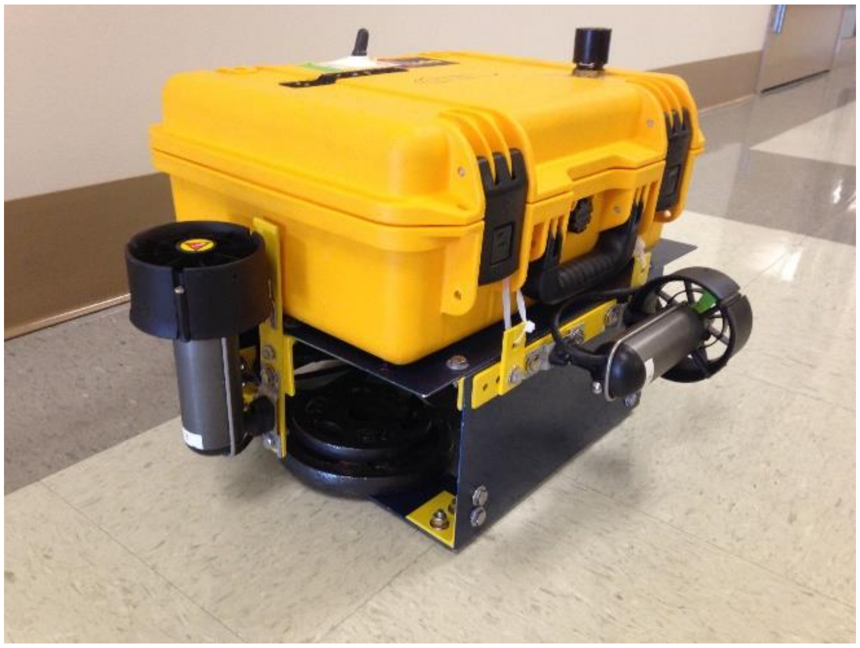
BlackFin
Embry-Riddle Aeronautical University
Autonomous Underwater Vehicle



http://www.roboticsassociation.org/RoboSub.html

Team Leader:

Donald Patrick Bennett Jr.

Team Members:

Robert Goring, Kelsey Klein, Chris Fahey

Faculty Advisors:

Dr. Charles Reinholtz
Professor and Chair - Mechanical Engineering Department

Dr. Brian Butka
Professor - Electrical, Computer, Software, and Systems Engineering Department
&
Dr. Timothy A. Wilson
Professor and Chair - Electrical, Computer, Software, and Systems Engineering Department

Embry Riddle Aeronautical University
600 S. Clyde Morris Blvd.
Daytona Beach, FL 32114

Abstract

The Robotics Association at Embry-Riddle (RAER) is proud to present its entry, BlackFin, for the 17th Annual International RoboSub Competition. The year's RoboSub team consists of several new members with mentorship being provided by students and faculty members who are well versed in the unmanned and autonomous systems regime. The team working on BlackFin consists of a handful of undergraduate students representing different majors from the College of Engineering at Embry-Riddle. The RoboSub competition consists of several tasks, which require navigation, image processing, and object manipulation. These tasks model the applications of current commercially available AUVs. The Embry-Riddle RoboSub team has designed a new vehicle from the ground up for this year's competition.

1. Introduction

The Association of Unmanned Vehicle Systems International (AUVSI) and Office of Naval Research (ONR) are hosting the seventeenth annual international RoboSub Competition to be held in San Diego, California at the SPAWAR Transdec facility. The goal of this competition is to create a vehicle that can autonomously complete a set of tasks, which are mimic the abilities of current autonomous vehicles in industry. To create a vehicle and system capable of accomplishing all of these tasks, requires years of development and testing. Recognizing the concept of complexity through iteration, the RAER RoboSub team has devised the goal of creating a simple system that can perform the required tasks, with room for future modifications, as determined from competition performance.

For this year's entry the BlackFin system has been mechanically overhauled for a smaller, more efficient design. The new frame is designed to be light and portable, while maintaining structural integrity and performance. This engineering decision has provided the team with a robust, mobile and durable platform.

