RoboWhaler: A Robotic Vessel for Marine Autonomy and Dataset Collection

Michael DeFilippo and Michael Sacarny
Dept. of Mechanical Engineering
Massachusetts Institute of Technology
Cambridge, Massachusetts, USA
mikedef@mit.edu, msacarny@mit.edu

Paul Robinette
Dept. of Electrical and Computer Engineering
University of Massachusetts at Lowell
Lowell, Massachusetts, USA
paul_robinette@uml.edu

***Abstract*—In this paper, we present a manned/unmanned vehicle that is used for autonomous surface vessel algorithmic development. This mobile marine autonomy laboratory is built upon a modified Boston Whaler and can be operated in both manual and fully autonomous modes. The primary driver in development was the creation of a marine autonomy sensor suite capable of high resolution sensing in all weather conditions. The vessel now serves as a mobile marine autonomy laboratory and is used to collect on-water datasets as well as testing behaviors and algorithms for autonomous operations. The vessel is equipped with a SBG Ellipse2-D RTK INS, a Velodyne VLP-16 lidar for high resolution 360° sensor coverage, a Simrad 4G broadband radar for long-range 360° sensor coverage, forward facing FLIR ADK infrared cameras, and forward facing FLIR Blackfly 1.3 MP cameras. All data is available in ROSBag format and can be manipulated and visualized using the open source Robot Operating System (ROS). We also make available tools to extract timestamped data presented in standard human readable formats. Datasets have been collected with the current marine autonomy sensor suite from fall 2019 to summer of 2021. These datasets have been curated for consumption and are available on the MIT Sea Grant website https://seagrant.mit.edu/auvlab-datasets-marine-perception-1.**

***Index Terms*—Autonomous surface vessel, marine robotics, computer vision, lidar, radar, infrared, sensor fusion, moos-ivp, ros, aquaculture**

1. Introduction

In this paper, we review the design of a mobile marine autonomy laboratory used to collect coordinated multi-modal marine datasets. The mobile marine autonomy laboratory was developed and refined from 2018-2020 by the engineers in the Autonomous Underwater Vehicles Laboratory at MIT Sea Grant in conjunction with engineers from Mercury Marine, a subsidiary of Brunswick Corporation. During this time numerous sensors were integrated and tested to understand the efficacy of a diverse sensor suite in the ocean environment. We decided on the elements of the marine sensor suite based on criteria including range, field of view, resolution, height information, affordability, light sensitivity, weather, and marine hardiness. The final marine sensor suite represents an affordable and well marinized collection to be used for marine dataset collection, as well as autonomous surface vessel research and development. The research vessel R/V Philos not only is used to collect marine datasets but is also capable of autonomous waypoint navigation and obstacle avoidance, becoming "RoboWhaler".

Figure 1. R/V Philos (RoboWhaler) is a mobile marine autonomy laboratory that is capable of collecting coordinated marine perception datasets. This marine vehicle operates in two modes manned and unmanned. While in the unmanned mode, the vessel becomes 'RoboWhaler' and is used for advanced autonomy and sensor fusion algorithm development

The mobile marine autonomy laboratory, R/V Philos (Fig. 1), is a 7 meter, 500 HP Boston Whaler. R/V Philos has the electrical infrastructure on-board to accommodate multiple researchers with 120 volt power outlets and Ethernet ports to connect to the vessel's network. On-board HDMI connects to a daylight monitor with keyboard and mouse, making computing possible while underway in bright sunlight. The vessel is outfitted with multiple sensors include three forward facing cameras with a combined 140° horizontal field of view (FOV), two infrared cameras with a combined 145° HFOV, one lidar and one marine radar, both with 360° HFOV, and an XSENS 670 INS/GNSS. Each sensor is hardwired into an on-board desktop computer with an Intel i7 CPU, GTX 1060 Nvidia GPU, 32 GB of RAM, and 2 TB of disk space. Drivers for each sensor have been developed in the lab or adopted from open sources using the Robot Operating System (ROS). The system has been proven out over the past few years, collecting multiple TB of marine datasets.

