An Overview of MIT-Olin's Approach in the AUVSI RobotX Competition

Arthur Anderson, Erin Fischell, Thom Howe, Tom Miller, Arturo Parrales-Salinas, Nick Rypkema, David Barrett, Michael Benjamin, Alex Brennen, Michael DeFillipo, John J. Leonard, Liam Paull, Henrik Schmidt, Nick Wang, Alon Yaari

Abstract The inaugural RobotX competition was held in Singapore in Oct. 2014. The purpose of the competition was to challenge teams to develop new strategies for tackling unique and important problems in marine robotics. The joint team from Massachusetts Institute of Technology (MIT) and Olin College was chosen as one of 15 competing teams from five nations (USA, South Korea, Japan, Singapore and Australia). The team received the surface vehicle platform, the WAM-V (Fig. 1) in Nov. 2013 and spent a year building the propulsion, electronic, sensing, and algorithmic capabilities required to complete the five tasks that included navigation, underwater pinger localization, docking, light sequence detection, and obstacle avoidance. Ultimately the MIT/Olin team narrowly won first place in a competitive field. This paper summarizes our approach to the tasks, as well as some lessons learned in the process. As a result of the competition, we have developed a new suite of open-source tools for feature detection and tracking, realtime shape detection from imagery, bearing-only target localization, and obstacle avoidance. [1]

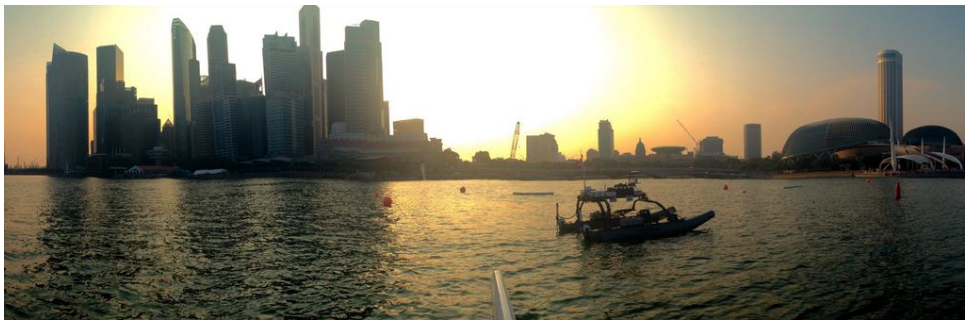

Fig. 1 The WAM-V [7] on the water in Singapore at the RobotX competition.

Arthur Anderson, Erin Fischell, Thom Howe, Tom Miller, Nick Rypkema, Michael Benjamin, Alex Brennan, John J. Leonard, Liam Paull, Henrik Schmidt, Nick Wang, Alon Yaari
MIT, 77 Massachusetts Avenue, Cambridge, MA 02139, {arthura, emf43, thomhowe, mille219, rypkema, mikerb, vab, jleonard, lpaull, henrik, hcnwang, ayaari}@mit.edu

Arturo Parrales-Salinas
Tufts University, 419 Boston Ave, Medford, MA 02155 Arturo.Parrales_Salinas@tufts.edu

David Barrett
Olin College, Olin Way, Needham, MA 02492 e-mail: David.Barrett@olin.edu

Michael DeFillipo
MIT Sea Grant, 292 Main Street, Cambridge, MA 02142 e-mail: mikedef@mit.edu

[1] See http://robotx.mit.edu for more details and updates.

1 Introduction

The inaugural RobotX competition, hosted by the association for unmanned vehicle systems international (AUVSI) Foundation, was held in Singapore in October 2014. The motivation for the competition was to increase the capabilities of marine vehicle systems to perform commercial tasks and operate in the vast and challenging ocean environment. Much larger in scope than previous competitions, such as RoboBoat and RoboSub, this was the largest autonomous surface vehicle (ASV) competition ever held. In total 15 teams competed from five different countries (USA, South Korea, Singapore, Australia, and Japan). Each of the 15 teams were provided with an identical platform, shown in Fig. 2, and were responsible for equipping it with sensors, propulsion, electrical systems, and onboard autonomy to achieve the tasks.