2. Hardware and Mechanics

Despite the relatively small profile of our previous year's competition entry (Figure 1 top), compared to other vehicles in the competition, the team has set out on a goal of making RoboSub a backpack-able platform. The result (Figure 1 bottom) of this goal is a portable robust vehicle that is capable of performing all of the necessary tasks in the RoboSub competition.

The electronics, computer, and the majority of our sensors are housed in a pelican case mounted to the top of the vehicle. This packaging allows for easy transport and maneuverability of the newly designed vehicle.

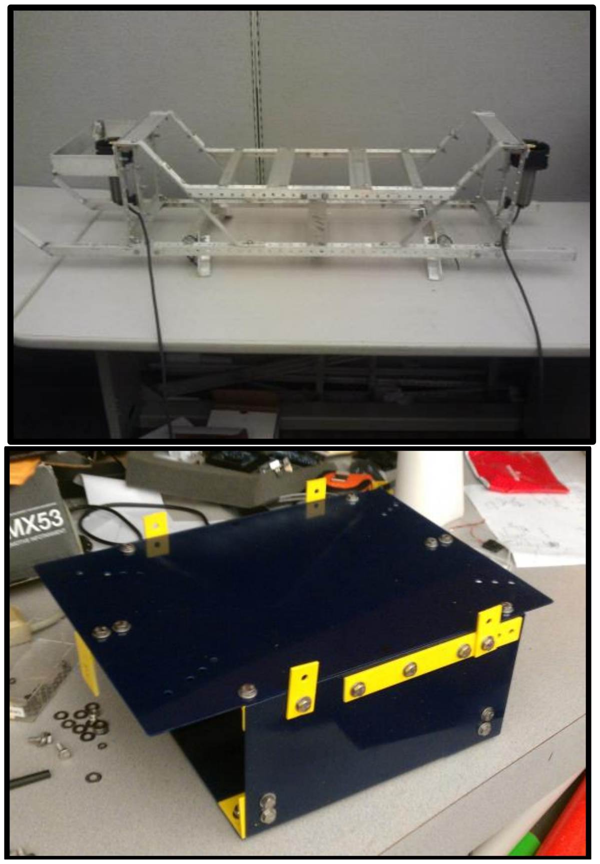


Figure1 - Old Frame (top) and New Frame (bottom)

2.1 Frame / Structure

A significant change in this year's platform is our smallest frame design yet. This frame is developed to be safe, simple, and extremely portable.

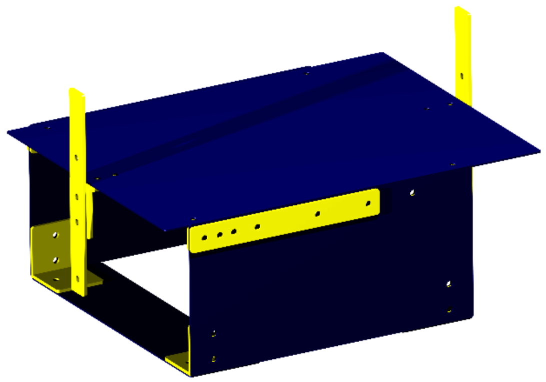


Figure 2 - CAD Rendering of Frame

Mechanical frame of the new vehicle is cut from sixteenth-inch plate aluminum and one-eighth-inch right angle extruded structure for support and motor mounts. The Pelican Storm IM2200 case that holds the electronics is secured to the top of the frame. The selected case has internal dimensions of 15 by 10 by 6 inches, providing plenty of room for our internal electronics.

The vehicle is designed to be statically stable, which simplifies maneuvers in the water. Majority of the vehicle's weight is concentrated on the base, giving BlackFin a low center of gravity, thus reducing pitch and roll.

2.2 Internal Layout

Majority of our electronics are placed in a waterproof pelican case, as seen in Figure 3.

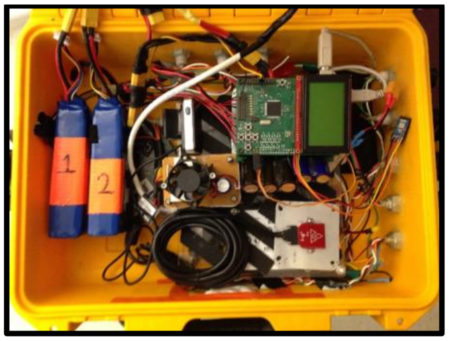


Figure 1 - Vehicle Internal Layout

The USB cameras used for vision processing are housed in separate waterproof cases. These waterproof cases are designed such that a small USB device could be housed inside them.