Building on top of the systems developed in R/V Philos for the mobile marine autonomy laboratory, we worked with engineers at Mercury Marine to interface with the vessel's Brunswick Vehicle control system. The MOOS-CAN interface enables us to monitor the Whaler status and command vessel heading and speed, all done through custom applications developed on top of the Mission Oriented Operating Suite Interval Programming (MOOS-IvP) autonomy software. MOOS-IvP is a set of open source C++ modules for providing autonomy on robotic platforms, particularly suited for the marine domain. Through this interface and associated sensor suite we are now able to develop and test advanced marine perception autonomy algorithms [2].

R/V Philos, aka RoboWhaler, will continue to be used for marine autonomy research as well as collection of coordinated datasets over the next few years at MIT. Each year the vessel is upgraded with additional sensors and autonomous capabilities, further enabling the advanced research capabilities for our partners and researchers on campus. The datasets collected are available publicly for all researchers to use through the MIT Sea Grant website (https://seagrant.mit.edu/auvlab-datasets-marine-perception-1/). This paper will further detail the design of our mobile marine autonomy laboratory, an overview of the marine datasets collected, and basic marine autonomy test missions with RoboWhaler.

2. System Description

We used a modified 7 m (25 ft) Boston Whaler as our data collection platform. The vehicle was integrated with the following perception and navigation sensors to form a mobile marine autonomy laboratory. It's purpose is to collect coordinated datasets to assist with the design of robust algorithms for autonomous surface vessels (ASVs), as well as serving as a test platform for these algorithms when the vessel is in its fully autonomous mode as the RoboWhaler.

2.1. Vessel Infrastructure

We designed our marine mobile research laboratory to have the infrastructure to support multiple sensors, computers, and researchers. With this in mind we installed a power inverter and DC battery bank to support the below sensors and associated equipment, such as a daylight viewable monitor (Fig. 2) as well as Ethernet and power ports in the fore and aft of the vessel's hard top. The marine radar was installed in a traditional way on top of a raised mount to help avoid blind spots around the vessel. The lidar and cameras were installed on custom railings above and below the front bimini. A custom USB/Ethernet and power hub was designed and installed in the upper glove box. All associated wiring for the sensors as well as the HDMI for video run through this glove box and into the cabin where the main computer is mounted and the power inverter is located. All sensors except the marine radar were designed to be removed at the end of the day with finger tightened bolts. The removable sensors are rigidly mounted to a rail to ensure sensors are put back in the same place for ease of calibration.

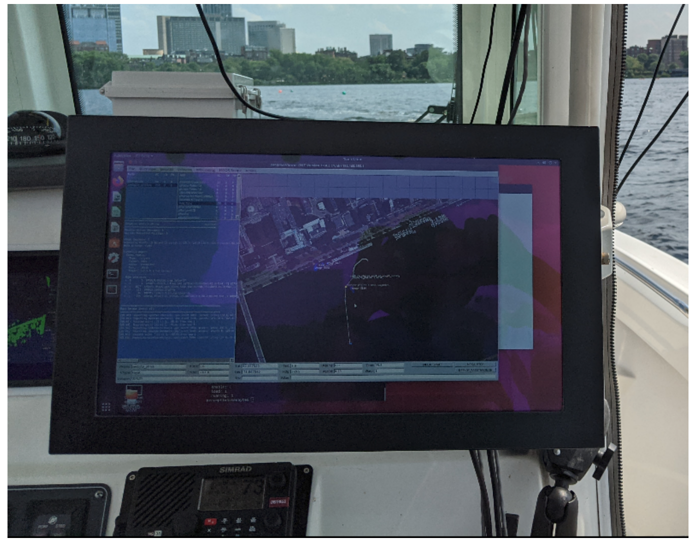

Figure 2. R/V Philos daylight viewable monitor

2.2. Sensors: Perception

Data collection from 2020-2021 was accomplished with the following sensors mounted on the vessel, see Fig. 4 for a reference of how the sensor suite is mounted. Our marine sensor suite provides a forward facing camera visibility of approximately 145° horizontal field of view (HFOV) as well as an all-around ranging capability 360° HFOV (Fig. 3). The 2019 perception sensor setup was slightly different and subsequently datasets from 2019 will reflect a slightly modified setup with the information provided within the dataset.