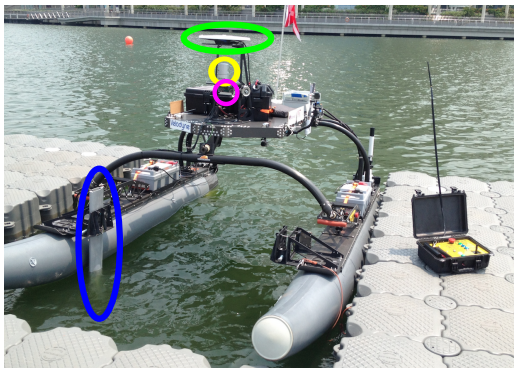

Fig. 2 The "WAM-V" ASV platform used in the competition. Sensor payload includes: GPS (green), 3D laser scanner (yellow), camera (pink), and sonar transducer and mount (blue).

The competition consisted of five tasks:

- Task 1: Navigate through two sets of colored buoy gates;
- Task 2: Report the location of an underwater pinger and also the color of the closest buoy to the pinger;
- Task 3: Identify the correct docking location based on a placard on the seawall and then subsequently dock;
- Task 4: Find a buoy that is emitting an LED light pattern and then report the light pattern;
- Task 5: Enter an obstacle field through a buoy gate (specified by color) and then navigate through a densely cluttered field of obstacles, and finally exit through the specified gate.

Each task had a unique scoring system and the sum of all task points was used to rank teams. After three qualification days, the top six ranked teams advanced to the finals. In the finals, the points accumulated in the last attempted run were used.

In this paper we summarize our approach to each of the five tasks. These required basic capabilities such as object detection and autonomy, as well as task-specific capabilities such as pattern recognition and acoustic target localization. The remainder of the paper is structured as follows: In Sec. 2, we detail our laser/vision based approach to object detection and tracking for navigation tasks (required for Task 1, 2, 4, and 5). In Sec. 3, we discuss our approach to specific vision-based pattern identification tasks (Task 3 and 4). In Sec. 4, we present the particle-filter based acoustic localization system (Task 2). In Sec. 5, we present an overview of our approach to autonomy and control based on behavior-based multi-objective optimization. In Sec. 6, we provide some details about the choice of hardware used. Finally we provide some of the competition results in Sec. 7 and some conclusions in Sec. 8.

2 Object Detection, Tracking, and Classification

A prerequisite for Tasks 1, 2, 4, and 5 is to be able to detect and track objects floating on the water surface. Above-water perception onboard the vehicle was achieved through a combination of 3D laser and vision. Laser-based sensing was particularly effective in this case since the water surface only produced weak returns that could be easily removed through laser intensity filtering, leaving only solid objects such as buoys.

Algorithm 1 Object Detection and Tracking

Input: Laser scan
Output: Feature List
1: Cull points outside of desired sector
2: Downsample to voxel grid
3: Euclidean clustering
4: Update the persistent cluster list
5: Transform persistent features to world frame
6: Try to assign feature color through image sub-windowing
7: Associate features and update tracked feature list

Since laser provides limited color information, object detections were fed to a vision system for classification. An overview of the approach is summarized in Algorithm 1.

2.1 Laser-Based Feature Detection

Each point in the laser scan, $p = \{r, \phi, \theta, I\}$, is a tuple consisting of a range r, an azimuth ϕ, an elevation θ, and an intensity I. One scan of laser data consists of a collection of N points $P = \{p_i\}_{i=1..N}$. The points are first culled using thresholds for minimum and maximum range and azimuth, as well as minimum intensity:

$$P_c = \{p_i | r_{min} < r_i < r_{max}, \phi_{min} < \phi_i < \phi_{max}, I_{min} < I_i\}. \quad (1)$$

These points are then downsampled using a voxel grid and ordered into clusters, $\mathscr{C} = \{C^j\}_{j=1..J}$, $C^j = \{P^j, \mu^j\}$, where P^j and μ_j are the set of points and centroid of cluster j respectively. We wish to be able to detect buoys at the maximum possible range, at which point there may be only one or two returns from a buoy. In order to mitigate the impact of false returns while still being able to track small features at long distances, we use a temporal persistence filter. A persistent cluster list, $\mathscr{C}^{pcl} = \{\mathscr{C}, K\}$, is maintained, where K is the "lifetime" of the cluster. As a new set of clusters arrives at time k, they are fused with the persistent cluster list. For each new cluster, if its centroid, μ, is within ε of one of the centroids of the clusters in $\mathscr{C}^{pcl}$, then the associated cluster's lifetime is incremented, otherwise the cluster is added to the persistent cluster list with a lifetime of $K = 1$. A laser scan and associated camera image are shown in Fig. 3. This particular snapshot is from the obstacle avoidance task. In this case there are four persistent features.

