2021 IEEE International Conference on Robotics and Automation (ICRA 2021)
May 31 - June 4, 2021, Xi'an, China

# Development of a Perception System for an Autonomous Surface Vehicle using Monocular Camera, LIDAR, and Marine RADAR

Thomas Clunie*, Michael DeFilippo†, Michael Sacarny† and Paul Robinette*
*University of Massachusetts at Lowell, Lowell, MA, USA
†Autonomous Underwater Vehicles Laboratory, MIT Sea Grant, Cambridge, MA, USA

***Abstract*—This paper describes a set of software modules and algorithms for maritime object detection and tracking. The approach described here is designed to work in conjunction with various sensors from a maritime surface vessel (e.g. marine RADAR, LIDAR, camera). The described system identifies obstacles from the input sensors, estimates their state, and fuses the obstacle data into a consolidated report. The system is verified using experiments conducted on a live system and successfully demonstrates the ability to detect and track obstacles up to 450m away while operating at 7 fps. The software is open source and available at https://github.com/uml-marine-robotics/asv_perception.**

***Index Terms*—Autonomous systems, marine robotics, intelligent robots, mobile agents unmanned autonomous vehicles, autonomous surface vehicles, marine RADAR, LIDAR, segmentation, classification, object detection, calibration, sensor fusion.**

## I. INTRODUCTION

Despite the significant research in the autonomous ground vehicle domain, relatively little effort has been expended to research the solutions to the unique problems in the autonomous surface vehicle (ASV) domain. These unique challenges include wind, waves, current, glint, sea fog, etc. Even further, there has been relatively little research in software solutions which combine data from multiple heterogeneous sensors to form a comprehensive view of the surrounding environment while addressing these challenges.

The worldwide ASV market is projected to reach $1.2B by 2027 [1], with the bulk of that funding coming from the defense segment. The US Navy alone has requested $579.9M in FY2021 for the research and development of large unmanned vehicles and their enabling technologies [32], with various levels of human involvement, ranging from human operators in the loop to fully autonomous systems.

An awareness of the environment; specifically, the perception of both static and dynamic obstacles on the water surface along with their classification, estimated speed and heading, is a prerequisite for effective autonomous and semi-autonomous operations, including International Regulations for Preventing Collisions at Sea (COLREGs) [33] compliance, navigation, and other specialized tasks. This perception task is most critical in dense environments such as inland navigation, where the water surface may be populated by non-cooperative agents such as human swimmers, boats, other ASVs, etc. as well as fixed structures such as docks, moorings, etc. Additionally, the detection of these obstacles must be completed early enough to allow adequate time for the ASV to perform avoidance maneuvers, especially at near-field ranges (<275 meters [25]).

Furthermore, the capabilities and cost of computer hardware, software, and sensors continues to improve, offering new solutions to problems that were previously difficult to solve. In the ASV domain, the integration of LIDAR as well as advancements in deep learning make reliable obstacle detection a more tractable problem.

In this paper, we describe an efficient obstacle detection, localization, and tracking system for ASVs. We start with the sensor payload recommended by Robinette et al. [40] and DeFilippo et al. [8] (marine RADAR, LIDAR, visible-light camera), and integrate state of the art research in object detection [6] and on-water image segmentation [7]. This open source, modular system integrates a suite of heterogeneous sensors and runs on the Robot Operating System (ROS) platform at 7 fps. Aside from the system itself, unique contributions include a method of qualitative monocular camera and point cloud calibration, along with a fast and modular 3D object tracking and fusion algorithm.

## II. RELATED WORK

### A. Maritime Sensor Evaluation

Prasad et al. [37] includes a survey of different sensor types for maritime situational awareness, including RADAR, visible light camera, infrared, and sonar. Robinette et al. [40] analyzes the utility and practical limitations of RADAR, LIDAR, stereo camera, and visible light monocular cameras for inland maritime navigation.

Practically speaking, marine RADAR is currently the only sensor which can currently provide 360-degree situational awareness at ranges needed for effective navigation in all weather conditions. However, it has a slow refresh rate, lacks contact height information, lacks sufficient resolution at close range, and does not provide sufficient obstacle identification characteristics.

LIDAR has shown [3] [28] [40] [8] to be useful for short-range obstacle detection in a marine environment, overcoming the short-range limitations of RADAR but fails to detect objects at medium to long range (typically limited to <100m). Lastly, obstacle detection using visible light cameras in a marine environment has been shown to be effective, and can be used to provide supplementary obstacle information for effective COLREGs compliance, despite the limited range and field of view.

Utilization of thermal imagery is promising, as it is more resistant than visible light cameras to solar glare and lens flare, and operates during the day or night. However, it still may suffer from fog [40]. Despite these benefits, there has been limited research into the utilization of this sensor in the marine environment. Recently, Helgesen et al. [16] found that combining infrared with RADAR and LIDAR improved small obstacle tracking over RADAR and LIDAR alone in a maritime environment, in both daytime and nighttime conditions. Schöller et al. [43] demonstrated some success detecting ships using Convolutional Neural Networks and data from a Long Wavelength Infrared sensor, with the detection range varying from 65m for small vessels to 850m for large vessels. Additional research is needed on the ability to extract and fuse useful obstacle information from thermal data while the sensor is mounted on an ASV.

### B. Multi-Sensor Systems

In 2017, Schiaretti et al. [42] provided a survey of existing ASV prototypes. Of the 60 systems listed, 11 of the systems list obstacle avoidance capabilities. Only 3 of those systems combined

multiple perception sensor modalities into a cohesive system and have literature available for review. These 3 systems are discussed below.