- One Velodyne VLP-16 lidar sensor has a range of 100 m and generates 600,000 points/second, with a 360° horizontal field of view (FOV) and a 30° vertical FOV.
- One Simrad Broadband 4G radar, consisting of two radar units with 360° FOV, min/max range 50/66,000 m, beam width 5.2°
- Two FLIR ADK infrared (IR) camera, 640 x 512 resolution, each with 75° horizontal FOV for a total of 145° horizontal FOV, nominal frame rate of 30 frames per second (fps), timestamped
- Three FLIR Pointgrey cameras, 1280 x 1024 resolution, each with 48° horizontal FOV for a total of 145° horizontal FOV, 36° vertical FOV, nominal frame rate of 12 fps, compressed and timestamped

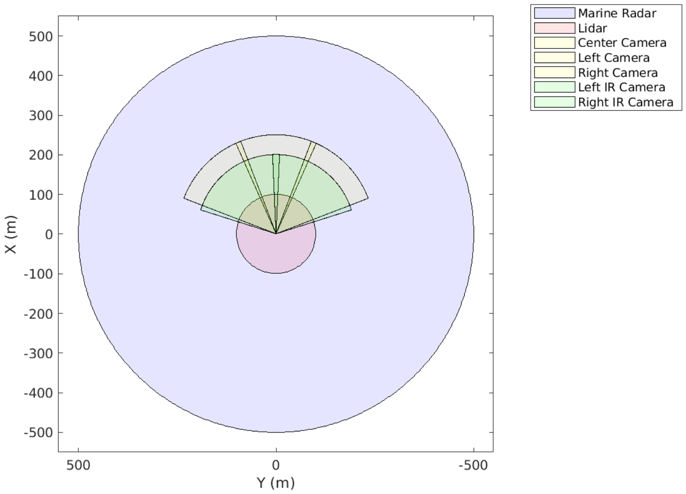

Figure 3. R/V Philos Sensor approximate coverage map. Each sensor as part of the 2020/2021 sensor suite is represented. The radar was capped at 500 meters as this is our predominant setting in use for inner harbor and river navigation.

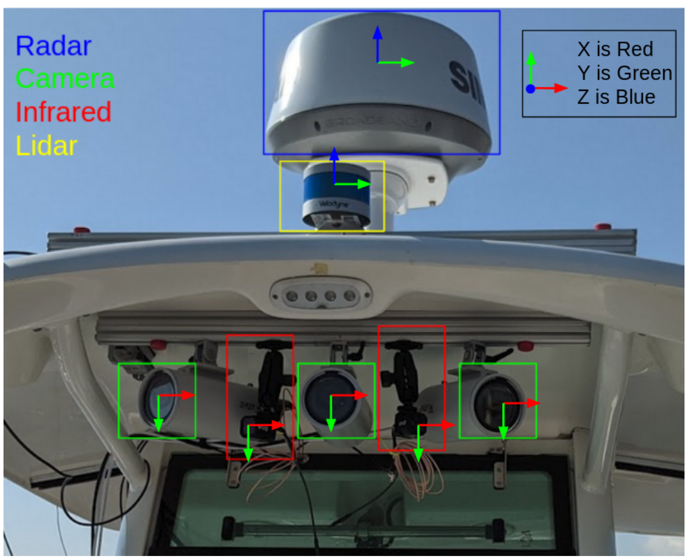

Figure 4. R/V Philos Sensor Coordinate Systems

2.3. Sensors: Navigation

The navigation solution is provided by both the R/V Philos SBG Systems Ellipse2-D Dual Antenna RTK INS and a secondary MTi 670 GNSS/INS. They both provide roll, pitch, yaw, heading, heave, velocity, and position. We installed the second INS (MTi 670) as raw data from the SBG INS is not exposed and we wanted the raw output of a high performance INS for our datasets and navigation algorithms.