The set of persistent clusters with a lifetime larger than K_{min} are deemed to be active objects in the world and are transformed to world coordinates and added as features, f:

$$f^j = T_l^w \mu^j \quad (2)$$

where the transformation T_l^w transforms the centroid of the cluster in the laser frame to a point in the global frame. This feature is rejected if it is outside of course boundaries.

In order to compute T_l^w we directly used the output from our GPS sensor, which provided a stable pose estimate in practice. Nevertheless, a more reliable approach would be to implement a full SLAM system, or use some other form of marine vehicle navigation [11].

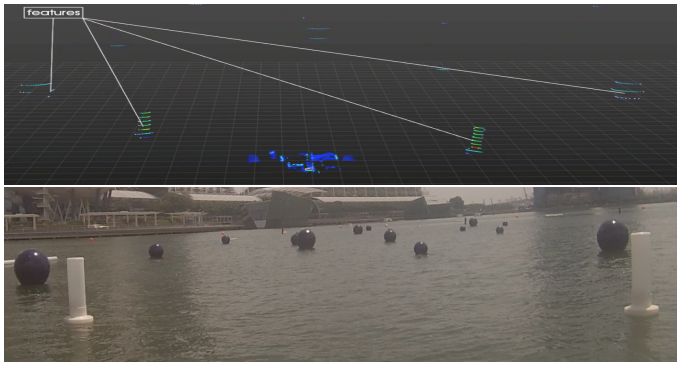

Fig. 3 Object detection from point cloud Top: Point cloud from 3D laser with buoys identified. Bottom: Corresponding image from camera used for buoy color detection.

2.2 Buoy Detection

Each reception of an object detection from the laser triggers an attempt to classify the color of the object. The feature location is back-projected into the camera frame to try and identify color [5]:

$$\begin{bmatrix} l_u^j & l_v^j & 1 \end{bmatrix}^T = \mathbf{K} \begin{bmatrix} \frac{l_x^j}{l_z^j} & \frac{l_y^j}{z_l^j} 1 \end{bmatrix}^T, \tag{3}$$

where (l_u^j, l_v^j) and (l_x^j, l_y^j, l_z^j) are the locations of the feature in pixel and world coordinates respectively, and $\mathbf{K}$ is the camera calibration matrix. A sub-windowed image around around the landmark pixel location is created, which is then subjected to a series of thresholding operations in the hue-saturation-value (HSV) color space. Using the HSV colorspace is beneficial for color detection in images because it is less sensitive to lighting conditions as the majority of the color information should be contained within the hue channel and the aggressive sub-windowing was found to be critical to avoid false detections.

Fig. 4 shows a sub-windowed image from one of the test trials as well as the output of the red filter showing correct color identification.

2.3 Feature association

We use a simple nearest neighbor [1] approach to associate features. If an incoming feature is at location l^j then the feature is associated to feature i if two conditions are met:

$$\begin{aligned} l^i &= \arg\min_{l \in L} ||l - l^j|| \\ d_{min} &< ||l - l^j|| \end{aligned} \tag{4}$$

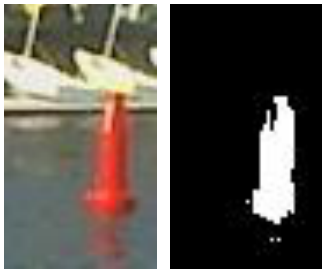

Fig. 4 Left: Sub-windowed image from camera. Right: Output from "red" color segmentation filter.

We refer to these associated features as "tracked features" $L^m = \{l^{1..|m|}, c^{1..|m|}\}$ where $|m|$ is the number of times that tracked feature L^m has been detected and the c values corresponds to the color decisions made for each detection. The final set of M distinct tracked features

$\mathscr{L} = \{L^m\}_{m=1..M}$ are used by the control and autonomy system (Sec. 5) to complete the specific tasks.

3 Pattern Identification

Pattern recognition was a required capability for two tasks. The first was Task 3 where a spatial pattern, either a cross, circle, or triangle, was used to identify the correct bay for docking. The second was Task 4 which required the identification of a temporal pattern. In both cases, aggressive sub-windowing in the image was performed to guide the visual search and decrease false positives while maintaining low computation.

3.1 Placard Detection for Docking

The key objectives for our placard detector were:

1. Robustness to degradation caused by motion, scale and perspective transformation from different viewing positions, warp and occlusion caused by wind, and variants of color from light condition, and
2. Speed and accuracy to support real-time decision-making.