Larson et al. [25], [26] used a two-tier obstacle avoidance system, with the near-field/reactive avoidance system using RADAR, stereo vision, nautical charts, monocular vision, and millimeter-wave radar. The far-field system utilized AIS, nautical charts, and ARPA contacts. For depth estimation in a monocular camera image, the horizon is estimated using a Hough line transform and then the distance to obstacle was calculated via trigonometry. However, the researchers faced obstacle detection and horizon estimation challenges due to the dynamic water surface and other environmental factors. In calm waters, RADAR obstacle tracking was successful up to 200m.

Gray et al. [13] combined camera and LIDAR data in ROS-based PropaGator I, while Frank et al. [11] added infrared, and passive SONAR in PropaGator2. LIDAR data was projected onto the 2D camera image and OpenCV image algorithms were used for obstacle identification.

Since Schiaretti's work in 2017, we found 3 more multi-sensor perception systems available for review. Norbye [31] combined LIDAR and camera data. The ROS-based system used a custom-trained YOLOv3 [38] object detector and successfully tracked obstacles up to 60m away in real time. Sorbara et al. [44] utilized a camera, infrared, and a single-beam laser range finder on a rotating platform to achieve a 180 degree field of view. Sorial et al. [45] combined a camera with LIDAR and used YOLOv3 [38] for boat detection, and tracked the boats in the camera and LIDAR frames.

Each of the aforementioned systems fail to provide adequate perception data for an ASV. Larson's system was simply ahead of its time; performant deep learning frameworks for on-water detection were not yet conceived, and commercial LIDAR sensors were generally not available. The remaining systems have been shown to be effective at short range detection. We aim to build on these concepts to extend obstacle detection and tracking to distances beyond 100m, exploiting the long range capabilities of both the camera and marine RADAR, as well as potentially improving short-range obstacle detection during adverse environmental conditions.

### C. Single-Sensor Systems

Recently, there have been several other systems which utilize a single perception sensor and/or include sensors not considered in this work. However, the authors' approaches towards solving their respective problems may be utilized to solve subproblems of the current task at hand.

Kufoalor et al. [23] utilized Automatic Identification System (AIS) for obstacle tracking. Benjamin et al. [3] utilized LIDAR to generate pointclouds, which were then clustered to produce obstacles. Muhovič et al. [29] used LIDAR to generate obstacles from a pointcloud, with depth fingerprint for tracking, and later [28] employed a stereo camera to generate a point cloud, filtered out the water surface, and then used a histogram-like depth appearance model for obstacle identification and tracking. Zhuang et al. [52] used radar-generated images and image processing algorithms to detect and track obstacles. Kuang et al. [22] applied clustering methods to generate obstacles using UHF radar. Paccaud et al. [34] used a low cost, consumer-grade camera for obstacle detection on a lake. Manderson et al. [27] used deep neural networks for collision avoidance and object detection on an underwater vehicle. Fiorini [10] et al. developed an efficient machine learning approach towards person and vessel detection with a UAV-based moving and zooming camera. Ueland et al. [47] used LIDAR for SLAM on a surface vessel for autonomous marine exploration. Heidarsson et al. [15] used a profiling SONAR for obstacle detection and avoidance.

Many of these systems are designed for smaller, specialized, low power vessels thus do not completely solve the current problem. However, these researchers employed a variety of methods in processing sensor data and provided useful insights into current methods of obstacle detection.

### D. Maritime Object & Horizon Detection, Segmentation

There have been many recent developments in the area of on-water object detection, segmentation and horizon detection from visible light imagery. Prasad et al. [37] presents a survey of maritime object detection and tracking approaches from video, and Muhovič et al. [28] provides a thorough review of obstacle detection on the water surface. Additionally, there has been recent success by Bovcon and Kristan [7] and Steccanella et al. [46] in the use of neural networks for segmentation of on-water images to distinguish between water, sky, and obstacle pixel data. Petković [36] et al. provides an overview of horizon detection methods in maritime video surveillance.

Although these approaches do not directly address the current problem in its totality, they are useful when considering approaches to processing the data from visible light cameras and can be utilized within the context of a larger perception system. The ability to differentiate between obstacle and non-obstacle, even if the type of obstacle is unknown, is an important capability of a perception system. Additionally, the ability to identify the horizon helps decrease the size of the search space while reducing the number of irrelevant detections.

## III. System Architecture

The primary goal of the perception system is to identify both static and dynamic obstacles (of known and unknown types) and estimate their position, size, and heading. Building on the work of previous researchers, we start by selecting the sensor platform which provides the necessary raw data which meet the obstacle detection needs of an ASV, and then select the software approaches which process the sensor data to generate candidate obstacles for consolidation into a cohesive view. We then report this information to an external navigation/obstacle avoidance system (such as [3]) and/or human operator.

From a sensor perspective, we utilize the recommendations of Robinette et al. [40] and DeFilippo et al. [8], choosing a marine RADAR, LIDAR, and 3 forward-facing monocular visible light cameras for our experiments. See Section IV-A for details on the selected sensor configuration utilized in this work.

From a software perspective, we utilize the ideas employed by Norbye [31] and Sorial et al. [45] to integrate state-of-the-art object detection algorithms along with LIDAR data to detect, identify, and localize obstacles using the visible light camera and LIDAR. Detection of unidentified obstacle types is provided by Bovcon and Kristan's WaSR segmentation network [7] . Lastly, the integration of RADAR data is inspired by the Larson et al. model [25] to achieve obstacle detection at the desired range. We then fuse the obstacle information together using a simple fusion system, as described below. Fig 1 provides an overview of the software architecture.

