiments were only possible due to our team's WebGL and Unity 3D based TRANSDEC Simulators "SimBox" and "SimBox 2".

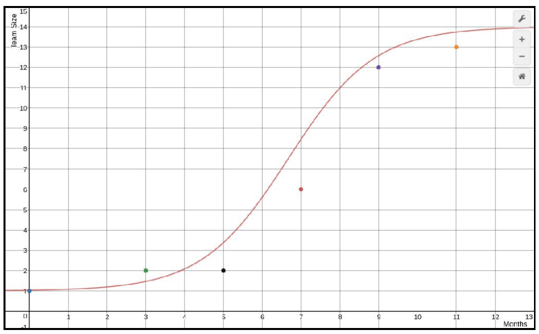

Fig 3. Team headcount as a function of project time in months.

When it became apparent in April 2016 that a pure Neural Network solution would require better training data to assist us in the localization of our vehicle, the decision was made to grow the team to a level which could support a more predictable and conservative software development approach.

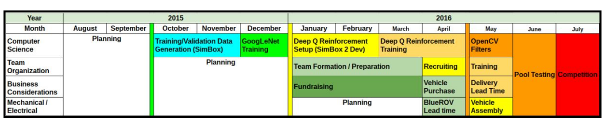

Fig 4. In addition to saving money, we are estimating the decision to purchase our vehicle as a kit bought us 7 months of project schedule. We choose to use this time to study reinforcement, and convolutional neural networks using our TRANSDEC Simulators.

Although this deviation from the original strategy was a less exciting path forward, the team had been entrusted with the necessary seed capital to purchase a vehicle and the supporting equipment. The responsible decision to our project's investors and our team's future credibility was to focus the remaining project schedule on a goal we knew could be achieved within the time we had left before competition.

III. STUDENT ENGINEERING WORK

A. Design Strategy

In the previous section, we outlined the progression of our team's efforts to prepare a vehicle capable of doing well in the "Performance" component of the Robosub Competition. After 11 months of preparation, experimentation, coding, and testing: we are now well-positioned to describe our plan for making optimal use of our vehicle's time in the water of the TRANSDEC competition arena.

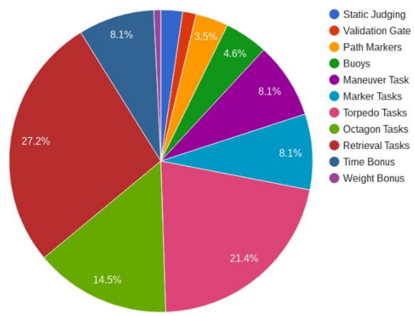

Fig 5. To select specific mission tasks, an attempt was made to estimate task difficulty, and generate a chart of competition-point *expected value.*

The above analysis was fed into a *Bill of Materials cost-impact* matrix. By knowing both the cost, and benefit of a capability on a task-by-task basis: we were able to rank each task according to the ratio between the cost of acquisition and the benefit of success. The Validation Gate and Pathmarker tasks were both excluded from this analysis. While the Validation Gate is simply a competition requirement, Pathmarkers play an essential role in facilitating localization and navigation to other mission tasks.

1. *Inertial Navigation*

Possibly the least expensive capability to implement poorly, and also one of the most expensive capabilities to implement well: our budget for an inertial navigation system(INS) capability was $200. The PixHawk flight controller we selected for pitch/roll/yaw stabilization came with the sensor hardware, and the ArduSub firmware we are using for the same purpose integrates an Extended Kalman Filter(EKF) for sensor fusion. Testing demonstrated that after an initial GPS Fix, the EKF can continue to provide position and velocity data after GPS signal loss(see Figures 6 and 8).

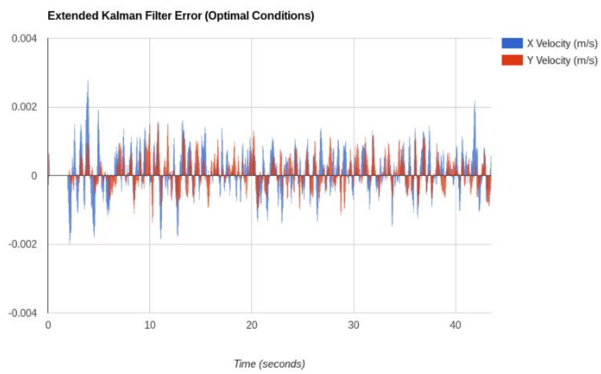

Fig 6. Visualization of EKF error by recording estimated velocity from a stationary PixHawk flight-controller running ArduSub firmware w/ GPS unplugged.