We tackle this problem by using two-step pipeline. First, a *detection* phase identifies candidate regions and we subsequently process each region in a *decoding* stage to see if it matches any of the three placards.

Detection

To minimize unnecessary computation and to avoid looking for placards in nearly empty image regions, in the first stage we extract candidate regions using Extremal Regions (ERs) [9]. An ER is a region R whose outer boundary pixels ∂R have strictly higher values in the intensity channel $\mathbf{C}$ than the region R itself, i.e., $\forall p \in R, q \in \partial R : \mathbf{C}(p) < \theta < \mathbf{C}(q)$, where θ is the threshold of the ER. Let a grayscale input frame $\mathbf{I}$ be a mapping $\mathbf{I} : D \subset \mathbb{R} \to \{0,\ldots,255\}$. $\mathscr{R}_b$ and $\mathscr{R}_w$ donate the sets of detected ERs from $\mathbf{I}$ and inverted $\mathbf{I}$, respectively. We extracted features $\mathbf{F}$ for each region in $\mathscr{R}_b$ and $\mathscr{R}_w$, and then filter according to size, aspect ratio, and number of holes. We observed that a placard is designed as a black symbol on a white board. The set of candidate regions $\mathscr{R}_c \subset \mathscr{R}_b$ is formed when a region r_b in $\mathscr{R}_b$ satisfies $r_b \cap r_w = r_b$, as well as certain conditions on relative size, location, and intensity of r_b and r_w, where $r_w \in \mathscr{R}_w$. Imposing such constraints drastically reduced false positives, and typically only the black symbols on placards are detected.

Decoding

In the decoding stage, we desired very high precision at the expense of recall since occasional missed detections are tolerable but false positives will cause significant problems. During the competition the system needed to be able to adapt quickly and there were only limited training examples of placards available. We set up one template for each of circle, triangle, and cross, and match a candidate region $r_c \in \mathscr{R}_c$ to one of the placards, when the number of good matching keypoints is higher than a threshold. SIFT [6] and FAST [12] keypoints and SIFT descriptors are appealing choices to distinguish each placard. For example, the cross contains many SIFT keypoints, typically corners surrounded by gradients, and triangle contains FAST keypoints (typically "sharp" corners), where a a pixel p has

Fig. 5 Placard feature detection The three correctly detected placards are circled in blue, red, and green.

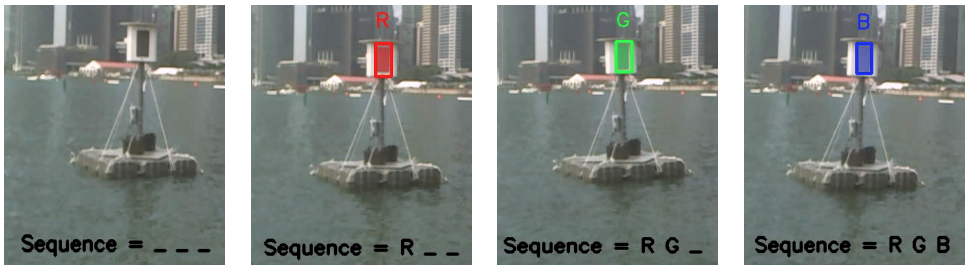

Fig. 6 Light tower sequence detection.

contiguous n pixels in the circle around p brighter or darker. Fig. 5 demonstrates the decoded candidate regions shown with blue, red, and green circles. The computation runs at two frames per second for an image resolution of 1280×720 pixels.

3.2 Light Buoy Sequence Detection

The light buoy color sequence consists of an LED panel mounted on top of a buoy that emits a sequence of three colors (each for half a second), followed by a two second break. Detection of color on the LED is done with a similar process as for the buoy color detection (Sec. 2.2) except that there is an added temporal component required to detect the sequence. An overview of the approach is given in Algorithm 2. An example of a sequence being detected is shown in Fig. 6.

Algorithm 2 Light Buoy Sequence Detection

Input: Video stream
Output: Light sequence Φ

$\Phi \leftarrow \emptyset$
Wait until first detection is made
Wait until no detection is found for 2 seconds
while $|\Phi| < 3$ **do**
  $C \leftarrow$ color detected in image
  if $C \neq$ last entry in Φ **then**
    $\Phi \leftarrow \Phi \bigcup C$
  end if
  if No Detection **then**
    Return to Step 3
  end if
end while

This system requires the light buoy to be within the field of view of the camera for minimum of four seconds (the end a sequence and then one full sequence). If no detections are being made the segmentation thresholds are adapted automatically to be more admissive. Similarly, if the pause in the sequence is never being found (caused by false detections) then the thresholds are adaptively made more restrictive.