2.4. Coordinate Frames

The vehicle body frame is linked to the body of the vehicle. The center of the body frame is the center of the

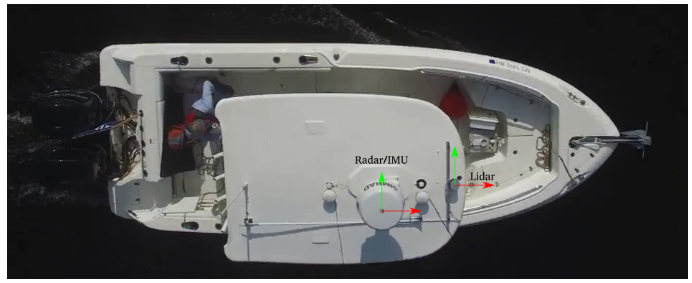

Figure 5. R/V Philos navigation coordinates, red: x-axis, green: y-axis, z-axis positive upwards

TABLE 1. SENSOR COORDINATE FRAMES RELATIVE TO VEHICLE BODY FRAME (INS)

Transform	Sensor	Value (meters and degrees)
X_R	Radar	[0 0 1.78 0 0 0]
X_L	Lidar	[0.79 0 1.17 0 0 0]
X_{CC}	Center Camera	[0.94 0 0.86 0 10 0]
X_{LC}	Left Camera	[0.94 -0.27 0.86 0 10 45]
X_{RC}	Right Camera	[0.94 0.25 0.86 0 10 -45]
X_{RI}	Left Infrared	[0.94 -0.14 0.86 0 11 35]
X_{LI}	Right Infrared	[0.94 0.14 0.86 0 11 -35]

INS, which is located approximately at the vehicle's center of gravity at the waterline below the radar. The coordinate frames are shown in Fig. 4/5. The transformations are given in Table 1 given as [x y z ϕ θ ψ], with xyz in meters and roll angle (ϕ), pitch angle (θ), and yaw angle (ψ) in degrees.

3. The Data Sets

Public datasets are very useful to the research community to evaluate the performance of algorithms with relation to single sensors or a suite of sensors. Recently, we have seen an influx of datasets [13]–[15] from autonomous car companies to help aid in developing robust algorithms for autonomous cars and trucks. What has been lacking from our point of view is a meaningful dataset from the marine domain. Here we set out to release a marine perception dataset that can be useful to both industry and researchers alike. The datasets consist of multiple files of raw unprocessed data from all of the navigation and perception sensors. The data can have occasional dropouts, glare from the sun, rain/snow on camera lenses, or sensor jitter from rough seas. The dataset provides multiple vehicle path trajectories, some with loop closures, in diverse weather conditions, and at multiple vessel speeds.

4. Data Access Tools

The datasets are provided as ROSBag format that can be manipulated and visualized using the open source middleware ROS. There are many tools for reading and writing of ROSBag files using C++/Python [10] that are provided as well. Additionally we have provided data parsing tools to assist in creating CSV files of data arrays, as well as parsing tools to create time stamped images and videos from the ROSBag stored camera data. Below are a selection of some of the data processing tools that we have developed and are available to be used with our datasets.

- **extract_images2:** Script to extract images and create videos from a ROSBag
- **rosbag_to_pcap:** Script to convert laser point cloud data to pcap file
- **rosbag_to_csv:** Script to extract parameters from ROSBag to a CSV file
- **alog_to_csv:** Script to extract parameters from MOOS-IvP alog to CSV file
- **alog_to_bag:** Script to extract parameters from MOOS-IvP alog and combine with ROSBag file

5. RoboWhaler

Our Boston Whaler comes equipped with some advanced features right out of the box, including a joystick control and digital throttles. We worked with engineers at Mercury Marine to develop an interface into the vessel's control system to create a fully autonomous surface vessel, which we call RoboWhaler. The interface between MOOS and the Brunswick Vessel (BV) is through a custom designed MOOS-CAN interface using an embedded computer. This allows us to monitor the status of the vessel, such as motor RPM, as well as to have control over the vehicle's heading and speed, with the combination of the perception and navigation sensors mentioned above has created a very capable autonomous vehicle that has been used as a test bed for many research projects over the last two years.

RoboWhaler's autonomy and perception systems are built upon MOOS-IvP and ROS respectively. We use MOOS-IvP to take advantage of its extensive set of marine behaviours and ROS for its rich vehicle ecosystme and deep sensor driver support. MOOS-IvP and ROS are bridged over custom applications that pass NMEA style messages back and forth. All of the mission data is stored in ROSBag format and collected on the vessel's main autonomy computer.