2.3 Thrusters

This iteration of BlackFin uses four thrusters, two vertical and two horizontal, for maneuverability. This configuration provides four degrees of freedom: translation (horizontal and vertical), yaw, and pitch. This setup was determined to have the optimal level of maneuvering freedom to perform most tasks. To account for future modifications of the BlackFin system, there are four motor mounting positions available for a holonomic configuration. Such configuration would allow for the vehicle to move freely within the horizontal plane.

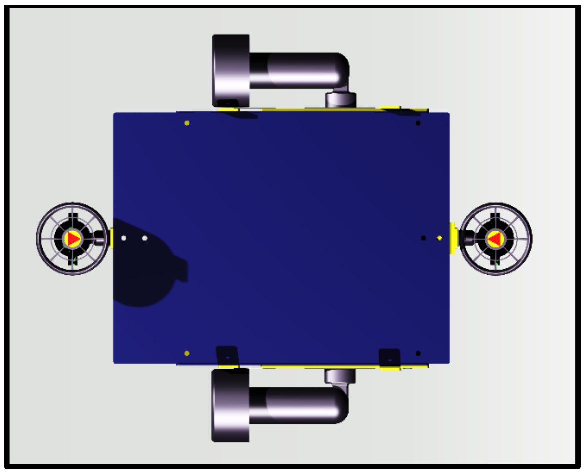


Figure 2 - Thruster Positions

Current motor configuration allows for simplified control algorithms. In this setup, motors are placed symmetrical to each other to prevent unexpected movements. To achieve reliable motion in the horizontal plane, only the side thrusters need to be balanced.

2.4 Connectors

For connectors to the motors and cameras deep sea rated in-line and bulkhead connections from SubConn are used. Each thruster uses two pins, power and ground, so the motors are paired together and use four pins of a five pin connector. The camera case is USB so it uses four pins out of a five pin connector. All of the connectors used are rated for 150 meters.

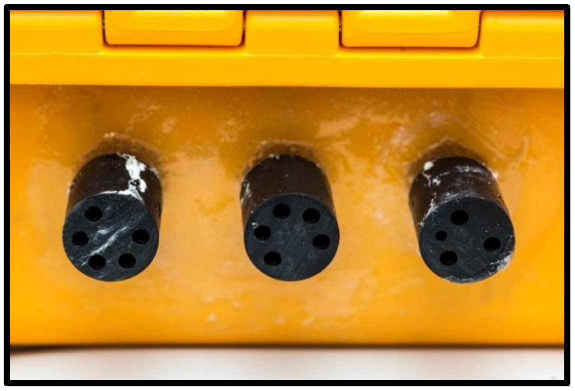


Figure 3 - External SubConn Connectors

2.5 Camera Mounting

For vision two Odroid 720p USB cameras are mounted facing forward and down. The cameras record at 30 frames per second, providing the sub with plenty of quality images for vision processing. Careful consideration was put into choosing our cameras, as image processing is easily the most important tool in completing the tasks.

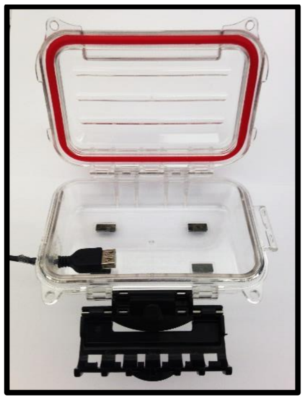


Figure 4 - Camera Case

2.6 Current Development

BlackFin does not yet have systems for completing all of the tasks. Currently, the droppers and torpedo launchers are being developed.

The dropper system in development consists of small 3D-printed torpedoes that are released by a servo. These 3D-printed torpedoes are hollow so that they can be weighted to fall straight.

The torpedo system consists of Toypedos that are spring launched and released by a servo. Having them spring launched ensures that they are released at a safe speed and won't cause injury to the pool or diver.

