Coleman University [2017] Journal Paper[1]

Team Captain: Ryan Jones | Project Manager: Nitin Naik
Vehicle Autonomy: Jake Ramirez | Machine Learning Research: Nicholas Cantrell, Choi Ng, Phelps Witter
ROS Integration: Henry Gaudet | Unity Developer: Juanita Krzymowski | DevOps Lead: Justony Nguyen

Abstract—This paper will document the efforts of the [2017] Coleman University and Electric Networked Vehicle Institute(ENVI) Robosub Team. This included fundraising, outreach, and the development of a Deep Neural Network for the AUVSI Foundation's [2017] Robosub Competition.[1]

I. INTRODUCTION

ENVI Unmanned Underwater Vehicle Team is a 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. [1]

II. PROJECT TIMELINE

Like many projects, the scope of work completed by the Coleman 2017 RoboSub team over the past year was heavily influenced by project budget. In August of 2016, the team set a fundraising target of $2500 in order to achieve their 2017 Competition goals. This number was generated based on the necessary expenditures in Table 1.

Capability	Component Required	Price	Date Purchased	Donotion Recieved
Participation	Competition Entry Fee	-$650.00	2/21/17	4/20/17
Deep Neural Network (training)	NVIDIA® GTX 1080 Ti	-$800.00	3/14/17	3/14/17
Deep Neural Network (vehicle)	NVIDIA® Jetson™ TX2	-$400.00	6/08/17	6/08/17
Deep Neural Network (vehicle)	Orbitty Carrier for NVIDIA® Jetson™ TX2	-$200.00	6/16/17	6/16/17
Deep Neural Network (vehicle)	4" Pressure Housing	-$300.00	TBD	N/A
Hydrophone Data Collection	4-Channel 192khz Sample Rate ADC	-$300.00	TBD	N/A

Table 1. After incurring large cloud-computing expenses on individual student's credit cards, it was determined that purchasing a graphics card for Deep Learning research was a much more cost-effective method of continuing our team's machine learning research from last year.

In 2016, our team conducted a requirements analysis for the purpose of designing last year's vehicle. In 2017, the team's goals remained unchanged which allowed us to reuse the previous year's vehicle for this year's competition.

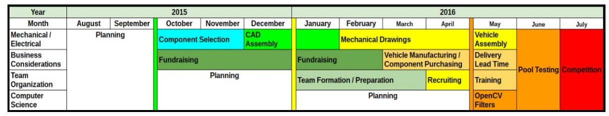

Fig 1. An estimated project schedule if we had elected to build a custom vehicle within our resource constraints.[1]

[Considering the cost of building a new vehicle, and no obvious benefit to replacing the one we already had, it was an easy decision to continue to use the old vehicle in the 2017 Competition.] This allowed us to reallocate resulting free time, energy, and acceptable risk on topics such as as vehicle autonomy [...] and systems reliability testing.[1]

III. STUDENT ENGINEERING WORK

A. Design Strategy

In our 2016 Journal paper, 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. None of this information has changed since last year; so, for the sake of completeness, an exact copy is being included in this year's revision of the journal paper. Any paragraph or annotation terminated with a "[1]" is copied whole-cloth from last year's paper.

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 2 and 3). [1]

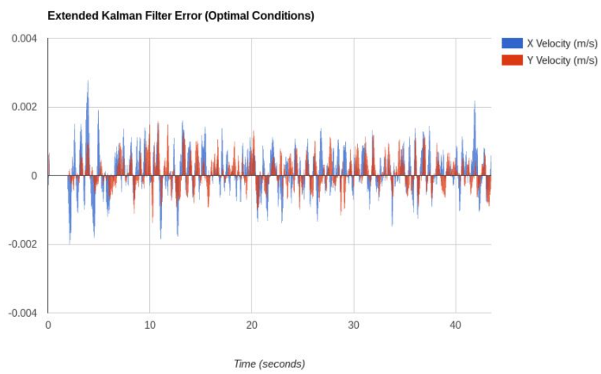

Fig 2. Visualization of EKF error by recording estimated velocity from a stationary PixHawk flight-controller running ArduSub firmware w/ GPS unplugged. [1]

Fig 3. Visualization of EKF error (42 seconds) by recording estimated position from a stationary PixHawk flight-controller running ArduSub firmware w/ GPS unplugged. [1]

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. [1]