Conceptually, we have $N$ heterogeneous sensors, each associated with a sensor-specific module. This module is responsible for creating localized obstacle detection data for each obstacle it encounters

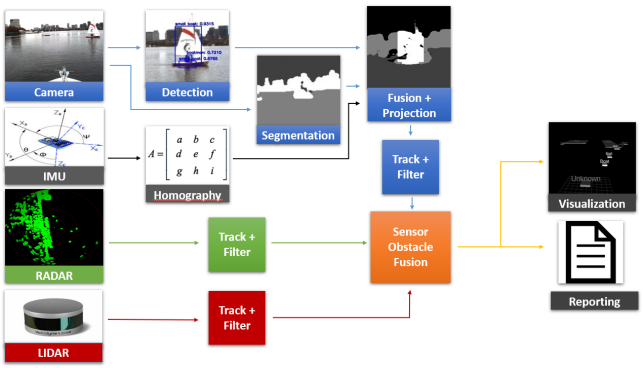


Fig. 1. Perception System Architecture

in every frame. These detections are then processed with a Kalman filter [21] to reduce noise, and are then fused with detections from the other $N$-1 modules to create a cohesive report of all obstacles. Additionally, the system is designed to be flexible in the quantity and type of sensor inputs, easily accommodating additional sensor inputs (e.g. extra cameras, radar, etc) and fusing their outputs into a singular, cohesive view. Below are the details of the sensor-specific modules which generate obstacle detection data.

### A. RADAR and LIDAR Obstacle Generation

Both of the RADAR and LIDAR sensors utilized on our test platform output their data as point clouds. Therefore, the problem can be reframed as one of identification of obstacles from point cloud data, while allowing some flexibility in the identification process to account for differences in sensor data. For example, a marine RADAR limited to 100m may provide a denser point cloud for the same obstacle as generated by the same marine RADAR when operating in excess of 1km. Additionally, marine RADAR only returns data in two dimensions as compared to LIDAR which provides data in all three dimensions.

For the efficient manipulation of point clouds, we utilize the Point Cloud Library (PCL) [41] to filter out land masses and generate obstacles from point clusters. Land mass filtering is accomplished by removing points which can be clustered into two-dimensional areas greater than some two-dimensional area $A$, and then candidate obstacles are generated by clustering points based on some Euclidean distance $D$. In our tests, the optimal values of $A$ and $D$ varied by sensor, as described above, and were discovered empirically. Moreover, the land mass filtering process was not completely successful in removing all land-based obstacles, resulting in many false positives when operating near shore. Ideally, our land mass filtering would utilize known map data for the removal of land mass obstacles to significantly reduce the number of false positives.

After the candidate obstacles are generated from the point clouds, their data is sent to the sensor fusion module (described below) for further processing.

### B. Monocular Camera Obstacle Generation

To generate candidate obstacles from a camera image, we must have a way to 1- identify both known and unknown obstacles in an image and 2- understand where those obstacles are located relative to our vessel. Furthermore, we wish to preserve unique information that is acquired from the camera image, such as object classification (e.g. boat, person) and color information (e.g. colored navigational buoy). This information can be used to supplement detection data from other sensors, as well as identify unique obstacles which cannot be identified by other sensor types alone.

Fig. 2. Refined visualization of a sailboat when combining the classification and segmentation images

In order to accomplish this, we combine an object detector using YOLOv4 [6] with the semantic segmentation network developed by Bovcon and Kristan [7]. After performing object detection and image segmentation independently, the two images are fused (Figure 2) to refine the predicted object detection bounding box, create candidate obstacle data, and then project the remaining unclassified obstacle pixels to a pseudo-LIDAR pointcloud for further processing.

*1) RADAR-camera calibration:* For the accurate estimation of 3D position of an obstacle identified in a monocular camera image, we require a way to convert 2D image pixel location ($u$, $v$) to 3D vessel-relative position ($x$, $y$, $z$). An accurate transformation to the vessel's frame is critical for the perception task, where the difference of a single vertical image pixel can represent over a dozen meters in world position as an obstacle approaches the horizon.

Achieving proper calibration is especially challenging in this context due to 1- the continual shift of camera pitch and roll due to waves and boat propulsion and 2- the coarse returns and a slow refresh rate ($\sim$1 Hz) of our marine RADAR relative to the visible light camera. It is also possible that camera positions change slightly during operation due to adverse environmental conditions (e.g. choppy water). Maintaining calibration over time also presents a challenge, as operators may remove the cameras from the boat between missions, resulting in the need to recalibrate between the two sensors before each mission in order to achieve proper alignment.

Existing literature [2], [49], [48] on the topic of RADAR-camera calibration generally involve the transformation using corresponding points between the camera and RADAR planes, and then using an ordinary least squares or RANSAC methods to find the projection matrix $P$ which defines a transformation between the two planes. Alternative approaches using horizon detection [35] have limited utility in inland waterways (such as our targeted operating environment), while other georeferencing approaches (e.g. [17]) assume known/fixed camera extrinsics. However, in contrast to the RADAR type and calibration environment used in existing literature, our situation required an alternative approach to achieve some success.

To address the aforementioned challenges, we utilize a simple, yet effective method for RADAR-camera calibration using a visually intuitive, qualitative approach. This approach combines a 2D top-down RADAR point cloud with a monocular camera image, allowing the operator to visualize and qualitatively calibrate the camera and RADAR using multiple parameters to achieve adequate precision. More generically, this problem can be thought of as the minimization of error in a projection between two nearly perpendicular planes. This method can be employed in a live, on-water environment with unknown camera intrinsics and extrinsics, only requiring the existence of some arbitrary obstacles (e.g. land, ship, buoy) which are visible in both the camera image and RADAR returns to be used for alignment visualization and parameter tuning. An additional benefit of the user interface tool is that the calibration becomes resilient to hardware configuration changes; if the mounting of the camera is adjusted or the camera is replaced entirely, the user has the ability to adjust the calibration to the new environment. Moreover, pitch and roll measurements from an Inertial Measurement Unit are integrated into the calculations in order to reduce the calibration error that arises from continual pitch and roll changes.

