All-Around Above-Water Sensing for Recreational Boaters

Phase 2 Report

Michael DeFilippo[1], Michael Sacarny[1], Paul Robinette[1,2]

[1]AUV Lab at MIT Sea Grant, [2]University of Massachusetts Lowell

May 20, 2020

Abstract

This report details the progress of Phase 2 of our project with Brunswick Corporation in coordination with the Mercury Marine division to increase situational awareness of recreational boaters. In this phase, we integrated marine radar, a lidar, two cameras, and an infrared camera on board the Boston Whaler R/V Philos. We tested these sensors and collected time-synced marine perception datasets. We developed scripts to process and analyze the datasets. The scripts combine all data into a rosbag format, a standard log format used by Robot Operating System (ROS), which allows us to take advantage of the ROS based RVIZ data visualization program. We installed infrastructure throughout the boat to mount the sensors, as well as installing Ethernet and 120 V power throughout R/V Philos. Additionally we further developed the autonomous capabilities of “RoboWhaler” to test lidar based object detection and avoidance, as well as testing that the new SBG Systems INS was working as planned.

Objective

All vessels on the water need to maintain constant vigilance to avoid both moving and stationary objects. Vehicles with some amount of autonomy have an obvious need for a sensing solution to this problem, but piloted vessels will also benefit from technology to assist in avoiding collision. Currently, piloted vessels typically use radar, AIS, charts, and human eyes and ears to spot obstacles and other vessels. In this work, we begin developing software that leverages existing open-source efforts in sensor processing and marine autonomy to fuse several forms of sensing data into a coherent list of obstacles and vessel contacts.

Figure 1: R/V Philos with marine perception sensor suite installed

This project has general applications to recreational boaters, but our primary testing platforms are the REx 4 robot and the Boston Whaler R/V Philos (Figure 1). REx 4 is a 17 foot pontoon vessel (WAM-V by Marine Advanced Research, Inc) that has been under development by MIT since 2014. It is capable of remote or autonomous operation in the inland waterways around Boston as well as offshore in favorable weather conditions. R/V Philos is a 25 foot Boston Whaler on loan from Brunswick/Mercury to aid our development of autonomous systems. In separate work with Mercury, we have developed an interface (“RoboWhaler”) to allow our autonomy system to interface with and control R/V Philos.

In brief, our phase 2 goals were to:

1. Integrate new sensors onto R/V Philos
2. Develop ROS based software to interface with sensors
3. Install sensor and autonomy infrastructure on R/V Philos
4. Develop data replay programs
5. Develop data fusion algorithms
6. Gather marine datasets
7. Test and prototype RoboWhaler autonomy

Summary: Phase 2 Progress

For Phase 2, we focused on continued development of R/V Philos for vessel autonomy (i.e. “RoboWhaler”) and as a mobile research platform particularly suited to collection of marine datasets. In the beginning of the year, we completed development of software for marine radar. Mercury Marine engineers upgraded the Philos IMU, control software, GPS antennas, and wiring harness and installed marine radar to enhance vessel autonomy. We wired the vessel for 120V power, ethernet, and made other infrastructure improvements, then installed the initial marine sensor suite, consisting of two cameras, one infrared camera, one lidar, and one radar. We used RoboWhaler to demonstrate low-speed marine obstacle detection and avoidance. Finally, we collected data for multi-modal marine sensor datasets, focusing on diverse weather and traffic conditions. Real-time data visualization and data playback is accomplished using the open sourced ROS based RVIZ program. For Phase 2 development we have:

- Developed a marine perception floating laboratory,
- Collected multiple marine perception datasets in various locations, traffic settings, and weather conditions,
- Processed the collected datasets to be hosted for public distribution and algorithmic development, and
- Continued development of RoboWhaler, the fully autonomous MIT/Brunswick robotic Boston Whaler.

Later in 2019, we began processing the data into polished datasets, including time synchronization, image correction, clarification of units and frames of reference, conversion to common data formats, and creation of metadata. We are ready to host the first six datasets now on our internal server for public access.

Integration of Sensors onto R/V Philos

Much work went into creation of a capable marine data collection platform. We focused on integration of the sensor suite not only for data collection, but to be applied to advanced safety and autonomous navigation in the future. We made considerable modifications to the vessel to accommodate mounting and networking of the sensor suit, such as sensor power, computer power and mounting, and ethernet networking. Considerable software development was required to integrate multiple sensors, computers, and robotic middlewares (ROS and MOOS) into a functioning real-time system. Mission-related software was developed to enable data collection and obstacle avoidance. Real-time object detection through point cloud processing was demonstrated for obstacles <100m and will be further developed for those >100m as follow-on development.

Sensor Suite

The current sensor suite consists of the following instruments (Figure 2) and ultimate time-stamped data representations:

- 1 VLP-16 Velodyne 3-D Lidar
    - Point Cloud format
- 1 Simrad Broadband 4G Radar (two channels)
    - Point Cloud format
    - Spoke returns
- 2 FLIR Blackfly 1.3 MP Cameras
    - Compressed images

- 1 FLIR ADK Infrared Camera
    - Full resolution images
- 1 SBG Systems Dual Antenna RTK INS
    - IMU and GPS parameters in CSV