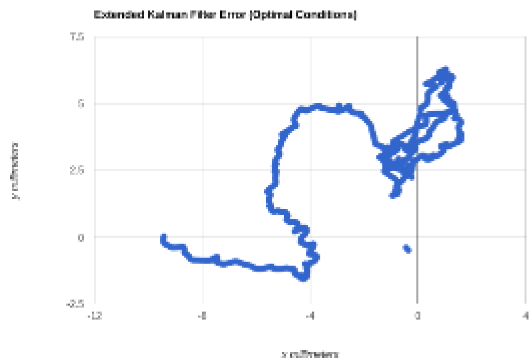

Fig 8. Visualization of EKF error (42 seconds) by recording estimated position from a stationary PixHawk flight-controller running ArduSub firmware w/ GPS unplugged.

2. Validation Gate

The absolute minimum requirement to achieve this capability appears to be the ability to stay submerged and swim in a straight line. There are many digital and analog solutions which could potentially facilitate this capability, however in addition to *depth-hold*, the ArduSub firmware already uses PID Control loops to stabilize the BlueROV in Yaw/Pitch/Roll.

Fig 7. Ground Truth Segmentation images have been generated from available media to allow for improvements in image filter code to be measured. A sample subset has been reserved for validation testing to avoid overtraining.

Going beyond *dead-reckoning*, our team intends to use one front-facing camera to extrapolate vehicle position in the TRANSDEC relative to the Validation Gate. These relative coordinates can then be combined with GPS sensor data collected during *survey runs* to constrain the vehicle's position in the TRANSDEC environment to an acceptable level of uncertainty.

The danger of adding a secondary yaw-reference is the need to reconcile disagreement between the two sensors. E.g. if the cameras were to incorrectly identify a reflection as an orange pipe, lack of appropriate fault handling could make this less reliable than a gyroscope-only approach.

3. Pathmarkers

Pathmarkers are an easy task to overlook, however they provide a lot of value relative to the small expense of gaining this capability:

- each Pathmarker is worth points
- they assist with both vehicle localization and orientation in yaw
- they are one of the less-difficult objects in the TRANSDEC to segment from an image

Our team is interested in using our GPS antenna mast to map the floor of the TRANSDEC. We will attempt to do so by recording video while conducting a *grid search pattern* with the BlueROV under tethered operation. This will provide us with the necessary environmental data to build a validation dataset for future attempts to train Artificial Neural Networks to output localization data.

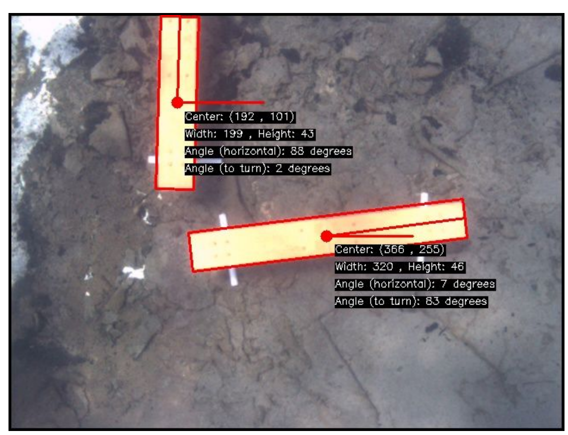

Fig 9. Our Pathmarker detection code running on a video frame. (Test Image Credit: CUAUV Robosub Team)[1]

This same data collection is also important for our team's success in this year's competition. Once these survey runs have been conducted, we can adjust the preset Pathmarker GPS coordinates written into the navigation code with the new, and more accurate, estimates.

4. White Marker Bins

White Marker Bins are a *stretch-goal* for our team this year. The decision was made to add a marker dropper to the vehicle (See: *Figure 10*) to give it the necessary hardware to attempt a more difficult mission.

Fig 10. 3D Printed ¾" SS Ball Bearing Marker Dropper Designed by Adam Thorp

Similar to the Pathmarkers: the white bins are less difficult to segment from images than some of the other alternatives, and they can be useful in assisting with localization.