Fig 4. 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. [1]

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. [1]

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. [1]

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.[1]

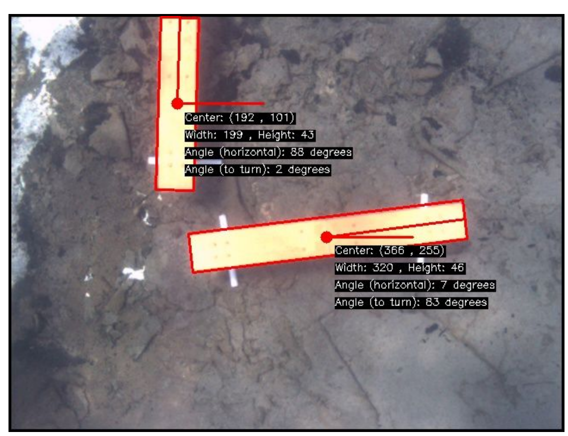

Fig 5. Our Pathmarker detection code running on a video frame. (Test Image Credit: CUAUV Robosub Team)[2][1]

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. [1]

4. White Marker Bins

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. [1]

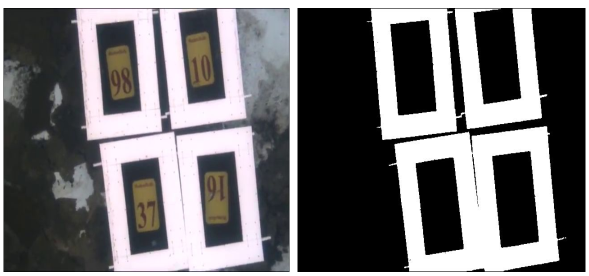

Fig 6. Ground Truth Segmentation images have been generated from available media to allow for improvements in image filter code to be measured. (Test Image Credit: CUAUV Robosub Team)[2][1]

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.[1]

Fig.7 The ENVI 2016 Vehicle "ENVI-UUV" was designed to accomplish four missions. [1]

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 [NVIDIA Jetson TX2] 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]

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. [1]

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.[1]

Fig 8. SJ4000 Waterproof Sports Camera [1]

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.[1]

(Editor's note: In July 2016, one of our SJ4000 cameras was replaced with a [MAPIR Survey 2](https://www.mapir.camera))

2. *Software*

Our team's vehicle is controlled by a [Python] middleware application running on a [Jetson TX2] *Companion Computer*. It's purpose is to provide high level instructions to the ArduSub firmware running on the PixHawk controller, [and to serve as a conduit for information flows between the various microservices. The decision was made to break the problem of vehicle autonomy down into several easily-managed subsystems, and to manage the flow of state information, a Redis in-memory database was selected to implement a Publisher/Subscriber pattern.] [1]

This middleware must coordinate processes and pipe messages between several subsystems including:[1]

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.[1]
2. [Localization.py] is responsible for [translating odometry information received from an open-source odometry application called "Semi-Direct Visual Odometry (SVO)" [3] and] generating NMEA 0183 GPS sentences to spoof the *ArduSub* firmware's Kalman filter.[1]
3. [Captain.py] reads a JSON file containing a sequence of desired waypoints and maximum time values before switching to the next task. [Captain.py] is also responsible for error-handling.
4. All PID control loops and [...] visual servoing code [are] stored in [Helm.py]. [1]

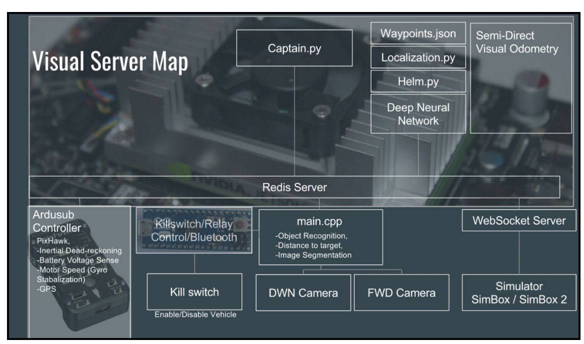

Fig 9. High level system diagram

3. SimBox and SimBox 2

SimBox(Figure 10) was originally created to give [first] and [second]-year college students an opportunity to hit "Fast-Fwd" and skip-over a mechanical submarine build so they could immediately start writing vehicle autonomy [or] computer vision code. [1]