5.1. MOOS-IvP

The surface vessel's navigation system is based on the MOOS-IvP [1] autonomy software suite. MOOS (Mission Oriented Operating Suite) is a robotic middleware that provides a platform-independent, efficient, and robust architecture for real-time applications components of a robotic system to execute and communicate. MOOS provides a middleware capability based on the publish-subscribe architecture and protocol. MOOS is an open source project maintained by Oxford [3]. The MOOS database (MOOSDB) is central to the architecture (Fig. 6), and provides a mechanism for communication between applications.

MOOS-IvP (Interval Programming) is a MOOS application designed to provide autonomy on robotic platforms and is particularly well-suited to marine vehicles. IvP behaviors determine how the vehicle responds to its environment in

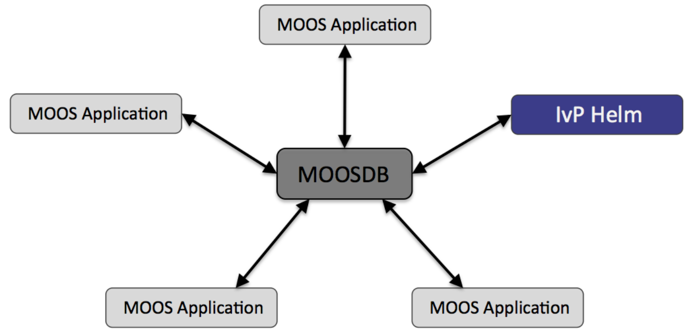

Figure 6. A MOOS community is a collection of MOOS applications typically running on a single machine each with a separate process ID. Each process communicates through a single MOOS database process (the MOOSDB) in a publish-subscribe manner. Each process may be executing its inner-loop at a frequency independent from one another and set by the user. Processes may be all run on the same computer or distributed across a network.

pursuit of some defined goal (Fig. 7). Examples include Waypoint behavior (traverse a set of given waypoints) [4] and Avoid Obstacle Behavior (avoid an obstacle based on a tracked object from pObstacleMgr [5]).

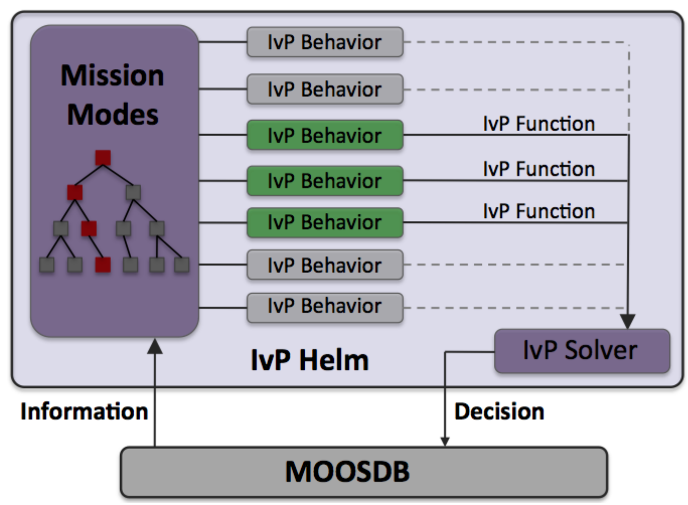

Figure 7. IvP Helm Architecture. 1: Mail is read in the MOOS OnNewMail() function and applied to the local buffer. 2: The helm mode and set of running behaviors are determined. 3: Behaviors execute - posting MOOS variables and an IvP function. 4: Competing behaviors are resolved with the IvP solver. 5: The helm decision and any behavior postings are published to the MOOSDB.

5.1.1. MOOS-IvP RoboWhaler Interface. The separation between vehicle control (the "front-seat") and vehicle autonomy (the "backseat") [8] is a key component to operation of the vehicle. MOOS, the backseat, commands vessel heading and speed to the MOOS-CAN interface and on to the vessel control system, the front-seat. The vessel control system issues commands to the engines and passes back navigation and vehicle information to the MOOS-CAN interface and the backseat (Fig. 8).