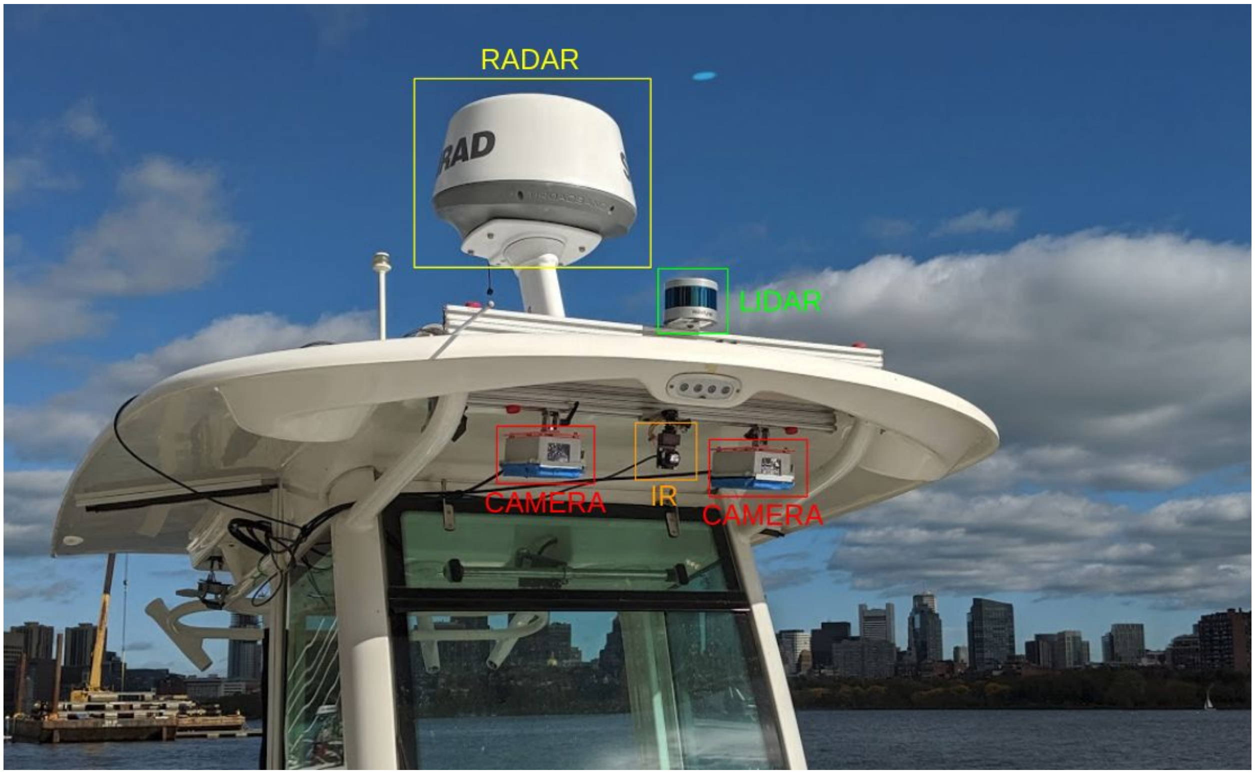


Figure 2: R/V Philos 2019 Marine Sensor Suite

Sensors Infrastructure

- We added hardware to the vessel to support sensors and computing including:
    - 2 kw pure sine wave inverter powered by auxiliary batteries
    - 12 to 48VDC converters for POE injection
    - Watertight combination 120 VAC/Ethernet ports fore and aft below the hard top for convenient access during missions
    - Ethernet switching
    - Strap-in shock mounting for the processing PC in cabin
    - Rails for mounting of sensors
    - Acquisition of 18 inch daylight monitor

As part of sensor integration, we documented the sensor installation and orientation of Camera/IR, laser, and radar mounts. As part of the integration and testing we had to work through some issues that arose during development:

- Mercury installation of new IMU and subsequent troubleshooting due to likely-GPS multipath distortion.

- Camera housings that did not allow for yaw rotation, only pitch. This resulted in reduced Horizontal Field of View (HFOV) coverage
    - Current housings have been modified for yaw rotation to allow for greater HFOV coverage with combined camera images.
- Camera image bandwidth issues due to limitations of Ethernet switching
    - Fixed offseason by replacing the Ethernet switch with a four port 10Gb POE networking card for the perception computer installed on the vessel. This allows each Blackfly POE camera to be directly wired into the perception computer, eliminating network collisions.

Software Development: Sensor Drivers and Data Fusion Algorithms

Integration of the sensors onboard R/V Philos required significant driver development and parameter tuning. The Simrad Broadband 4G radar lacked a driver for both MOOS and ROS.. The MOOS driver was developed as part of Phase 1 research. Lessons learned from this driver drove the development of the ROS driver. The ROS driver was developed after the 2019 data collection season due to time constraints, but has been tested in the lab and is now fully integrated in our ROS sensor data collection scheme.

In the ROS environment sensor drivers already existed for the Velodyne Lidar, FLIR Blackfly cameras, and the FLIR IR camera, but each sensor required a significant amount of parameter tuning in order to get them to operate in an efficient manner. All sensor driver parameters are set in launch files that were created to launch each sensor and record the associated data. The Velodyne driver was the least labor intensive and pretty much works as-is out-of-the-box. One particular issue that we did see was that of going from our original Velodyne HDL-32 to the provided VLP-16 lidars. On the HDL-32 the Y-axis is forward opposed to the VLP-16 which is X-axis forward. Some software changes were required to change which axis is forward for data processing and object detection. The FLIR IR camera works out-of-the-box with the downloaded driver, but requires updates to the operating systems UDEV rules to properly interface with the camera on startup. The FLIR Blackfly POE cameras required the most development time out of the sensor suit. These cameras are very capable cameras, but have around 40 parameters that can be tuned for optimal performance in any given environment. The main issue with the Pointgrey ROS driver for the Blackfly cameras is that most parameters are not documented nor exposed via arguments. Consequently, the default setting for the camera buffer is set so low that any frame rate greater than 5 FPS causes the system to drop packets or freeze. The issues with this driver are now understood and have been resolved for the 2020 data collection season.