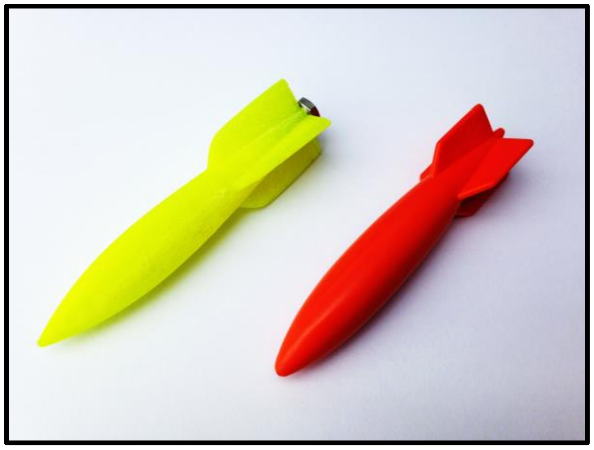


Figure 5 - Torpedoes for Dropping (left) and Firing (right)

3. Electronics

BlackFin's electronic system consists of purchased, donated, and homemade components. The electronics are designed to be compact and modular. All of the components that are not a part of the computer are mounted to the top of the computer case seen in Figure 8 using Velcro.

3.1 Computer

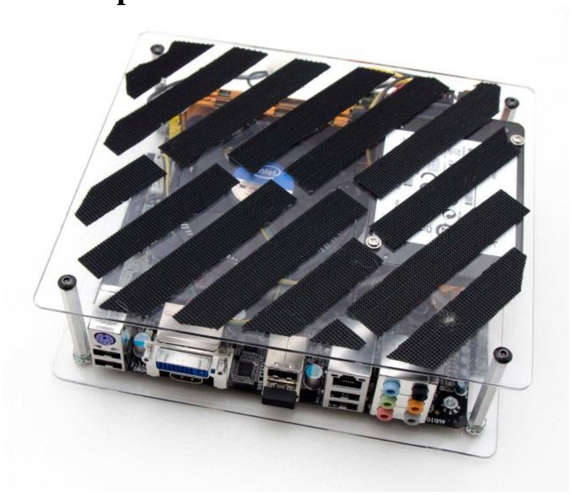


Figure 6 - Computer: Top

The computer consists of entirely off-the-shelf components. The computer specifications are seen below in Table 1.

Table 1 - Computer Specifications

Motherboard	Foxconn H67S
Processor	Intel® Pentium® CPU G630T @ 2.30GHz
Memory	8GB DDR3
Storage	80 GB Solid State
Power Supply	PICO-PSU

The computer is secured to the inside of the case with plenty of Velcro.

3.2 Computer Voltage Converter

The onboard 5 cell Lithium Polymer batteries output 21 volts. This is great for the thrusters but too much for much for the computer. Our homemade voltage converter takes in direct battery voltage and outputs 12 volts to the power supply, with a maximum output of 250 Watts. This voltage converter is shown in the following figure.

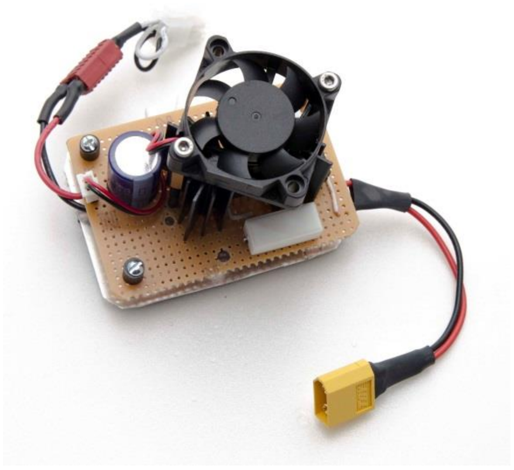


Figure 7 - Voltage Converter

3.3 VectorNav Compass

BlackFin uses a VN-100 Rugged Inertial Measurement Unit (IMU) to navigate in consistent directions. This IMU was generously donated to RAER by VectorNav. Fitting with our hardware goal, "the VN-100 Rugged is the smallest commercially available industrial grade IMU/AHRS."[1]

Figure 8 - VectorNav VN-100

[1] (VectorNav, 2014)

The data used from this sensor are the Euler Angles of the sub, yaw, pitch, and roll. The main features of this IMU are its .02º RMS precision measurements, simple 2D magnetic field calibration, and ability to automatically filter out soft and hard iron disturbances.