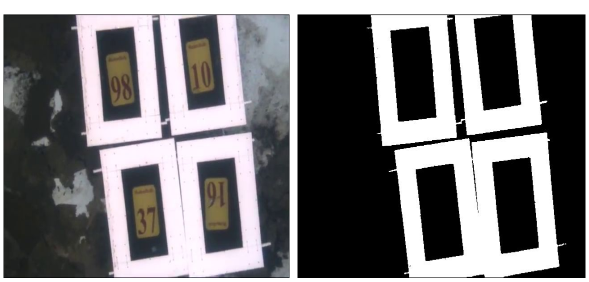

Fig 11. Ground Truth Segmentation images have been generated from available media to allow for improvements in image filter code to be measured. (Test Image Credit: CUAUV Robosub Team)[1]

5. Octagon

Although attempting a passive sonar would be an interesting challenge in the future, for this year's competition: our team has decided to forgo integrating a hydrophone array on our vehicle.

In order to get points for surfacing within the octagon, this leaves us dependent on computer vision and inertial navigation to identify the correct location for our vehicle to resurface.

6. Engineering Tradeoffs

When considering the tradeoff between reliability and complexity, our team sought to minimize complexity wherever possible. As a new team to the competition, we recognized that spreading our focus too thin presented a very real danger. In light of this, our strategy for this year's competition was "minimal complexity will maximize performance".

As an example of an area where our team chose simplicity: by simply adding a second camera to the front of the vehicle, we would have had the necessary sensors required for Stereo Correspondence based depth estimation.

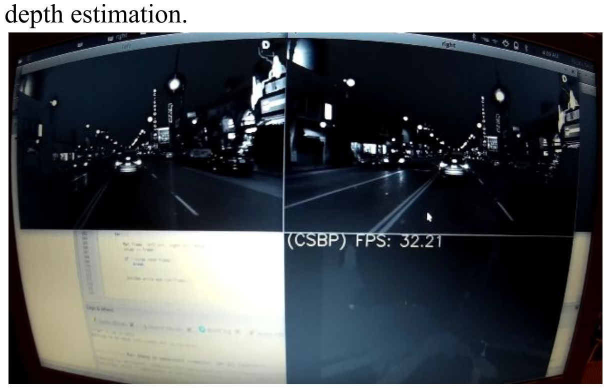

Fig 12. Some minor modification to GPU accelerated sample code allowed us to generate RGB-D data(30fps@480p) as early as January 1st, 2016.

Although this functionality could be added, doing so would require extending a task list at a time when our primary focus is on removing items from that list to reduce complexity, and thereby improve reliability.

B. Vehicle Design

Our vehicle was designed by BlueRobotics of Pasadena CA. It features six T200 Thrusters w/ 25A “Basic ESCs”, in addition to a 4in Dia. Pressure housing rated to a max depth of 100 meters, and weighs slightly under 7 kg(with electronics). The vehicle’s capabilities can be extended through the use of 6mm and 8mm Cable Penetrators. These threaded penetrators use Marine Epoxy to seal cables into a maintainable bulkhead penetration solution. These can be used in place of waterproof connectors when “quick-disconnect” connections is not a necessary feature.

Fig.13 The ENVI 2016 Vehicle “ENVI-UUV” was designed to accomplish four missions.

The BlueROV kit was designed for the purpose of tethered operation. As a consequence, it leaves the topics of energy storage(optional), vehicle autonomy(optional), and sensor-specifics to be selected by their customers. BlueRobotics publishes a reference design for an electronics payload which was designed for use as an ROV. This reference design served as our vehicle’s foundation, with modifications made on an *as-necessary* basis to enhance the vehicle's capabilities and performance in the competition.

These modifications included adding:

- a single 4s 10000mAh at 10C Lithium Polymer battery to the vehicle(to allow for untethered operation)
- a 75 meter *Fathom Tether*(CAT-5E) from BlueRobotics
- a kill switch (for diver control of the vehicle)
- 4x IP68 Sports Cameras(for data collection),
- a PixHawk flight controller
- an Odroid C2 Companion computer
- an externally mounted IP67 hobby servo(for the marker dropper)
- a GPS antenna mast (to allow individual video frames to be labeled with GPS data)

1. Imaging Sensors

Imaging sensor selection, high quality optics, and controlled lighting conditions can all reduce the difficulty of image segmentation. Low-quality USB webcams sometimes suffer from poor low-light performance. We estimated that any cost-savings achieved through their use in this project would be consumed by the associated labor costs of attempting to enhance their performance with DSP techniques.

It was deemed prudent to prioritize resource investments in the selection of an image sensor which offered vivid colors, acceptable low-light sensitivity, and ideally: this sensor would come packaged in an IP68 case. This would provide flexibility by allowing the placement of the camera outside the pressure housing. Finally, to minimize the integration difficulty: V4L2 driver support (as a USB webcam) was a priority for the sake of compatibility with OpenCV for Linux.