To interface and control the marine perception sensors, record all data, run the autonomy middlewares (MOOS and ROS), and interface with R/V Philos a fully functional Linux-based desktop computer was developed from the ground up. The computer was designed with power constraints in mind, but with significantly more CPU/GPU capabilities than is available with top-of-the-line embedded systems. This is to allow for maximum data processing power and allow for CPU/GPU vision processing algorithms to be developed and implemented without any hardware limitations.

The perception computer runs both ROS and MOOS robotic middlewares and handles the communication internally for all processes. ROS was selected to ease development of sensor drivers and also take advantage of the integrated Point Cloud Library (PCL) tools for point cloud processing. MOOS provides the mission behaviors that are used for autonomous missions such as waypoint following and object avoidance. To communicate between MOOS and ROS, applications were developed in each environment to pass data messages back and forth. Within these applications, data is passed as plain-text strings over a network connection similar in format to NEMA187.

Object detection through point cloud processing is accomplished through algorithm development within the Point Cloud Library. Proof-of-concept detection was on Lidar data. We are in the process of testing these algorithms with the radar point cloud data and plan on fusing the two point clouds together to expand our object detection horizon. To detect objects using PCL we

1. Filter the point cloud in the XYZ-axis for specific distances
2. Project the point cloud down to the XY plane
3. Downsample and cluster the pound cloud, and
4. Send clustered contacts to MOOS for object avoidance behaviors.

Gathering Marine Datasets

Over the course of the late Summer and into December we gathered data of typical marine subjects of interest in diverse weather conditions, such as:

- Multiple missions on the Charles River including:
  - Marine traffic of more than 30 sailboats and kayaks
  - Autonomous figure-8 slalom around bridge pylons
  - Transit through drawbridge raising and Charles River locks out to Boston Harbor
- Multiple missions in Boston Harbor including:
  - Marine traffic of moored and underway sailboats, commuter boats, dredges, barges, tankers, cruise ships, and small marine craft
  - Underway at planing speeds around the Boston Harbor Islands and out into Massachusetts Bay; turning in proximity to other vessels
- Multiple weather conditions:
  - Clear sky, cloudy sky, dusk, rain, fog, and snow (Figure 3)

Figure 3: Gathering marine datasets during a snowstorm

Postprocessing and Hosting of Marine Datasets

Once collected, the data had to be consolidated into consistent and accessible data sets. We created a series of Python scripts to perform this processing, including:

- MOOS variables to ROS message conversion
- time synchronization
- image correction (rotation)
- point cloud correction (rotation about axis)
- rendering of radar images
- translation of radar into point cloud
- creation of Lidar PCAP files
- clarification of system of units and frames of reference
- conversion to common data formats (eg CSV)
- consolidation into ROS bag files

Once processed, we created metadata files, a description of data formats, and a selection of images representative of the dataset.

MIT Sea Grant is now in the process of posting selected marine perception datasets from 2019 for academic and industry use to a publicly accessible web-server. The intent is to present variations in weather, lighting, marine traffic conditions, and varied speeds from an inshore to offshore perspective (Charles River through Boston Harbor). All data is available in Rosbag format that can be visualized and modified using the open source Robot Operating System (ROS) and related tools, such as RVIZ (Figure 4). Selected data is available in csv format, but the intention is for most researchers to work directly with the Rosbag files.

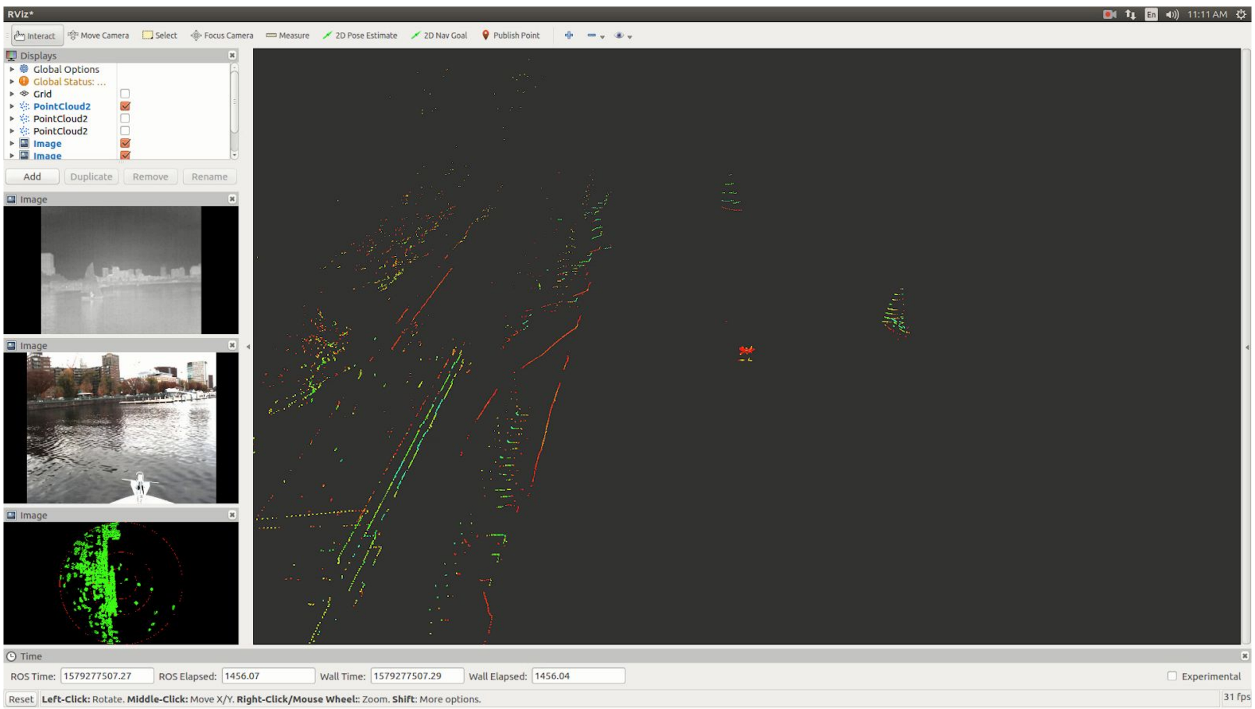


Figure 4: Post processed data visualization of rosbag data format using RVIZ

These datasets will be freely available to help spur development of robust algorithms in marine autonomy. We intend to engage and encourage sharing of algorithms and insights with users of the data. The selected highlighted datasets (Appendix A) will be hosted by MIT Sea Grant at the following link https://seagrant.mit.edu/auv-lab-data/#marine-perception.