3.4 Main Interface Board

On the top of our board stack (Figure 11) is our interface board, custom designed and built by a student. The board connects over USB and serves the following functions:

- Outputs vehicle status on the display (battery voltage, temperature, etc.)
- Reads analog values from sensors.
- Monitors battery voltage and current draw.
- Drives servos.

This custom made board features:

- 10 PWM outputs
- 10 analog inputs
- I2C
- Micro SD slot
- 5 buttons
- LCD Screen

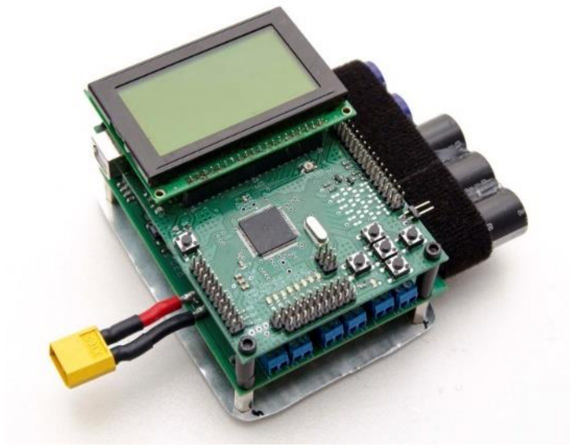


Figure 9 - Main Interface Board

3.5 Motor Controller Board

The motor controller board supplies power to the top board for any servos and controls up to six thrusters. This custom-built board is capable of supplying -18 to +18 Volts at up to 15A to each of the six bi-directional channels.

The board accepts a battery voltage of up to 40 volts, ensuring that all of our LiPos are safe to use with it. The board is controlled through a USB cable to the main computer.

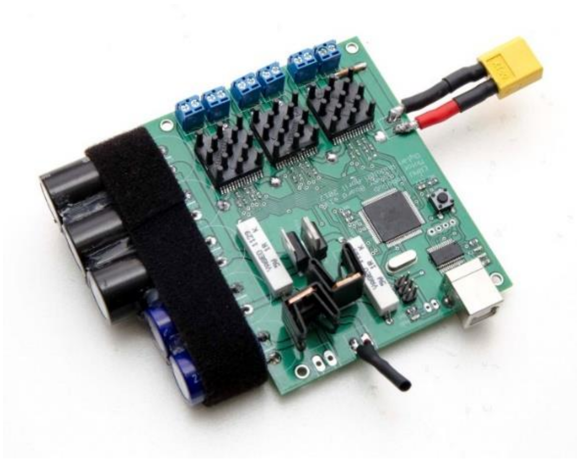


Figure 10 - Motor Controller Board

4. Software and Controls

The main software packages used on the subs are as follows: Python 2.7, NumPy, OpenCV, and matplotlib, as well as the Arduino IDE. Python was chosen as the primary development tool because it’s easy, quick to develop on, and intuitive to read. Many of the members the team acquires only have programming experience in MATLAB, so it was important for us to have a language that is easy to understand. All of the software on the main computer of the sub is written in Python. NumPy is used for array and matrix manipulation. OpenCV is used for vision processing and part of our GUI. Matplotlib is used to graph all of the data the sub records. Finally, the Arduino IDE is used for writing the firmware of our main interface board and motor controller board. All of the software is developed on Windows 7, but should be compatible on most platforms, due to the high portability of Python.