Fig 14. SJ4000 Waterproof Sports Camera

Identifying a camera which meets all of these criteria to a satisfactory level was not easy, and doing so while being mindful of cost was an even greater challenge. Ultimately, our team selected the SJ4000 Sports Camera from SJCAM as its primary camera due to its acceptable performance for every selection criteria.

2. *Software*

Our team's vehicle is controlled by a Node.JS middleware application running on an Odroid C2 *Companion Computer*. It's purpose is to provide high level instructions to the ArduSub firmware running on the PixHawk controller, to interpret the information generated by the OpenCV camera code, and to send/receive serial communications with the Teensy 3.2 MCU hardware interface board.

This middleware must coordinate processes and pipe messages between several subsystems including:

1. OpenCV Vision Code is responsible for sending JSON encoded messages containing information such as distance to objects, visual servo feedback, and optical flow velocity estimates
2. Localization.js is responsible for generating position estimates based on information received from the OpenCV code and generating NMEA 0183 GPS sentences to spoof the *ArduSub* firmware's Kalman filter
3. Captain.js reads a JSON file containing a sequence of desired waypoints and maximum time values before switching to the next task. Captain.js is also responsible for error-handling.
4. All PID control loops and mission-specific visual servoing code is stored in Helm.js

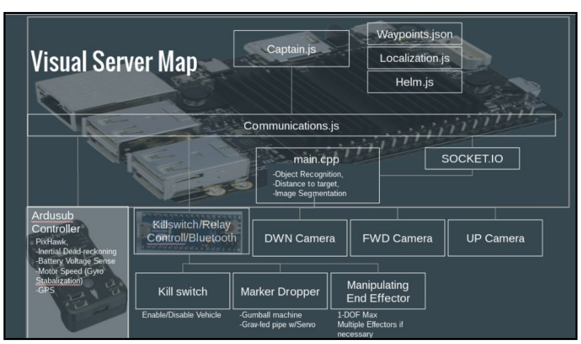

Fig 15. High level system diagram

3. *SimBox and SimBox 2*

For some members of this team who have done this competition before: one of the big lessons learned from past experience is "*Pool testing is almost as expensive as it is informative*".

In an ideal world, the workflow for designing an experiment and testing a hypothesis is as simple as compiling "Hello World". A perfect simulator would model the problem so accurately that the results inside the simulator would match the results in the real world 100% of the time.

SimBox(Figure 16) was originally created to give 1st and 2nd-year college students an opportunity to hit "Fast-Fwd" and skip-over a mechanical submarine build so they could immediately start writing vehicle autonomy and computer vision code.

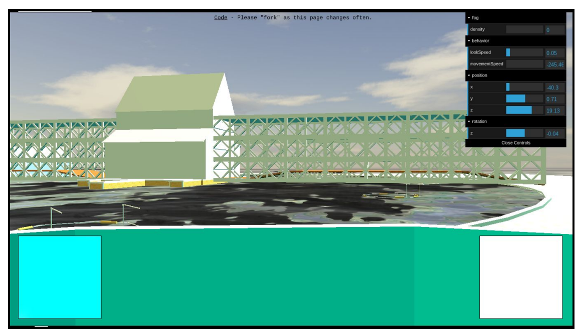

Fig 16. Textures and lighting of original SimBox

SimBox development didn't stop in 2015. A new version was created at the beginning of 2016 in order to give a competitive advantage to our team. Specifically: by allowing for the automatic labeling of images for "Deep Learning" neural network training(Figure 17).

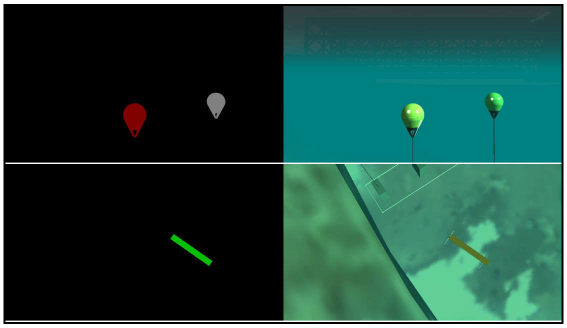

Fig 17. Automatic generation of *PASCAL 2012* formatted image segmentation training data

Unfortunately, when our neural networks(that had been trained on images from *SimBox)* were tested on real photographs from the competition: the results did not generalize well. This motivated an attempt to produce more realistic renders using a competing game engine: *Unity-3D(Figure 17).*

Fig 17. Improved Lighting and textures in SimBox 2

Unlike its ancestor, *SimBox 2* is no-longer entirely web browser-based. Despite this: *SimBox 2* maintains most of the cross-platform functionality, which is a feature of the *Unity-3D* engine.