In order to to project camera image point $(u_i, v_i)$ to RADAR image point $(u_r, v_r)$, we seek the 3x3 perspective transformation matrix $P_r^i$ such that $\begin{pmatrix} u_r \\ v_r \\ 1 \end{pmatrix} = P_r^i \begin{pmatrix} u_i \\ v_i \\ 1 \end{pmatrix}$

We first define the parameters of the perspective transformation which we seek to tune. Let $f$ be the field of view, $\alpha, \beta, \gamma$ represent the yaw, pitch and roll, respectively, along with $t_x$, $t_y$, $t_z$ for a translation in the x, y, and z dimensions. The pitch $\beta$ is computed by the multiplying the pitch $\beta_{imu}$ and pitch velocity $\dot{\beta_{imu}}$ acquired from the IMU with coefficients $\kappa$ and $\lambda$ such that $\left(\beta = \beta_{imu} \times \kappa + \dot{\beta_{imu}} \times \lambda\right)$. Roll $\gamma$ is calculated using the same method.

Similar to the work in [20], we seek the matrix $F$ from its constituent components. Let $R_\alpha R_\beta R_\gamma$ be the yaw, pitch and roll rotation matrices, $T$ is the translation matrix, and $P$ is the projection matrix, defined as follows:

$$P = \begin{pmatrix} 1/\tan(f/2) & 0 & 0 & 0 \\ 0 & 1/\tan(f/2) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & -1 & 1 \end{pmatrix} \quad (1)$$

$$T = \begin{pmatrix} 1 & 0 & 0 & t_x \\ 0 & 1 & 0 & t_y \\ 0 & 0 & 1 & t_z \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (2)$$

$$R_\alpha = \begin{pmatrix} \cos\alpha & -\sin\alpha & 0 & 0 \\ \sin\alpha & \cos\alpha & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (3)$$

$$R_\beta = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \sin\beta & \cos\beta & 0 \\ 0 & \cos\beta & -\sin\beta & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (4)$$

$$R_\gamma = \begin{pmatrix} \cos\gamma & 0 & \sin\gamma & 0 \\ 0 & 1 & 0 & 0 \\ -\sin\gamma & 0 & \cos\gamma & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (5)$$

$$F = PTR_\alpha R_\beta R_\gamma \quad (6)$$

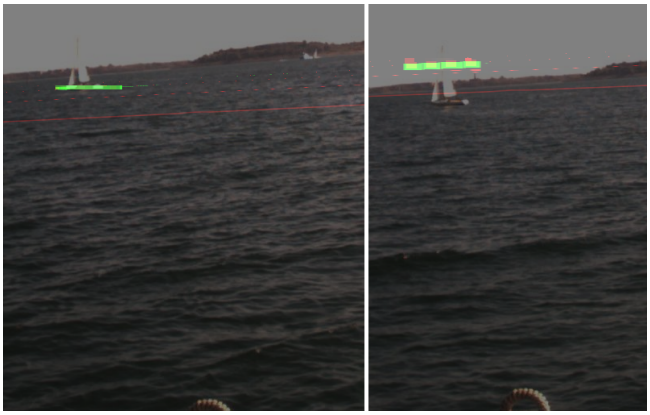

Fig. 3. Homography visualization. Sailboat (top left) and associated RADAR obstacle (green). Left: correct homography. Right: homography error during rapid pitch change. Also note the bow eye position change (bottom center), indicative of a change in camera pitch relative to our platform.

The projective transformation $P_r^i$ is then computed by using 4 static points from the the RADAR image dimensions *L* and matrix *F*, followed by solving a system of linear equations for these points, as described by Hartley and Zisserman [14].

Next, we define a conversion between radar image pixels and world coordinates. The RADAR image is generated by flattening the points in the RADAR pointcloud and placing them in a square image with an arbitrary side length of *L*=1024. Let $P_w^r$ be the projection from radar image pixels to world coordinates, then the transformation to a right-handed coordinate system is defined as

$$P_w^r = \begin{pmatrix} 0 & -1 & L/2 \\ -1 & 0 & L/2 \\ 0 & 0 & L/2R \end{pmatrix} \quad (7)$$

where *R* is the real-world range of the RADAR. Lastly, transforming camera image coordinates *(u,v)* to world coordinates *(x,y)* is simply

$$\begin{pmatrix} x \\ y \\ s \end{pmatrix} = P_w^r P_r^i \quad (8)$$

followed by a normalization of the resulting vector so that $s = 1$. Additionally, due to the isomorphic nature of the transformation, this process can be reversed in order to find image coordinates *(u,v)* from world coordinates *(x,y)*. See Figure 3 for an example of a visualized homography.

The resulting matrix can also be used to find an estimate of the horizon location in the camera image, which is constantly changing due to boat movement. Because we are primarily concerned with obstacles which are located on the surface of the water, we can use the location of the horizon to limit our search space, simultaneously decreasing processing time while filtering out some irrelevant detections.