RoboWhaler

In 2018 in conjunction with Mercury engineers we developed an interface between the MOOS-IvP autonomy system and the Boston Whaler’s control software. Initial tests were conducted to prove that autonomy inputs could be given to command the Whaler to follow a path of GPS waypoints. The initial tests were successful and the project was given the name RoboWhaler as a way to identify development of autonomous capabilities aboard the vessel.

While this project was not part of the Phase 2 proposed tasks, we consider the build out of a fully autonomous Boston Whaler an important way to test autonomous safety capabilities, and very much a related task. As part of the continued build out of autonomous behaviors, Mercury engineers upgraded and installed a state-of-the-art INS system with dual GPS antennas. After we completed the install of the perception sensor suit and perception computer we conducted multiple tests to validate that the system was working as planned and to prove that we could identify obstacles at low speeds and safely avoid them.

By the end of the testing season we were able to prove that we could identify and avoid obstacles. When we started the tests we found that the INS was not properly receiving the GPS signal from the dual antenna system. This led to the inability of the INS to provide a navigation solution that provides the correct pose of the vessel. The issue seems to be a GPS multipath issue and was resolved by using a different set of antennas. This is something that we will be watching this year and coordinating with Mercury Marine engineers to further troubleshoot and correct. A second issue that we found was with the MOOS pAvoidObstacle behavior (Figure 5). This is the behavior in MOOS-IvP that creates an avoidance buffer around an obstacle. The problem arose mostly for oblong obstacles, such as bridge pylons. The buffer zone would be correct at the ends but would stretch and shrink around the longer potions of the pylons. This was resolved with some code modifications to address these problems. We were only able to test obstacle avoidance at lower speeds (<2 m/s) as the control of the vessel at higher speeds in autonomous mode is not yet optimized.

Figure 5: Blue obstacle avoidance buffer produced by the MOOS-IvP behavior pAvoidObstacle

Marine Perception Floating Laboratory

In 2018 we started collecting marine perception datasets with REx 4 and learned much from using this robotic platform. Unfortunately the platform comes with limitations in range and power, which ultimately limited our ability to collect meaningful datasets beyond the Charles River and Inner Boston Harbor. Our main goal in 2019 was to create a marine perception floating laboratory that could be used not only to collect data in these locations, but also out into the open ocean. Detailed previously, we accomplished this aboard R/V Philos, and were able to collect data from the Charles River, in Boston, MA out to Massachusetts Bay. Now that the platform is built-out for both dataset collection and autonomous missions we plan to include the platform extensively in proposal development for future autonomous marine systems grants.

We will again use R/V Philos as a data collection platform with a few upgrades we received this winter. Firstly we are going to mount the daylight-readable monitor aboard the vessel. This will greatly improve our ability to run missions aboard the vessel and not need to rely on our laptop monitors. We then plan to collect new datasets with an upgraded sensor suite (Figure 6), in particular we have now added three forward facing FLIR Blackfly cameras along with two forward facing FLIR ADK IR cameras. The two IR cameras will provide approximately 140° forward facing HFOV and the three cameras will provide approximately 140° forward facing HFOV. The cameras will be used in conjunction with the previously installed 360° HFOV lidar and radar sensors to provide the marine perception datasets.

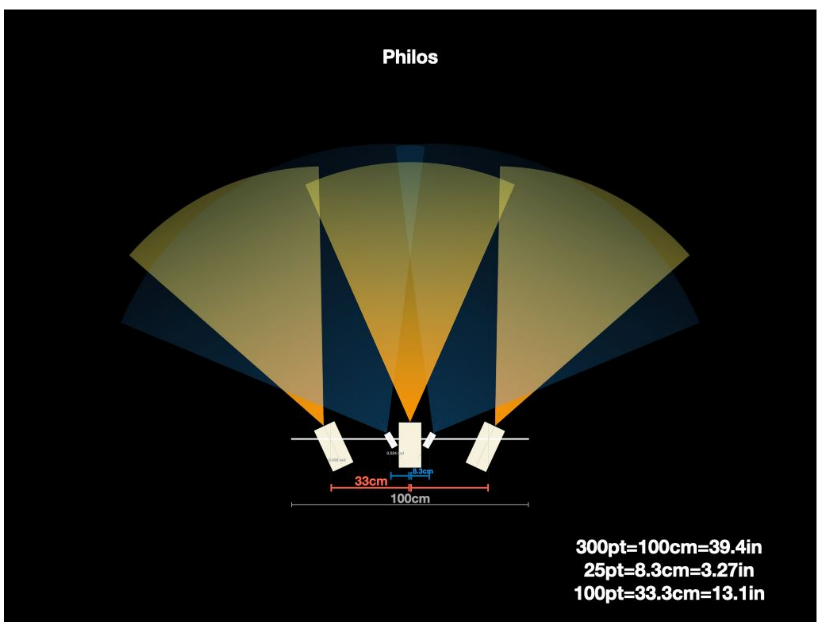


Figure 6: The three cameras are represented by the larger rectangles with each 48° HFOV represented by the yellow cone. The two IR cameras are represented by the smaller rectangles with each 75° HFOV represented by the blue cone. Each sensor has a small overlap such that we are able to combine images into a total forward facing coverage of 140° HFOV.