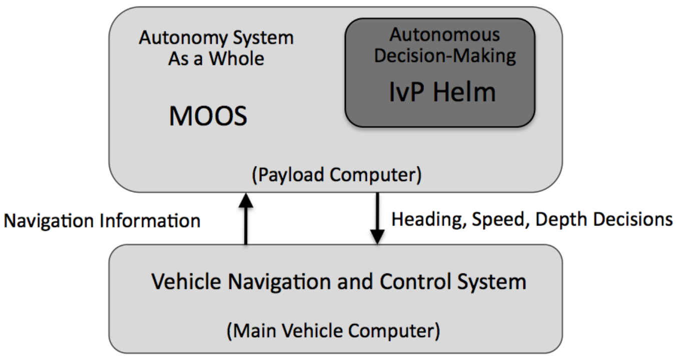

Figure 8. MOOS-IvP Backseat Driver Architecture

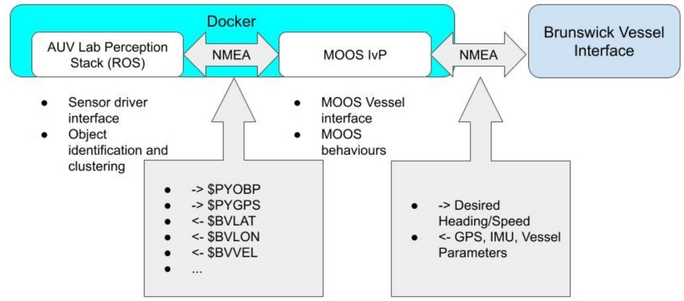

Figure 9. Multiple middlewares in operation to sense, plan, and act on-board an RoboWhaler. ROS and MOOS-IvP running in separate Docker Containers that communicate over TCP/IP to one another as well as to the vessel control computer

The MOOS-CAN interface provides a means for MOOS applications to monitor and control the state of the vessel. For safety and security, access to the vessel CAN bus is severely restricted, limited to a small set of custom CAN messages. The MOOS-CAN interface, currently implemented on a RaspberryPi running Python, makes the data in vessel status CAN messages available to a TCP socket monitored by MOOS. Likewise, MOOS vessel commands, such as desired speed and heading, are posted to a socket for translation into CAN messages.

5.2. ASV Perception Stack

The ASV Perception Stack is a Robot Operating System (ROS) based suite of sensor drivers and sensor fusion algorithms designed around integration of a marine perception sensor suite aboard RoboWhaler. The marine perception sensor suite is interchangeable and can range from a single camera to a full suite of various combinations of perception sensors, such as radar, lidar, camera, and IR. Each sensor is configurable through launch script environmental parameters via Docker. The sensor fusion algorithms are designed to cluster objects from the radar and lidar and report via NMEA message strings to the MOOS obstacle manager. The obstacles are then used to provide the optimal heading and speed to the vessel to safely navigate through an obstacle field. Preliminary fusion algorithms are being developed with our partners at the University of Massachusetts Lowell [5], that utilize modern machine learning models and retrained models to classify items detected in the camera images and associate the classifications to the radar detections for robust 360° tracking and identification.

5.3. Middleware and Platform Agnostic Software Paradigm

A common problem with most software designed for a particular type of robot in the research community is that it is typically not very modular and reusable. With that in mind we have been refactoring our code around a middleware and platform agnostic software paradigm taking advantage of Docker containers [12]. Using common images created from our software stacks we are able to transfer the working code on one of our ASV's and replicate that on a similar ASV with only a few changes in the parameters to launch these containers (Fig. 9). We have worked with partners to transfer our working code to their surface vessel with a similar, but not the same set of marine perception sensors. Our partners implemented their own autonomy on-board a surface vessel and used our docker containers to receive obstacle alert messages based on our sensor algorithms. Another useful part of using containers is that they provide a simple solution to pool together resources to test on a common robot. We are able to use this setup to have teams in different locations to develop algorithms on saved data and easily transfer those working algorithms back to a fully functional autonomous vessel, such as RoboWhaler.

5.4. Results

5.4.1. Marine Autonomy for Aquaculture. Most recently we have been developing basic autonomy algorithms for the future of aquaculture. Autonomous surface vehicles are posed to play an important role in monitoring aquaculture farms, both near-shore and off-shore. We are currently developing and testing some crucial capabilities like navigation and obstacle/contact avoidance. These capabilities are vital to using an autonomous surface vessel within an aquaculture farm and for general transiting.