Given an image pixel *(u,v)* and homography matrix *M*, where $M = P_w^r P_r^i$ (see equation (8)), we seek the vertical component *v* which represents the horizon at horizontal component *u*. Mathematically, this may be expressed as setting the denominator of the *v* component (where $v = (uM_{10} + vM_{11} + M_{12})/(uM_{20} + vM_{21} + M_{22})$

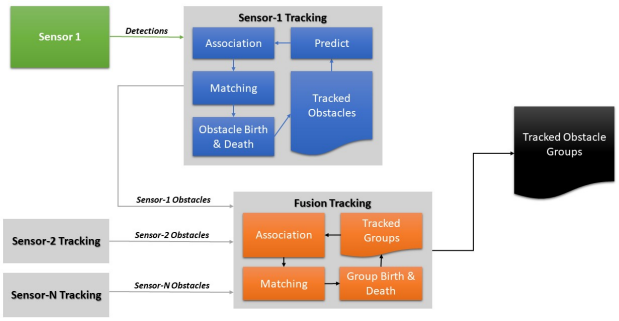


Fig. 4. Tracking and Fusion Framework

to zero and solving for $v$, while treating $u$ as the independent variable for a given perspective transformation matrix $M$:

$$\begin{aligned} &horizon(u, M) = \\ &uM_{20} + vM_{21} + M_{22} = 0 \\ &v = -(uM_{20} + M_{22})/M_{21} \end{aligned} \tag{9}$$

Future work in this area would focus on employing more sophisticated methods for heterogeneous sensor calibration, e.g. [53], [19], [9].

### C. Obstacle Tracking and Sensor Fusion

In order to operate autonomously in a maritime environment, it is important to understand information about nearby obstacles. Salient attributes include location, dimensions, classification, convex hull, and trajectory.

In our system, we have a collection of $n \subset N$ heterogeneous sensors, each with different noise characteristics and update rates. Each sensor $n$ generates $d \subset D_n$ detections (containing incomplete and/or inaccurate information) at time $t$. Our goal is to create a comprehensive list of all obstacles $K$ by combining the data from all sensors over the set of time points $T$, where $t \subset T$.

To accomplish this, we employ a simple, high performance, two-phase online method as pictured in Figure 4. At a high level, we manage the detections and obstacle state estimation at the sensor level using a single-frame global nearest neighbors approach, and then fuse the outputs of the sensor-level obstacle state into a comprehensive view for delivery to an external system and/or human operator. Here, we define *obstacle state*, or *obstacle*, as a collection of obstacle-related attributes to include position, heading, velocity, etc. and is defined in Table I.

*1) Phase One:* In the first phase, detections $d \subset D_n$ from sensor $n$ are associated with sensor-level obstacles $s, s \subset S_n$ via a single frame global-nearest neighbors approach. Each detection $d$ contains a subset of the obstacle state attributes, but requires at least a vessel-relative position. We then calculate a cost matrix between detections $d$ and sensor-level obstacles $S_n$ using user-configurable choice of distance metrics, e.g. Hellinger [18], Bhattacharyya [5], or Euclidean. Empirically, the Hellinger distance (which considers position covariance) was more effective than Euclidean distance for noisy detections such as those generated by our cameras, but was less effective than simple Euclidean distance for more accurate sensors such as LIDAR or RADAR.

After the cost matrix is calculated, we then attempt to match all detections in $D_n$ with the set of existing obstacles $S_n$, which were defined at time $t-1$. Here we use the Hungarian algorithm [24] to solve the bipartite graph matching problem with minimum cost in polynomial time. Recent work [50] [4] has shown this to be an effective combination for multi-object tracking in the autonomous car domain, and our research indicates this also applies to the autonomous surface vehicle domain as well.

Detections which do not match within a user-configurable maximum distance result in the birth of a new obstacle $s$, while existing obstacles within $S_n$ that are not matched with the current set of detections are removed after $i$ consecutive frames of failed matching.

Lastly, for all obstacles in $S_n$, the obstacle's position, velocity, and 3D rectangular dimensions are updated using a Kalman filter [21] with a constant velocity model. When the next set of detections $D_n$ arrives at time $t+1$, we execute the predict step of the Kalman filter for each obstacle $S_n$, and this updated information is used in the next cycle for detection association as described above.

*2) Phase Two:* In phase two (fusion), we employ a modified version of the phase one algorithm. In the fusion module, we define a fused obstacle $k \subset K$ where $k = \{s \subset S_0, ..., s \subset S_n\}$; i.e. each fused obstacle contains up to 1 obstacle $s$ from each of the sensors $S_n$. Distance cost, matching, and fused obstacle birth proceed as described above. However, rather than using a Kalman filter as in phase one, we combine the obstacle state from each obstacle $s$ to provide a fused state estimate. Fused obstacle position and covariance is computed using a fixed interval smoother [12], while size estimates are obtained from the largest estimate of the associated sensor obstacles $s \subset k$.

Finally, after the fusion step has completed, we output the list of obstacles over IP using the NMEA and JSON formats. This output can be used by external obstacle avoidance/navigation systems or human operators, for example. This information contains a rich set of features describing each fused obstacle, as listed in Table I.

TABLE I
GENERATED FEATURES FOR EACH OBSTACLE

| | | |
|---|---|---|
| Obstacle ID | Timestamp | Area |
| Classification Label | Classification Label Probability | 3D Position |
| 3D Position Covariance | Orientation | 3D Dimensions |
| 3D Linear Velocity | 3D Linear Velocity Covariance | 2D Convex Hull |

## IV. RESULTS

### A. Test Platform & Data

Experiments for this system were conducted on a live system in conjunction with the AUV Lab at MIT Sea Grant College. 60 minutes of data was collected along the Charles River from the MIT sailing pavilion towards the the Boston Harbor and back in October 2020 during fair-weather, daylight conditions. A total of 67 obstacles were observed, consisting of a variety of small- to medium-sized boats and static, free-standing obstacles such as pillars. The sensor platform was a Boston Whaler R/V Philos, with a sensor payload consisting of a Simrad Broadband 4G Radar, three forward-facing visible light cameras (FLIR Blackfly, BFLY-PGE-13E4C-CS, 1280x1024, 12fps) covering 140 degree horizontal FoV, a Velodyne VLP-16 LIDAR, and a SGB Systems Ellipse-D Dual Antenna RTK INS for GPS/IMU. The perception code was executed