Next Steps (Summary of Summer/Fall 2020 activities)

As we proceed into Summer 2020, we plan on continuing both data collection activities, as well as improving autonomy aboard RoboWhaler. For RoboWhaler, we plan to first address the issues we saw last season caused by the INS. We will continue to work with Mercury Marine engineers to improve autonomous control of the vessel at higher speeds. We are planning to fuse the point cloud data provided by the laser and radar to improve all around obstacle detection capabilities of the vessel.

During the post processing of the datasets from 2019 we noticed that camera data would at times slow to lower than 12 FPS and sometimes drop out completely. The camera bandwidth problem has been addressed and corrected as previously mentioned. Another issue that we found during post processing was that the radar was not operating at 48 sweeps per minute as per the operating manual. The resulting point cloud data is such that there is a noticeable lag in radar returns versus lidar returns. This issue has been addressed and will be corrected for the 2020 data collection season.

We plan on having multiple students working on data fusion and marine obstacle detection over the course of the year as well. Students will be focusing on identifying objects in camera data from fusing point cloud data with camera data (Figure 7). Another push will be to use modern machine learning algorithms and modify them for the marine domain to identify objects and vessels (Figures 8-9) in camera video streams. Lastly we will be working with students to modify local path planning algorithms to be used on board our marine vessels for enhanced maneuvering around both static and dynamic obstacles.

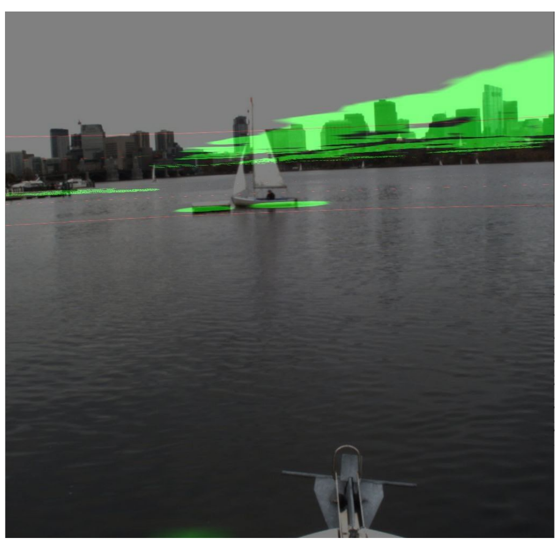


Figure 7: Fused radar image data with camera image

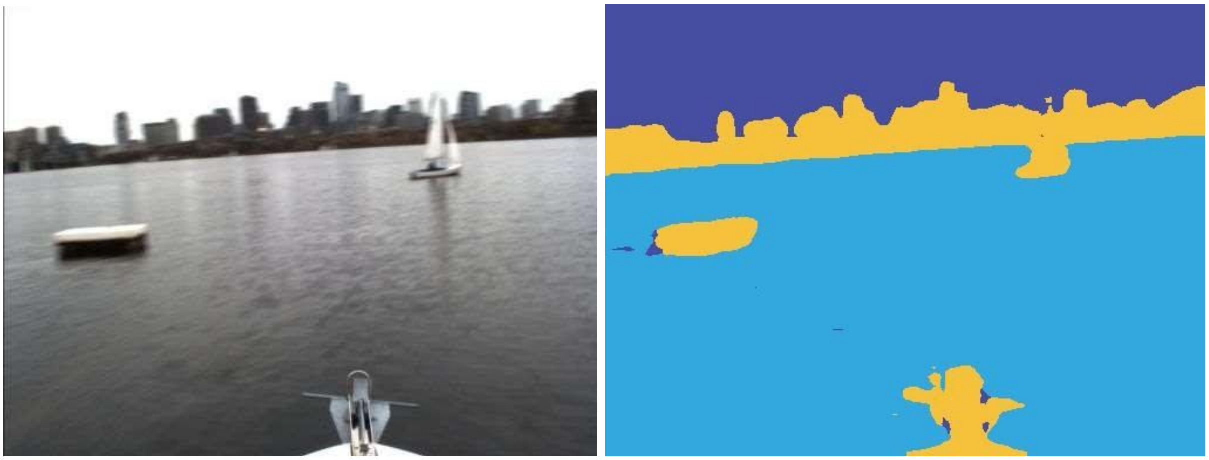


Figure 8: Left Image: Camera image of floating obstacle and sail boat
Right Image: Machine learning scene segmented output. Pixel representation, Purple = Sky, Yellow = Obstacle, and Blue = Open Space