Preliminary tests of navigating an ASV through an obstacle field are promising. We envision an aquaculture farm with many buoys and floating structures as an operational region. Obstacles are detected and passed to the MOOS-IvP obstacle manager [6]. Obstacles within the path of the vessel are then expanded upon based on a user defined buffer region. The obstacles are then safely avoided based on the new path around the obstacles within that region. If any new obstacles are detected or the current obstacles shift the obstacle manager is able to correct for these situations and provide a safe path for each update. Testing of similar situations are performed on the Charles River (Fig. 10) and has shown that we can safely provide a path around detected obstacles given that they are detected by our sensor systems.

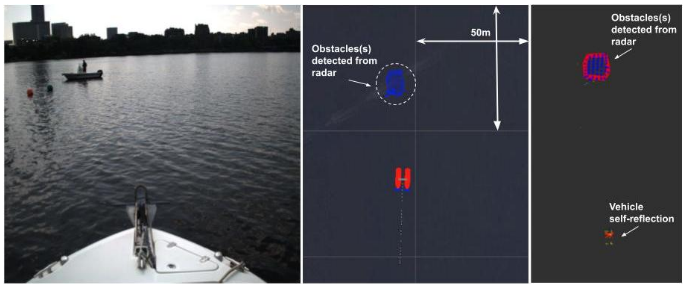

Figure 10. Left: Center camera showing buoys and vessel in front of Philos. Center: MOOS-IvP birds-eye view of Philos and detected obstacles. Right: RVIZ birds-eye view of clustered obstacles as a blue circle with red perimeter.

Other capabilities that are being developed for aquaculture involve integration of oceanographic sensors onto platforms that monitor aquaculture farm health. We developed an application to interface with the Eureka Manta water quality probe (pParseManta) for use at an aquaculture farm in Chesapeake Bay. The purpose of pParseManta is to connect to the sensor and collect data from each sensor within the probe and save that data onto the MOOS Database for offline processing. Data can also be used in realtime to direct the robot to new areas based on predefined tasks.

5.4.2. Marine Perception Datasets. Curated datasets are available for download via our website [11] via the data access tab. Datasets are separated into two categories: low speed and high speed. Within each category we have provided interesting sections of data labeled with a date and which vessel the data was collected by. Each dataset has a few tags next to them to help further explain why something may be interesting within that dataset such as solar glare, sailboats, or docking. There is also a short 20 second section provided with each dataset to view before downloading the data. Figure 11 is a screenshot of the metadata provided. This screenshot highlights a limited number of sensors onboard R/V Philos. We tried to capture many of the interesting scenarios that we see on a daily basis within the Charles River or out into Boston Harbor.

The dataset as downloaded consists of image outputs from each camera as well as the radar output. All images are time stamped files within a labeled directory. Videos are created from each of the camera sensors and stored in the videos directory. Radar is outputted as time stamped point cloud data (pcd) files within the radar pcd directory. Lidar is also provided as a pcap file for easy visualization. Lastly all of the raw data from each sensor on the vehicle at the time is saved in a bag file. The data is all time synced within the bag file and can be used in the ROS eco system.

5.5. Conclusions and Ongoing Efforts

We are working to expand upon the datasets that we have already collected. While the datasets collected are useful we recognize a lack of ground truth within the data. To solve

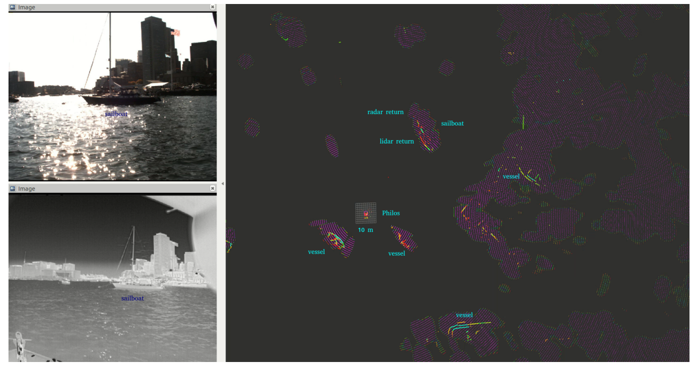

Figure 11. R/V Philos collecting data through a mooring field in Boston Harbor. Upper left image corresponds to the right hand camera and the lower left image corresponds to the right hand IR and the right image corresponds to the radar (purple) and lidar (multi-colored) sensor outputs. Labels are based on visual and IR camera data.

this we have developed an embedded system that will record GPS and IMU data on-board a target vessel. Having these systems on multiple vessels we can create curated datasets to provide ground truth data to further assist with sensor fusion and autonomy algorithmic development. Targeted datasets are also on the horizon. We encourage our partners and anyone that is interested to request are particular type of dataset and we will work to the best of our ability to create it. These enhansed datasets will be available via AUV Lab datasets website [11] by the end of 2021.