4 Acoustic Sensing

The process of localizing the pinger in Task 2 had two main components: First, relative bearing measurements are obtained from processing the signals received at the hydrophone. Second, subsequent bearing measurements are combined with a particle filter to yield a final estimate of the pinger location.

4.1 Relative Bearing Measurements

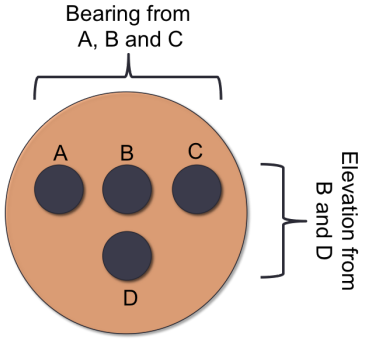

Fig. 7 "T"-shaped Acoustic Array

The acoustic system consisted of a 4-element hydrophone phased array, a custom amplification and filtering board (AFB), a data acquisition board (DAB), and a computer. The phased array was assembled into a 'T' shape (see Fig. 7) with uniform element spacing $d = 1.9cm$. This formed two subarrays, one horizontal for use in bearing estimation and one vertical for use in elevation estimation. The signal from each hydrophone channel passed through a 10kHz Sallen-key high-pass filter, a 2x amplifier, and then a 50kHz Sallen-key low-pass filter on the AFB [13]. The resulting signal was converted to digital by the DAB and used to determine pinger location from four channels of hydrophone data. First, matched filtering on the first acoustic channel was used to identify if a ping of the correct frequency occurred [10]. Conventional (delay-and-sum) beamforming was applied to the array data, and the maximum value in the beampattern was used to determine bearing to the pinger [14]. Let z represent the direction along the array. The discrete array has elements at locations $\mathbf{z} = [-d, 0, d]$. The goal of beamforming is to find the angle of incidence, θ_0, of the signal from a pinger with frequency f_0. This gives a wavenumber $k_0 = 2\pi f_0/c$. The z-component of the wavenumber can be expressed in terms of 'look' direction θ:

$$k_z = k_0 cos\theta. \tag{5}$$

This component of the wavenumber is used to calculate the delay vector $\mathbf{v}$:

$$\mathbf{v}(\theta) = e^{-j\mathbf{z}k_z}. \tag{6}$$

Delay-and-sum beamforming [14] is then applied by first multiplying the snapshot time series, $\mathbf{x} = [x_1, x_2, x_3]$, with the delay vector and then taking the Fourier transform:

$$Y = \mathscr{F}(\mathbf{x}\mathbf{v}'). \tag{7}$$

The beampattern function at look angle θ is the value of $\mathbf{Y}$ at frequency f_0, $B(\theta) = \mathbf{Y}(f_0)$. The bearing to the pinger of frequency f_0 is the look angle that results in the maximum for the beampattern:

$$\theta_0 = \arg\max_{\theta} \|B(\theta)\| \tag{8}$$

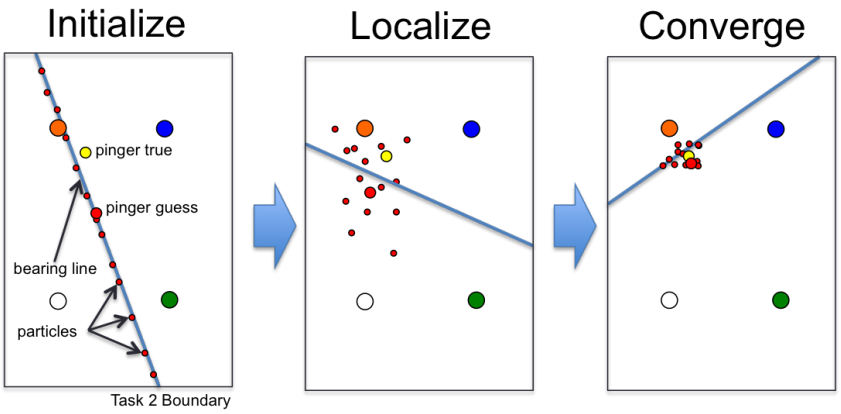

Fig. 8 Particle Filter Localization Left: The particles are initialized after the first bearing line is received. Middle: As more bearings are received, the particles begin to localize on the pinger location. Right: After more information is received the particles converge on a single point.