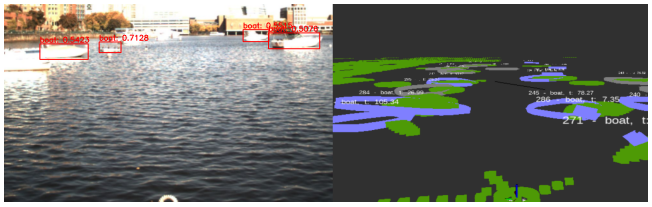

Fig. 5. Navigating a mooring field. Left: classified bounding boxes of boats. Right: 3D RViz view showing RADAR returns (green) and tracking of both classified (purple) and unclassified (gray) obstacles.

on a PC with an Intel i7-8700 3.2 GHz processor, 32GB of RAM, and a Nvidia GTX 1060 video card with 6GB VRAM, running Ubuntu 16.04. The machine learning model (trained on MS-COCO) for YOLOv4 [6] was downloaded from the author's website. WaSR [7] was not utilized due to insufficient hardware resources, but was shown to be effective in segmenting obstacle pixels in the camera image when executed offline against logged data.

### B. Evaluation

We are not aware of any annotated datasets which combine marine RADAR, LIDAR, and monocular camera to use as a baseline for this system. Therefore, to establish a ground truth, we use manual verification of the obstacles in the visible light cameras and compare that to the outputs of the sensors, both individually and as a whole. At the obstacle level, we define success as the system's ability to identify and localize free-standing obstacles which can be verified within a camera's field of view, along with the ability of the tracking and fusion system to correctly associate detections between sensors and track them over time.

System evaluation metrics include the total number of true positives (TP), false positives (FP), and false negatives (FN), along with the corresponding precision and recall scores. A definition of these metrics is available in [51]. Land-based obstacles are omitted.

TABLE II
SENSOR TRACKING EVALUATION

| Sensor | Range | TP | FP | FN | Precision | Recall |
|---|---|---|---|---|---|---|
| RADAR | 250m | 59 | 7 | 7 | 0.89 | 0.89 |
| Camera | 450m | 11 | 30+ | 1 | <0.27 | 0.92 |
| LIDAR | 50m | 15 | 1 | 3 | 0.94 | 0.83 |
| Totals | | 85 | 38+ | 11 | <0.69 | 0.89 |

## V. DISCUSSION

At the sensor level, detection and tracking generally worked well on the open water and in moderately congested areas, as shown in Figure 5. The marine RADAR tracker successfully identified and tracked obstacles up to 250m, and after some parameter adjustments, later demonstrated the ability to identify and track obstacles up to 500m (the configured limit of the RADAR) on logged data. The LIDAR tracker identified and tracked nearby boats in most cases. One of the false positives was generated from a wake, and false negatives can be attributed to obstacle generation parameters. Meanwhile, the camera tracker had little difficulty in identifying obstacles of the class 'boat', but had great difficulty in consistently localizing obstacles across frames, resulting in a large number of false positives. These difficulties can be attributed, in part, to a suboptimal homography and camera tracking parameter configuration. However, the primary source of the failures is the difficulty in maintaining a proper homography between the camera and the RADAR while the boat's pitch is changing rapidly (see figure 3). Additionally, not included in the metrics is a significant number of false positive camera detections which where generated from a single miscalibrated camera. After adjusting the calibration and embarking on a subsequent mission, these false positives were significantly reduced.

In small, confined areas (e.g. locks, narrow channels, under bridges), sensor-level detection and tracking did not perform well. The large, amorphous blobs produced by marine RADAR were generally filtered out as land masses, and were the source of the RADAR-level false negatives. Similar to the RADAR, the LIDAR data was often filtered out as land mass, leading to missed detections of two boats when in close proximity. In both instances, however, the boats were correctly detected by the cameras, though in one of those instances, the detection did not occur until 20m away due to lens flare.

The sensor fusion system performed adequately at combining the sensor-level obstacle data that was correctly localized, but failed at times to correctly localize the obstacle relative to the perceived ground truth, and failed in a few instances where false positive sensor obstacles were not consolidated into a single instance. With some additional work in combining sensor obstacle data more intelligently, the fusion system's accuracy and reliability would increase.

## VI. CONCLUSION

Overall, the system performed adequately on a live vessel, but there are many ways to improve this system in the future. Due to its use of visible light cameras, object detection capabilities are reduced during adverse environmental conditions (night, fog, etc). Research into object detection and segmentation of infrared imagery could improve detection capability in these scenarios. The RADAR-camera calibration remains an error-prone process with little margin of error, and an improved system would address these shortfalls to improve localization. With the integration of nautical charts, known land masses could be filtered and reduce the number of false positive detections, but charts alone do not provide sufficient obstacle information (especially while traversing confined areas), necessitating complementary sensor information. A probabilistic occupancy map could be generated from the various sensor inputs, complimenting the obstacle detection algorithms and improve autonomous decision-making in confined areas. AIS data could be integrated for improved obstacle type identification, rather than solely relying on visible light classification. A custom image classification model could be built to better identify more types of maritime obstacles in order to leverage that information for COLREGs compliance. More robust sensor fusion algorithms (e.g. GLMB [39], JIPDA [30]) may improve the output of the sensor fusion system. Subsampling the visible light image (e.g. across the horizon) may improve object detection range.

## VII. ACKNOWLEDGEMENTS

Partial support for this project was provided by Lockheed Martin. Partial support 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. The authors thank Shailesh Nirgudkar for feedback on a draft of this paper.

## References


[1] *1.2 Billion Worldwide Unmanned Surface Vehicle Industry to 2027 - Impact of COVID-19 on the Market - ResearchAndMarkets.com*. 2020. URL: https://www.businesswire.com/news/home/20200811005433/en/1.2-Billion-Worldwide-Unmanned-Surface-Vehicle-Industry (visited on 09/11/2020).