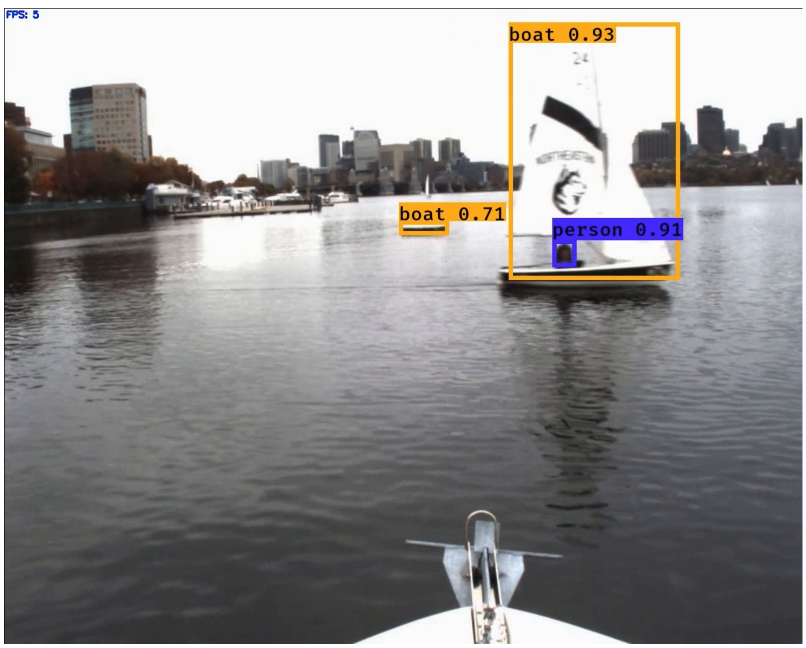
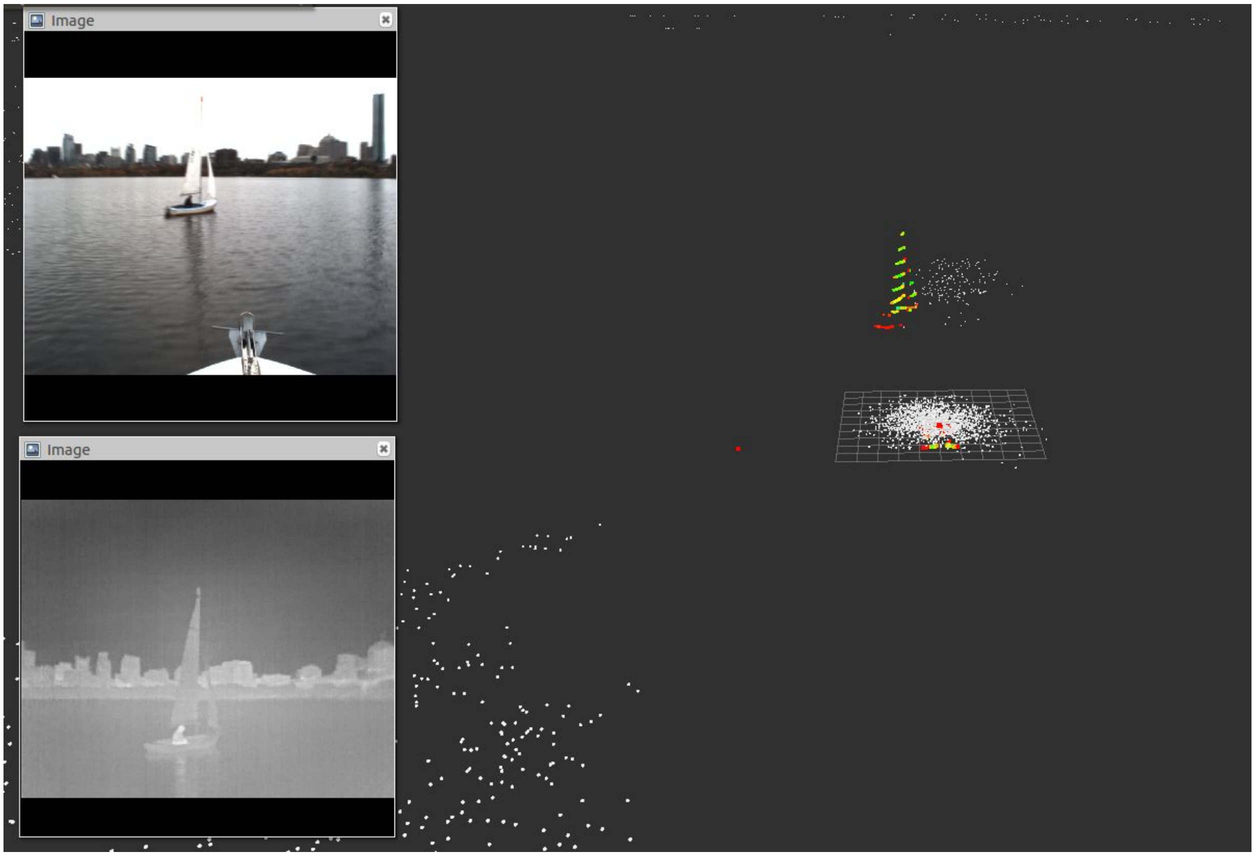


Figure 9: Machine learning algorithm for identifying and labeling objects

Appendix A: Select 2019 Images



Images from Philos (cloudy)- Top left: (Video) Sailboat dead ahead. Bottom left: (IR) Sailboat, crewman visible. Right: (LIDAR/radar fusion) LIDAR of sailboat followed by point cloud of radar. Philos is at center of grid in radar cloud