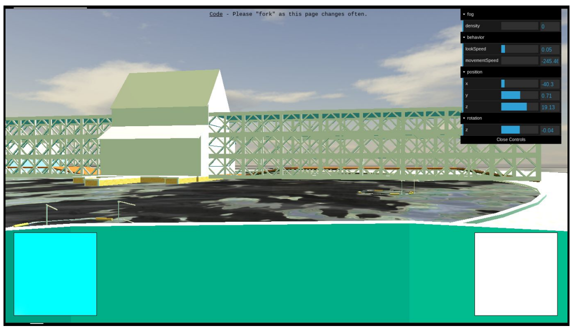

Fig 10. Textures and lighting of original SimBox

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 11).[1]

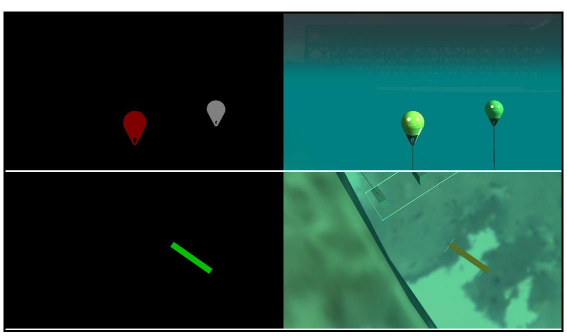

Fig 11. Automatic generation of *PASCAL 2012* formatted image segmentation training data

Our team published a dataset in our 2016 Journal Paper that was created with this tool. As we count down the days to this year's event, our fingers are crossed that some of this data was useful to other teams in their preparation for the 2017 Competition.

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 12).* [1]

Fig 12. 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.[1]

C. Experimental Results

Our team's vehicle has been waiting patiently in "dry-dock" for nearly a year since the 2016 Competition. This was largely due to challenges that were encountered in our attempts to finance necessary equipment purchases. Despite this setback, our Machine Learning team was able to gather some valuable experience with the TensorFlow Deep Learning framework using an old NVIDIA GTX 960 while they waited for the GPU procurement process. This experience with the difficulty of financing research also inspired our team to register a profile on XSEDE.org to facilitate future research grant requests.

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 [4].[1]

The DIGITS 3 web application outputs statistics and visualizations which assist in the process of identifying ways to improve network performance(Figure 19).[1]

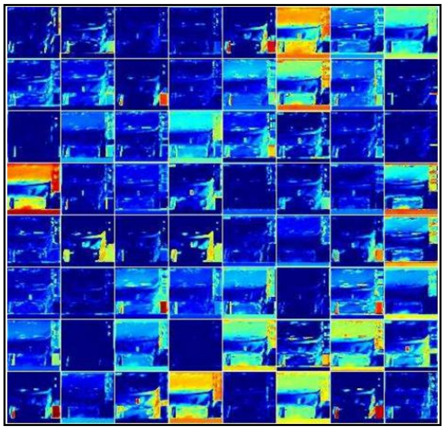

Fig 13. Example of a convolution kernel visualization[1]

In figure 13, 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.[1]

D. Acknowledgements

ENVI-UUV would like to thank: *Coleman University,* the *Electric and Networked Vehicle Institute (ENVI)*, [Jon Colby of LyteWi Internet Services,] the *San Diego Mesa College Associated Student Government(A.S.G.)*, the *San Diego Mesa College Interclub Council(I.C.C.), [Dr. James Burns,* and *the rest of our sponsors]* for their generous support, and financial contributions to this project.[1]

E. References

[1] Electric Networked Vehicle Institute Robosub Team (2016, June). "Coleman 2016 Journal Paper." [Online]Available: https://goo.gl/15n5QS

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

[3] Christian Forster, Matia Pizzoli, and Davide Scaramuzza. "SVO: Fast Semi-Direct Monocular Visual Odometry". In: IEEE International Conference on Robotics and Automation (ICRA). 2014

F. Outreach Activities

One of our team members was asked to speak to a group of high school students from Clairemont High School (video here). While it's difficult to know if an attempt to inspire an audience is successful, the students respectfully listened to the presentation. We hope that this means the presentation raised their level of interest in STEM subjects.

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. [1]

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. [1]

https://goo.gl/QIXss4

If you are considering participating in Robosub [2018](or later): We hope that these files are useful to you, and look forward to meeting you in person![1]