[2] Anubhav Agarwal, CV Jawahar, and PJ Narayanan. "A survey of planar homography estimation techniques". In: *Centre for Visual Information Technology, Tech. Rep. IIIT/TR/2005/12* (2005).

[3] Michael R Benjamin et al. "Obstacle avoidance using multi-objective optimization and a dynamic obstacle manager". In: *IEEE Journal of Oceanic Engineering* 44.2 (2019), pp. 331–342.

[4] Alex Bewley et al. "Simple Online and Realtime Tracking". In: *CoRR* abs/1602.00763 (2016). arXiv: 1602.00763. URL: http://arxiv.org/abs/1602.00763.

[5] Anil Bhattacharyya. "On a measure of divergence between two multinomial populations". In: *Sankhyā: the indian journal of statistics* (1946), pp. 401–406.

[6] Alexey Bochkovskiy, Chien-Yao Wang, and Hong-Yuan Mark Liao. *YOLOv4: Optimal Speed and Accuracy of Object Detection*. 2020. arXiv: 2004.10934 [cs.CV].

[7] Borja Bovcon and Matej Kristan. "A water-obstacle separation and refinement network for unmanned surface vehicles". In: *arXiv preprint arXiv:2001.01921* (2020).

[8] M. DeFilippo et al. "The Remote Explorer IV: An Autonomous Vessel for Oceanographic Research". In: *OCEANS 2019 - Marseille*. 2019, pp. 1–8.

[9] B. Della Corte et al. "Unified Motion-Based Calibration of Mobile Multi-Sensor Platforms With Time Delay Estimation". In: *IEEE Robotics and Automation Letters* 4.2 (2019), pp. 902–909.

[10] Michele Fiorini, Andrea Pennisi, and Domenico Bloisi. "Optical Target Recognition for Drone Ships". In: *The International Conference on Marine Navigation and Safety of Sea Transportation (TRANSNAV 2017)*. June 2017, pp. 371–375. DOI: 10.1201/9781315099132-64.

[11] Daniel Frank, Andrew Gray, and Eric Schwartz. "PropaGator 2015: UF autonomous surface vehicle". In: *AUVSI Foundation's 7th Annual RoboBoat Competition* (2015).

[12] D. Fraser and J. Potter. "The optimum linear smoother as a combination of two optimum linear filters". In: *IEEE Transactions on Automatic Control* 14.4 (1969), pp. 387–390. DOI: 10.1109/TAC.1969.1099196.

[13] A Gray et al. "Propagator 2013: Uf autonomous surface vehicle". In: *AUVSI Foundation's 6th Annual RoboBoat Competition, Virginia Beach, VA* (2013).

[14] Richard Hartley and Andrew Zisserman. *Multiple view geometry in computer vision*. Cambridge university press, 2003.

[15] Hordur K Heidarsson and Gaurav S Sukhatme. "Obstacle detection and avoidance for an autonomous surface vehicle using a profiling sonar". In: *2011 IEEE International Conference on Robotics and Automation*. IEEE. 2011, pp. 731–736.

[16] Oystein Kaarstad Helgesen et al. "Sensor Combinations in Heterogeneous Multi-sensor Fusion for Maritime Target Tracking." In: *2019 22th International Conference on Information Fusion (FUSION), Information Fusion (FUSION), 2019 22th International Conference on* (2019), pp. 1–9. ISSN: 978-0-9964527-8-6. URL: https://umasslowell.idm.oclc.org/login?url=http://search.ebscohost.com/login.aspx?direct=true&db=edseee&AN=edseee.9011297&site=eds-live.

[17] Øystein Kaarstad Helgesen et al. "Low Altitude Georeferencing for Imaging Sensors in Maritime Tracking". In: *IFAC World Congress* (2020).

[18] E. Hellinger. "Neue Begründung der Theorie quadratischer Formen von unendlichvielen Veränderlichen." German. In: *J. Reine Angew. Math.* 136 (1909), pp. 210–271. ISSN: 0075-4102; 1435-5345/e.

[19] Kaihong Huang and Cyrill Stachniss. "Extrinsic multi-sensor calibration for mobile robots using the Gauss-Helmert model". In: *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE. 2017, pp. 1490–1496.

[20] I. I. *How to calculate perspective transform for OpenCV from rotation angles?* 2018. URL: https://stackoverflow.com/a/20089412 (visited on 04/30/2020).

[21] R. E. Kalman. "A New Approach to Linear Filtering And Prediction Problems". In: *ASME Journal of Basic Engineering* (1960).

[22] Chunming Kuang et al. "An Applied Method for Clustering Extended Targets with UHF Radar". In: *IEEE Access* (2020).

[23] D Kwame Minde Kufoalor et al. "Autonomous maritime collision avoidance: Field verification of autonomous surface vehicle behavior in challenging scenarios". In: *Journal of Field Robotics* 37.3 (2020), pp. 387–403.

[24] Harold W. Kuhn. "The Hungarian Method for the assignment problem". In: *Naval Research Logistics Quarterly* 2 (1955), pp. 83–97.

[25] Jacoby Larson, Michael Bruch, and John Ebken. "Autonomous navigation and obstacle avoidance for unmanned surface vehicles". In: *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series*. Vol. 6230. Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series. May 2006, p. 623007. DOI: 10.1117/12.663798.

[26] Jacoby Larson et al. *Advances in autonomous obstacle avoidance for unmanned surface vehicles*. Tech. rep. SPACE and NAVAL WARFARE SYSTEMS CENTER SAN DIEGO CA, 2007.