A similar process was used to determine the pinger elevation angle: conventional beamforming was applied to the vertical array (elements 1 and 3) and the beamforming angle with the maximum response identified as the elevation angle of the pinger.

4.2 Particle Filter Pinger Localization

The estimated elevation and bearing angles reported by the hydrophone system were used by a particle filter [8] to estimate the possible pinger location. An overview of the method is illustrated in Fig. 8. When the first relative bearing measurement is received, the particles are initialized uniformly along the portion of the bearing line that falls within the task boundary.

When the second and subsequent bearing measurements are received, the particles are each given a weight based on their proximity to the new bearing line based on the following equation:

$$w_i^t = w_i^{t-1} \frac{p(r_i^t|\zeta_i^t)p(\zeta_i^t|\zeta_i^{t-1})}{q(\zeta_i^t|\zeta_i^{0:t}, r_i^t)} \tag{9}$$

where ζ_i^t is the xy-positon of particle i at time t, and r_i^t is the orthogonal distance from the particle position, ζ_i^t, to the line anchored at the current vehicle position, x_t, y_t with slope corresponding to the bearing measurement, θ_0 calculated in (8). If we set the transition prior $p(\zeta_i^t|\zeta_i^{t-1})$ equal to the importance function $q(\zeta_i^t|\zeta_i^{0:t}, r_t)$ [8], and assume a normal distribution for $p(r_i^t|\zeta_i^t)$, we can simplify (9) to:

$$w_i^t = w_i^{t-1} \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(r_i^t)^2}{2\sigma^2}} \tag{10}$$

Finally, we use sequential importance resampling to avoid particle depletion. This involves a check to determine if the effective number of particles N_{eff} has fallen below a threshold $N_{threshold}$. The effective number of particles is:

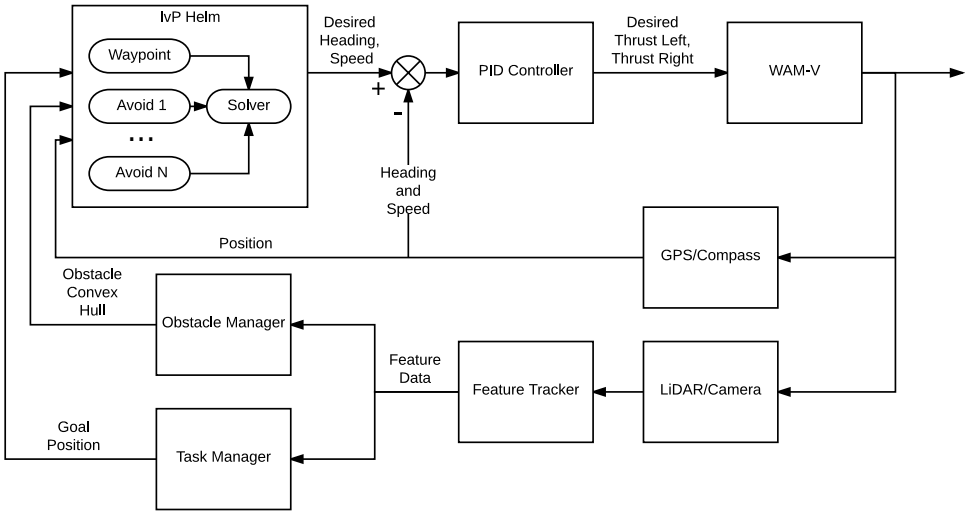

Fig. 9 Control system used in autonomy mode. Behaviors in the IvP Helm generate objective functions which are weighed at runtime to determine a best choice desired heading and speed. These values are tracked by an inner loop PID controller.

$$N_{eff} = \frac{1}{\sum_{i=1}^{N}(w_i^2)} \tag{11}$$

where N is the total number of particles. The best guess for the pinger location at any given time is computed is the average location of the particles.

5 Autonomy and Control

The operation of the vehicle is broadly characterized into two modes: (1) *Autonomy*, which is used for moving the boat around and avoiding obstacles, and (2) *Observation*, which is used for keeping the vehicle's sensors pointed in a specific direction.

5.1 Autonomy

In autonomy mode, the vehicle has to balance different objectives, such as transiting to a goal point while avoiding obstacles. This balance is achieved using multi-objective optimization with interval programming (IvP), [3], [4], where each goal is represented by a piecewise linearly-defined objective function for evaluation in conjunction with all other active objective functions. The optimization engine on-board the ASV considers and solves for the resultant maneuver (ordered course, speed) using

$$\overrightarrow{x^*} = \arg\max_{\overrightarrow{x}} \sum_{i=1}^{k} (w_i \cdot f_i(\overrightarrow{x})) \tag{12}$$

where each $f_i(x_1, ..., x_n)$ is an objective function for the i^{th} of k active goal, and the weights, w_i are used to prioritize the different objectives.