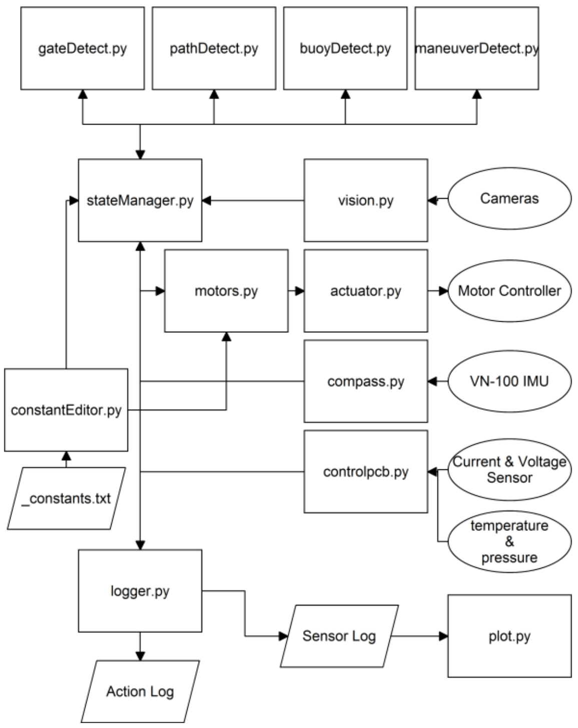


Figure 11 - Software Diagram

4.1 Software Design

BlackFin's software system can be broken down into four categories: hardware communication, data processing, data logging, and value manipulation.

4.1.1 Hardware Communication

The motor controller board and main interface board are interacted with using the Python library PySerial. Through actuator.py, each of the four motors can be independently controlled via a PWM signal. The motors.py script handles higher level movement commands, translating them into individual motor movements. Both the AttoPilot current and voltage sensor and the BMP – 085 barometric pressure and temperature sensors are read using controlpcb.py. This script communicates serially to the interface board to return the sensor value. Compass.py interfaces directly with the VN-100 IMU via serial RS232. Images are retrieved from the camera using OpenCV.

4.1.2 Data Processing

All data from the sensors is processed through the state manager. Through a set of conditional statements and the vision processing scripts (gateDetect, pathDetect, buoyDetect, maneuverDetect) the state manager determines what actions should be sent to motors.py. The vision processing scripts interpret a target heading and depth, and the state manger communicates this to the motor control script which uses the current heading to determine motor power changes.

4.1.3 Data Logging

New to this year's vehicle are action logs and data logs. Writing to any of the logs is handled through logger.py. On startup, both logs are automatically created. The data is written directly to text files, instead of through a buffer, to ensure that if a crash is to occur, the logs are intact.

All action logs names have the format of month, day, year, hour, minute, and second (6_25_2014_16,7,19.txt). Every action ranging from connecting to a communication port to changing the power of a motor is logged with a source and a timestamp. The files are easily viewed and organized in a spreadsheet reader.

The sensor logs function much like the action logs. They are comma delimited text files that are named with the current time in seconds. On average, the data from every sensor is logged every half second. These logs have been crucial for finding relationships between some of our data such as internal case temperature and pressure.

From several days of testing logs a strong set of data which showed the relationship between internal case pressure and temperature while at the surface. The linear relationship found between these two values allows us to reliable determine our depth even though our barometer reads case pressure.

4.1.4 Value Manipulation

Many of the tasks which are performed require careful tuning which may not be effective in all environments. To facilitate expedient value tuning, the constants used in the motors for calibration, heading and depth control PI controllers, thresholds in vision and more can be altered by loading a text file using constantEditor.py.

4.2 Micro-Controller Level Software

The software that interprets commands within the micro-controllers is very simple in its approach. To read from sensors, a call and response protocol is used. A single character is sent from the main computer via RS-232, to which the interface board responds with the value. To move a motor a character and a value is sent. BlackFin's computer system contains two interface boards based around the ATMEGA1280 microcontroller. The first interface board is capable of the following tasks:

- Serial communication with main computer.
- Read analog signals from voltage and current sensor.
- Communicates to pressure sensor via I2C.
- Status updates to onboard LCD display.
- Log data to Micro-SD Card.

The second microcontroller is embedded on the motor controller. This board is in charge of the following tasks:

- Provide external power to servos and auxiliary boards.
- Provides PWM signal to control main thrusters.
- Communicates with main computer via RS-232.

Overall these two boards take care of all low-level computations and input/output. If needed their firmware can easily be updated via the Arduino IDE.

4.3 Software Algorithm

This section contains the software processes that are used to control the vehicle through the competitions tasks. At a high level a simple finite state machine can model the process. All code described in this section was written in Python, and is composed of in-house written scripts and libraries.