Standard interfaces enable open source and proprietary software to work together, leading to rapid development in both the public and private sectors. For ASVs to enter the real-world and provide much-needed services in shipping, patrol, and environmental monitoring/cleanup, we need an environment that allows systems from different sources to work together with ease. Our testbed is a prototype of how that can work on a fully autonomous vessel.

5.6. Acknowledgments

The authors would like to thank the Brunswick Corporation and our partnership with the Emerging Technologies and Innovation Processes group at Mercury Marine (a division of Brunswick Corporation). The loan of the Boston Whaler (R/V Philos), prior funding, and engineering expertise made it possible to outfit the vessel with sensors and interface into the vehicle control system,which was the beginnings of RoboWhaler. We would also like to thank the Charles River Yacht Club for donating a slip for R/V Philos.

This publication is the result of research sponsored by The MIT Sea Grant College Program under project number 2020-R/RD-43. Partial funding for this project was provided by the Brunswick Corporation. The Views, opinions and/or findings expressed are those of the authors and should not be interpreted as representing the official views or policies of our sponsors.

References

[1] Benjamin, M., Schmidt, H., Newman, P., Leonard J.J.: Nested Autonomy for Unmanned Marine Vehicles with MOOS-IvP. Journal of Field Robotics. 27(6), 834-875 (2010).

[2] Michael R. Benjamin, Henrik Schmidt, and John J. Leonard. The moos-ivp open source marine autonomy project. http://www.moos-ivp.org(2021)

[3] Waypoint Behavior: http://oceanai.mit.edu/ivpman/bhvs/Waypoint.

[4] Paul Robinette, Michael Sacarny, Michael DeFilippo, Michael Novitzky, and Michael Benjamin. "Sensor Evaluation for Autonomous Surface Vehicles in Inland Waterways". In: OCEANS 2019 Marseille. IEEE. 2019, pp. 1–8.

[5] Thomas Clunie, Michael DeFilippo, Michael Sacarny, and Paul Robinette. "Development of a perception system for an autonomous surface vehicle using monocular camera, lidar, and marine radar". In: ICRA 2021 Xi'an. IEEE. 2021.

[6] Michael R Benjamin, Michael DeFilippo, Paul Robinette, and Michael Novitzky. "Obstacle Avoidance Using Multi-Objective Optimization and a Dynamic Obstacle Manager". In: IEEE Journal of Oceanic Engineering 44.2 (2019), pp.331–342.

[7] MIT-Battelle Marine Autonomy Bay: http://oceanai.mit.edu/pavlab.

[8] IvP Helm: https://oceanai.mit.edu/ivpman/pmwiki/pmwiki.php

[9] ROSBag: https://wiki.ros.org/rosbag

[10] ROSBag tools: https://wiki.ros.org/rosbag/Code

[11] AUV Laboratory Marine Datasets: https://seagrant.mit.edu/auvlab-datasets-marine-perception-1/.

[12] Merkel, D., 2014. Docker: lightweight linux containers for consistent development and deployment. Linux journal, 2014(239), p.2

[13] Geiger, A., Lenz, P., Stiller, C., and Urtasun, R. "Vision meets Robotics: The KITTI Dataset". In: IJRR. 2013,. pp. 1-6.

[14] Caesar, H., Bankiti, V., and Beijbon, O. "nuScenes: A multimodal dataset for autonomous driving". In: IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). 2020. pp. 1-15.

[15] Siddharth Agarwal, Ankit Vora, Gaurav Pandey, Wayne Williams, Helen Kourous and James McBride, "Ford Multi-AV Seasonal Dataset", in arXiv preprint arXiv:2003.07969,2020