[27] T. Manderson and G. Dudek. "GPU-Assisted Learning on an Autonomous Marine Robot for Vision-Based Navigation and Image Understanding". In: *OCEANS 2018 MTS/IEEE Charleston*. 2018, pp. 1–6.

[28] Jon Muhovič et al. "Obstacle Tracking for Unmanned Surface Vessels Using 3-D Point Cloud". In: *IEEE Journal of Oceanic Engineering* (2019).

[29] Jon Muhovic et al. "Depth Fingerprinting for Obstacle Tracking using 3D Point Cloud". In: *Proc. 23rd Comput. Vis. Winter Workshop*. 2018, pp. 71–78.

[30] D. Musicki and R. Evans. "Joint integrated probabilistic data association: JIPDA". In: *IEEE Transactions on Aerospace and Electronic Systems* 40.3 (2004), pp. 1093–1099. DOI: 10.1109/TAES.2004.1337482.

[31] Håkon Gjertsen Norbye. "Real-time sensor fusion for the ReVolt model-scale vessel". In: *Master's thesis, Norwegian University of Science and Technology* (2019).

[32] Ronald O'Rourke. *Navy Large Unmanned Surface and Undersea Vehicles: Background and Issues for Congress*. Tech. rep. 2020. URL: https://fas.org/sgp/crs/weapons/R45757.pdf.

[33] International Maritime Organization. *COLREG: Convention on the International Regulations for Preventing Collisions at Sea*. 1972.

[34] Philippe Paccaud and DA Barry. "Obstacle detection for lake-deployed autonomous surface vehicles using RGB imagery". In: *PloS one* 13.10 (2018), e0205319.

[35] Jeonghong Park, Jinwhan Kim, and Nam-sun Son. "Passive target tracking of marine traffic ships using onboard monocular camera for unmanned surface vessel". In: *Electronics Letters* 51.13 (2015), pp. 987–989.

[36] Miro Petković, Igor Vujović, and Ivica Kuzmanić. "An Overview on Horizon Detection Methods in Maritime Video Surveillance". In: *Transactions on Maritime Science* 9.01 (2020), pp. 106–112.

[37] Dilip K Prasad et al. "Video processing from electro-optical sensors for object detection and tracking in a maritime environment: a survey". In: *IEEE Transactions on Intelligent Transportation Systems* 18.8 (2017), pp. 1993–2016.

[38] Joseph Redmon and Ali Farhadi. "Yolov3: An incremental improvement". In: *arXiv preprint arXiv:1804.02767* (2018).

[39] Stephan Reuter et al. "The labeled multi-Bernoulli filter". In: *IEEE Transactions on Signal Processing* 62.12 (2014), pp. 3246–3260.

[40] Paul Robinette et al. "Sensor Evaluation for Autonomous Surface Vehicles in Inland Waterways". In: *OCEANS 2019-Marseille*. IEEE. 2019, pp. 1–8.

[41] Radu Bogdan Rusu and Steve Cousins. "3d is here: Point cloud library (pcl)". In: *2011 IEEE international conference on robotics and automation*. IEEE. 2011, pp. 1–4.

[42] Matteo Schiaretti, Linying Chen, and Rudy R Negenborn. "Survey on autonomous surface vessels: Part II-categorization of 60 prototypes and future applications". In: *International Conference on Computational Logistics*. Springer. 2017, pp. 234–252.

[43] Frederik ET Schöller et al. "Assessing deep-learning methods for object detection at sea from LWIR images". In: *IFAC-PapersOnLine* 52.21 (2019), pp. 64–71.

[44] Andrea Sorbara et al. "Design of an obstacle detection system for marine autonomous vehicles". In: *OCEANS 2015-Genova*. IEEE. 2015, pp. 1–8.

[45] Mina Sorial et al. "Towards a Real Time Obstacle Detection System for Unmanned Surface Vehicles". In: *OCEANS 2019 MTS/IEEE SEATTLE*. IEEE. 2019, pp. 1–8.

[46] Lorenzo Steccanella et al. "Waterline and obstacle detection in images from low-cost autonomous boats for environmental monitoring". In: *Robotics and Autonomous Systems* 124 (2020), p. 103346.

[47] Einar S Ueland, Roger Skjetne, and Andreas R Dahl. "Marine autonomous exploration using a lidar and slam". In: *International Conference on Offshore Mechanics and Arctic Engineering*. Vol. 57724. American Society of Mechanical Engineers. 2017, V006T05A029.

[48] Tao Wang et al. "Integrating Millimeter Wave Radar with a Monocular Vision Sensor for On-Road Obstacle Detection Applications". In: *Sensors (Basel, Switzerland)* 11 (Sept. 2011), pp. 8992–9008. DOI: 10.3390/s110908992.

[49] Xiao Wang et al. "On-road vehicle detection and tracking using MMW radar and monovision fusion". In: *IEEE Transactions on Intelligent Transportation Systems* 17.7 (2016), pp. 2075–2084.

[50] Xinshuo Weng et al. "3D Multi-Object Tracking: A Baseline and New Evaluation Metrics". In: *IROS* (2020).

[51] Wikipedia contributors. *Precision and Recall — Wikipedia, The Free Encyclopedia*. [Online; accessed 22-Oct-2020]. 2020. URL: https://en.wikipedia.org/wiki/Precision_and_recall.

[52] Jia-yuan Zhuang et al. "Radar-based collision avoidance for unmanned surface vehicles". In: *China Ocean Engineering* 30.6 (2016), pp. 867–883.

[53] David Zuñiga-Noël et al. "Automatic Multi-Sensor Extrinsic Calibration for Mobile Robots". In: *CoRR* abs/1906.04670 (2019). arXiv: 1906.04670. URL: http://arxiv.org/abs/1906.04670.