An overview of the control methodology in the autonomy mode is shown in Fig. 9. The outer loop desired heading and speed values are generated by the IvP Helm which operates within the mission oriented operating suite (MOOS) environment [2]. In our case, each feature outputted by the feature tracker (described in Sec. 2) is treated as an obstacle and spawns a new obstacle avoidance behavior. These avoidance behaviors ("Avoid 1" to "Avoid *N*" in Fig. 9) are then used to prioritize actions that move the vehicle away from obstacles. These are weighed with a waypoint behavior that is used to steer the vehicle towards the desired goal.

5.1.1 The Obstacle Manager

The association of features is performed by the feature tracker that processes the clustered output from the laser. Due to noise in the system, as well as the fact that features (such as buoys) may be actually moving on the water surface, the reported locations of features can be variable. To be conservative, we track the history of reported feature locations and avoid all of them.

This is done in the obstacle manager by tracking all reported locations for a given feature, and then defining a convex hull for each feature as shown in Fig. 10.

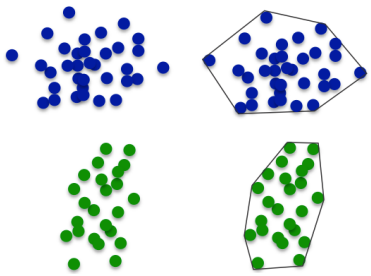

Fig. 10 Conversion of Feature Locations to Convex Hull: As new points (features) arrive, the convex hull is incrementally updated.

The obstacle manager reports the convex hulls as polygons to the IvP Helm. The IvP Helm is configured with an obstacle avoidance behavior template that will spawn a new behavior with each new obstacle ID that is received and subsequent updates from the obstacle manager may change the shape of the polygon representing the obstacle. In Fig. 11, the vehicle is transiting through an obstacle field in a qualification run where four of the obstacles are "active" (generating objective functions) and they are shown in the figure as filled in polygons. An additional buffer is added around each obstacle but if necessary this buffer is shrunk for the vehicle to be able to fit through tight spaces. The collective objective function (Fig. 11-right) is the sum of the waypoint behavior and the four active obstacle avoidance behaviors. In the figure, colors closer to red are higher utility and closer to blue are worse. The angles on the circle denote desired headings (in the same reference frame as the picture on the right) and distance from the center of the circle denotes desired speed. The pink dot in the figure is the outputted desired heading and speed.

5.2 Observation

For observations required in Tasks 2, 3, and 4, we developed a control mode that bypasses the IvP Helm and directly maintains a certain observation point within the field of view of the sensor. In this mode, the desired heading is generated by comparing the actual robot pose (observed through GPS and compass sensors) and the heading required to maintain the observation point in the field of view (Fig. 12). This value is computed in the "Pose Keeping" block (Fig. 13).

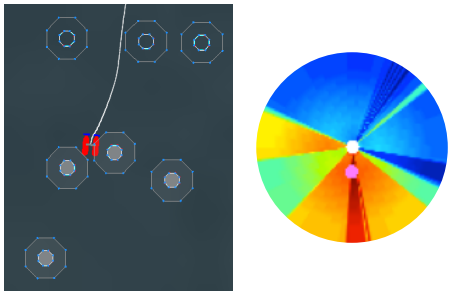

Fig. 11 Left: The ASV navigating through a field of obstacles. The filled polygons are currently active and generating objecting functions. **Right**: A color plot showing the sum of all objective functions where redder colors are higher utility and bluer colors are lower utility. Vehicle not to scale.

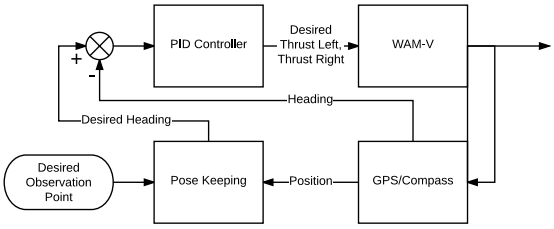

Fig. 12 Control system for maintaining observation of a fixed point. The pose keeping block is used to generate the reference heading and error is minimized through feedback PID control loop.