1. Load waypoints folder containing each point of interest, and the angle between them. This information is pre entered based off the condition of the pool.
2. Using file as a reference, head in the direction of the next waypoint, while using the cameras to attempt to lock onto the object. If the target can be seen, head towards it. If not hold heading and continue forwards. A simple PI controller is used to maintain heading. Additionally, depth is also monitored though each control cycle. If the measured depth is out of the accepted range, a correction is made using a P-controller.
3. If waypoint is located, run the script associated with that task, update the waypoint location, and return to the main control loop.

4.4 Vision Processing

It was determined that vision processing is the best solution to object detection. The onboard cameras were selected due to their accurate color detection. This decision was based off of many years of research using varying types of cameras. By having an image with few color defects, a threshold can easily be applied. To increase the color accuracy of the image, the OpenCV is used to normalize the histogram. This ensures that the lighting conditions between images are consistant. A binary threshold is then applied to the Hue, Saturation, and Value (HSV) planes of the image. Once a binary threshold is applied, contours are used to get an array of onscreen objects. Once a list of objects is made, they are sorted by area, location, and shape. This method of computer vision can be seen in Figure 14. In addition to testing vision algorithms on new images, we use past footage from the Transdec pool. This is done due to the water clarity differences between the competition pool and the pool at Embry-Riddle. The top image demonstrates our algorithms ability to find the gate. The bottom image shows the usage of contours. The outline of found images is shown.

The vision software was tested and modified in several varying lighting conditions. This allows for us to be able to tune the environmental variables so that the algorithms to work in nearly all conditions.

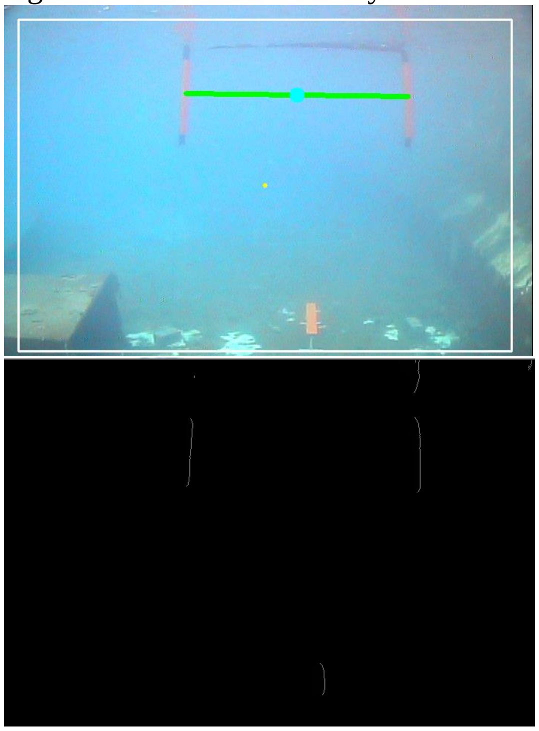


Figure 12 - Detected Object (Top) Threshold Contours (Bottom)

5. Remote Debugging

For testing purposes BlackFin can be remotely controlled. An Ethernet tether can be connected to the vehicle to use Window's Remote Desktop application to log into the sub's main computer. Once connected the vehicle can be controlled, as well as viewing the output from cameras and sensors through our self-made GUI.

6. Conclusion

BlackFin is a fully autonomous underwater vehicle designed and manufactured by engineering students at Embry-Riddle Aeronautical University. In addition to being designed to compete in the AUVSI RoboSub competition, BlackFin was also designed to be a research platform. We believe that BlackFin is an excellent demonstration of systems engineering. Through careful planning and implementation all hardware and software systems were able to interact together with no problems. Through the design and development of this platform, we as a team were able to grow significantly.

7. Acknowledgments

We would like to thank Embry-Riddle Aeronautical University's College of Engineering specially the Department of Electrical, Computer, Software, and Systems Engineering and the Mechanical Engineering Department for their continued and substantial support of Team BlackFin. We would especially like to thank our faculty advisors for all their support, encouragement, and advice. We would also like to thank all of our sponsors for the outstanding support and donations that have made this project possible.

8. References

VectorNav. (2014, June 26). *VN-100 Rugged IMU / AHRS*. Retrieved from VectorNav: http://www.vectornav.com/products/vn100-rugged