Image from REx - (IR) Harvard Bridge, abutments, pedestrian

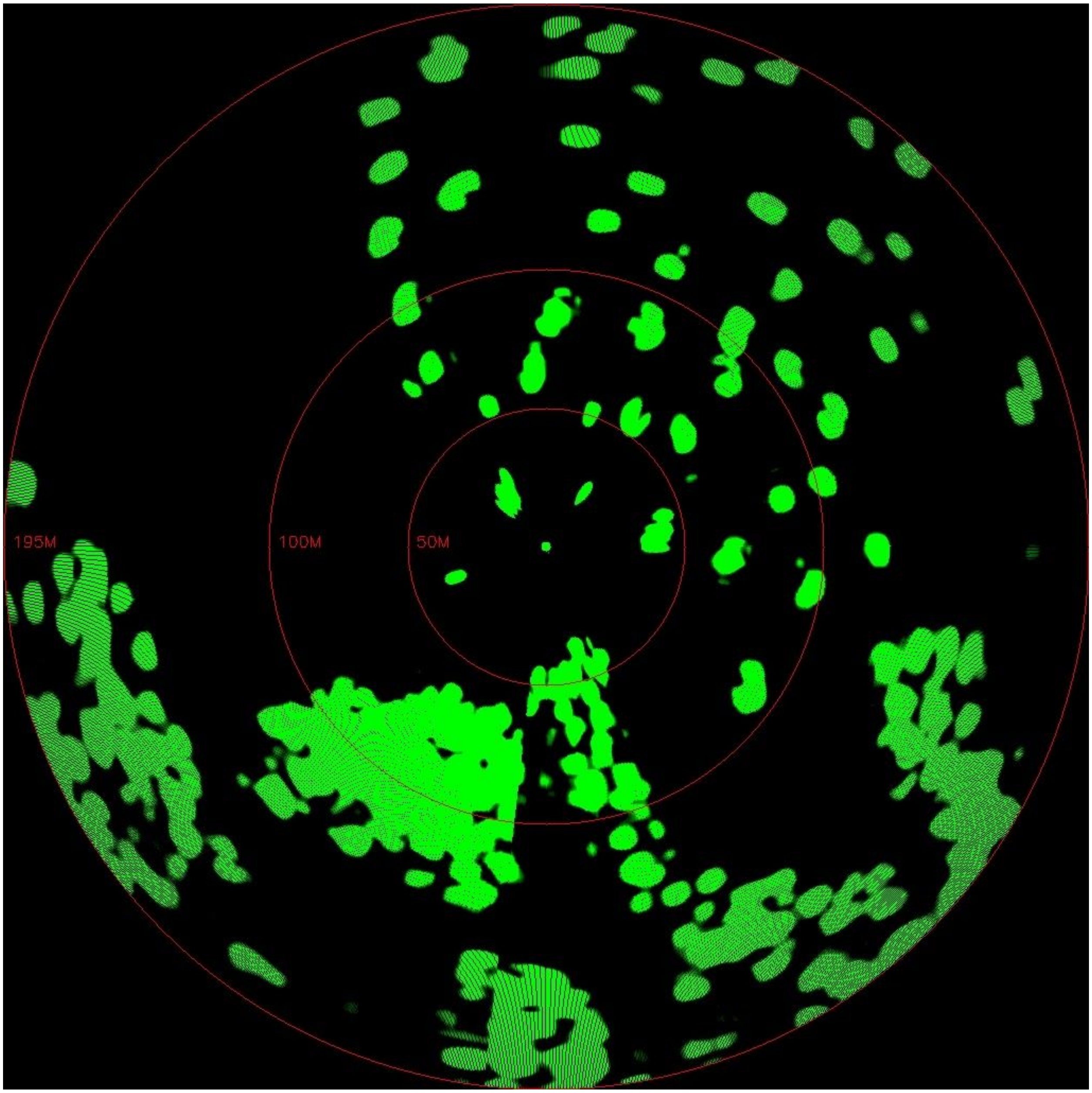
Images from REx (cloudy): (LIDAR) Harvard Bridge, annotated

Images from Philos(snow squall clearing)- (Video) Hard turn at plane, snow on lens, tug and fuel lighter

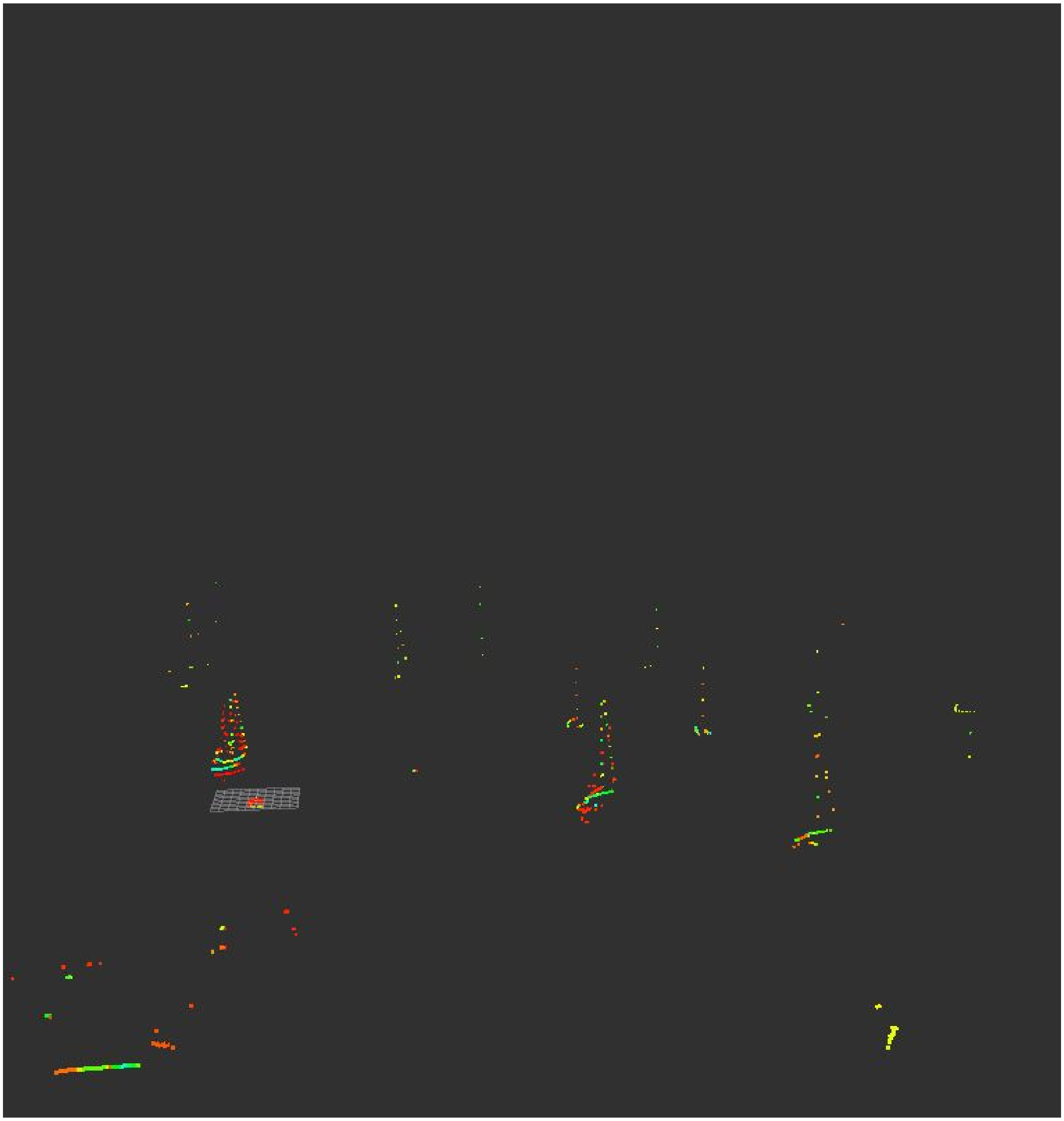
Images from Philos(snow squall clearing)- (IR) Hard turn at plane, snow on lens, tug and fuel lighter