4. *Useful Applications*

Some of the more challenging and time consuming problems we encountered when working on our Robosub Simulators was *Image Stitching, Photogrammetry,* and *Texture Synthesis*. Depending on the specific implementation: each one of these problems can slow down even an extremely powerful computer to a crawl.

Although we did not conduct an exhaustive search, we found: Microsoft *Image Composite Editor(ICE)*, Agisoft *PhotoScan*, and the *Resynthesizer* plugin for GNU Image Manipulation Program to be the most effective tools for each respective problem.

Ultimately we were unable to identify a satisfactory workflow for labeling ground truth image segmentation data. In the future, we may consider a custom solution based on the *Watershed* algorithm. We ended up resorting to Photo Manipulation tools such as GNU Image Manipulation Program, for lack of a satisfactory alternative. Although this worked, it was not ideal.

C. Experimental Results

A significant focus of our team's effort this year was to work on our software using simulated data. The theory behind this workflow was: it would yield a higher quality of code vs. delaying testing until an actual pool-test date. As of June 19, 2016: our team's vehicle has only been in the water once, for a little over an hour, to verify that the pressure housing was waterproof. Despite the low number of pool-hours, we have been able to significantly improve our software and understanding of the problems posed by vehicle autonomy through our experimental research with the TRANSDEC simulator *SimBox*.

1. *Experiments with "Deep" Neural Networks*

Our team's experimental data came in the form of Convolution Neural Networks. Specifically: training a GoogLeNet network using NVIDIA DIGITS 3. After the training was complete, we were able to feed the resulting network batches of screen captures like one seen in Figure 18.

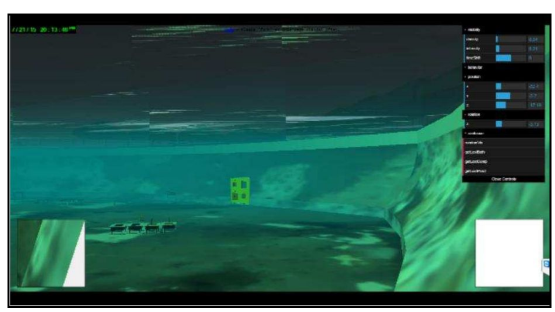

Fig 18. Example of a batch test picture

The DIGITS 3 web application outputs statistics and visualizations which assist in the process of identifying ways to improve network performance(Figure 19).

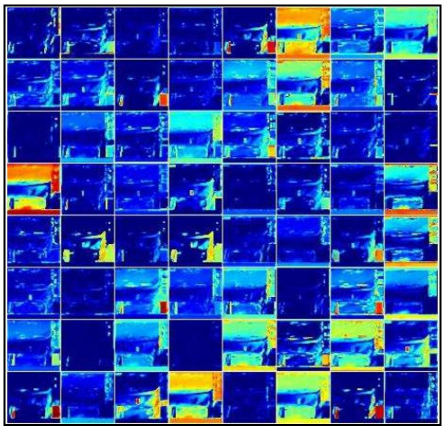

Fig 19. Example of a convolution kernel visualization

In figure 19, we can see the network activation on the clouds and *skybox* game texture. This helped develop a new video-based sky texture to help the network avoid learning patterns which won't generalize in the actual TRANSDEC environment.

D. Acknowledgements

ENVI-UUV would like to thank: *Coleman University,* the *Electric and Networked Vehicle Institute (ENVI)*, the *San Diego Mesa College Associated Student Government(A.S.G.)*, the *San Diego Mesa College Interclub Council(I.C.C.),* and *Dr. James Burns* for their generous support, and financial contributions to this project.

E. References

[1] CUAUV. Cornell University (2016, May). "Collaborate with Us." [Online]. Available: http://www.cuauv.org/open_source.php

F. Outreach Activities

To reduce the barriers to entry for Unmanned Systems, Machine Learning, and Computer Vision education: ENVI-UUV has decided to release several image datasets, and various related media, which we have produced over the past year for our own use.

We hope that this data will promote an interest among students in studying STEM topics in school, or otherwise facilitate the proliferation and development of useful skills.

https://goo.gl/QIXss4

If you are considering participating in Robosub 2017(or later): We hope that these files are useful to you, and look forward to meeting you in person!