Images from Philos(squall passed, now clear)- (Video) Hard turn on tug and fuel lighter

Images from Philos (cloudy)- (Video) Moored boats

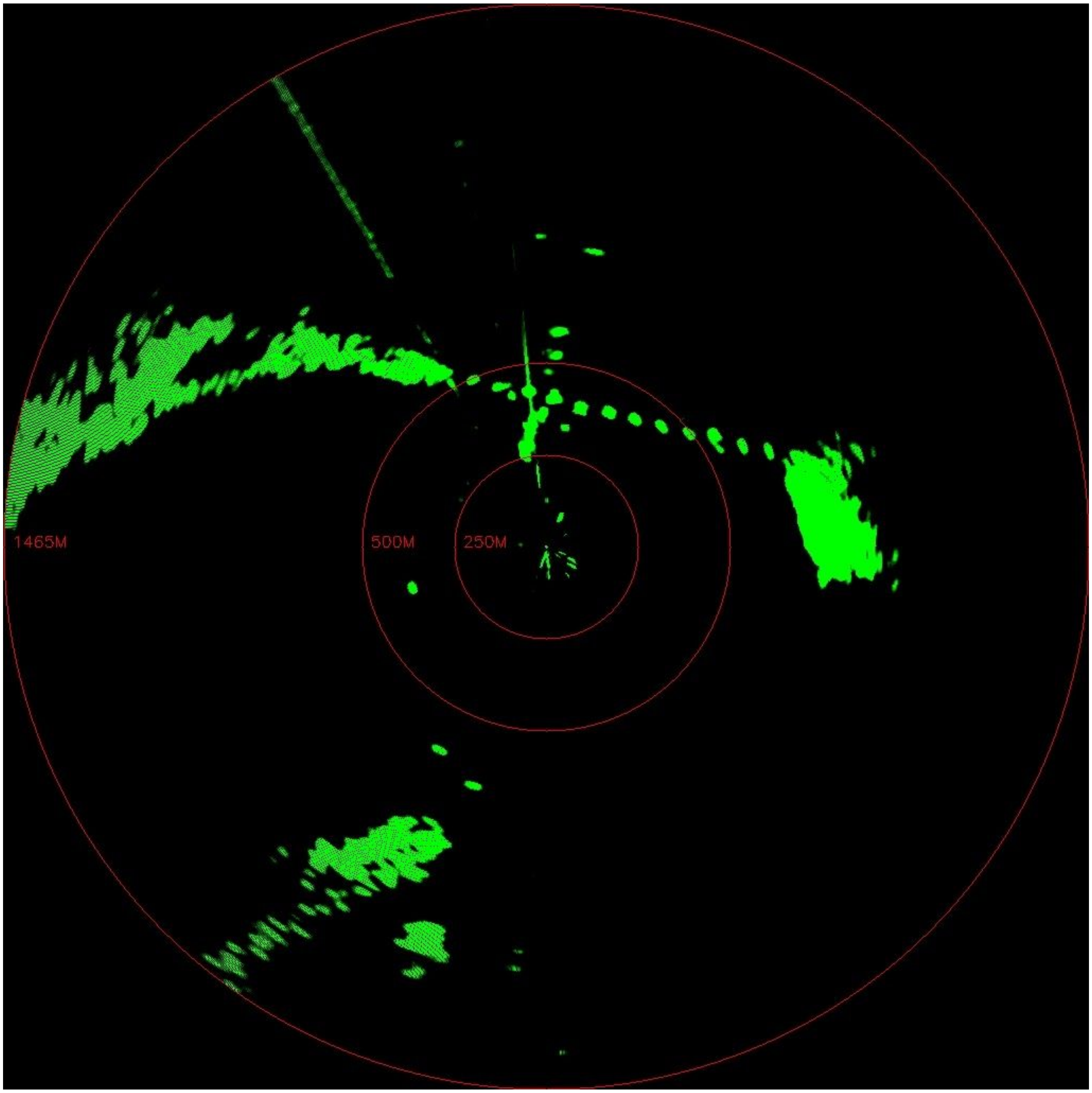


Images from Philos- (radar) Moored boats



Images from Philos- (LIDAR) Moored boats

Images from Philos (raining)- (Video) Ferry passes bridge abutments, Philos at speed



Images from Philos (raining)- (Radar) Ferry passes bridge abutments, Philos at speed