6 Hardware Setup

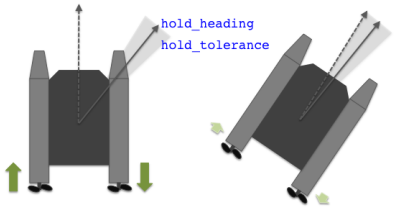

Fig. 13 Pose Keeping: A vehicle with differential thrust applies opposing thrust of equivalent magnitude to turn a vehicle in place until it achieves a desired `hold_heading`, with a given `hold_tolerance`.

The platform base of the vehicle provided to the team was the WAM-V [7], which is a 13-foot, double-pontooned hull with a dynamic suspension system that supports a platform for the vehicle's sensors and electronic components above. The custom-designed power and propulsion system consisted of two Torqeedo Power 26-104 batteries, which rested at the back of the pontoons and powered two Riptide Transom 80 saltwater transom mount trolling motors. The vehicle was steered using a differential drive paradigm through a Roboteq VDC2450 motor controller. The batteries had enough capacity to last all day, and the motors provided enough thrust to be practical and proved easy for folding and stowage.

The vehicle had four computers on board: two Portwell NANO-6060's and two Intel NUC kits, which were configured to be used interchangeably. These computers provided the processing power to process the sensor data from the laser, run the autonomy system, and also communicate with a shoreside computer through a WiFi antenna. The system received location and heading data from a Vector V102 GPS system, and the acoustic data was processed on a PC-104 stack, both of which talked directly to the four main computers. This system was powered by a lead-acid battery, separate from the propulsion power system.

An emergency stop system was designed to sit between the vehicle's computers and the motor controller.

The emergency stop system can communicate directly to an operator control unit (OCU) box, which allowed a human operator to override the autonomy system at any point in time with manual control. Arduino microcontrollers using Xbee radios communicating over a 2.4GHz signal were used. In addition, another layer of safety was designed by tying a pair of on-board emergency stop buttons directly into the motor controller. The whole emergency stop system had its own separate power source, for an added level of safety. An overview of all of the hardware components and connections is shown in Fig. 14.

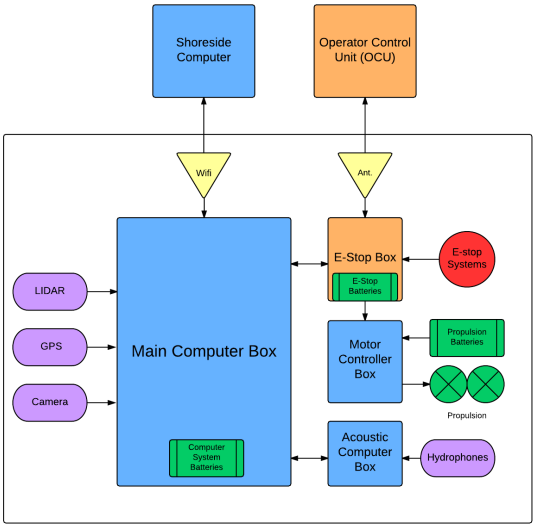

Fig. 14 RobotX hardware system layout and connections.

7 Results and Discussion

A snapshot of the vehicle performing each task is shown in the left column of Fig. 15[2]. On the right column is a task-specific snapshot built from the data collected. For Task 1 (top), the figure shows the navigation through the buoys. We were able to reliably achieve this task throughout. For Task 2 (second row) we show the output of the particle filter as well as the last bearing generated. Row three shows the docking task. Our feature detection based on the method in Sec. 3 was reliable. The fourth row shows the light buoy sequence detection. This was perhaps the most challenging task since it involved color detection in variable light conditions. Additionally, the colorful background enhanced the probability of false detection. The final task was obstacle avoidance. The feature detection and tracking system was reliable, but the overall system had some latency issues as described below.

7.1 What Went Wrong - Lessons Learned

We were able to successfully complete all the tasks successfully in qualifications. However, a few mishaps prevented us from completing each task on the final run. Due to time constraints, we reduced the amount of time that we would wait for the acoustic system to process data, and therefore only received two bearing measurements. This gave us partial points for identifying the color of the closest buoy but not the exact pinger location. On the docking task, we were able to correctly identify the "CIRCLE" placard which was designated at the start of the run, but our right pontoon caught the edge of the dock. This was likely due to incorrect extrinsic calibration of our camera system. At the last second before the final run, we decided to add functionality such that if the light buoy sequence was not

[2] A video of our qualification and final has been made available (`http://robotx.mit.edu/fsr_video`).

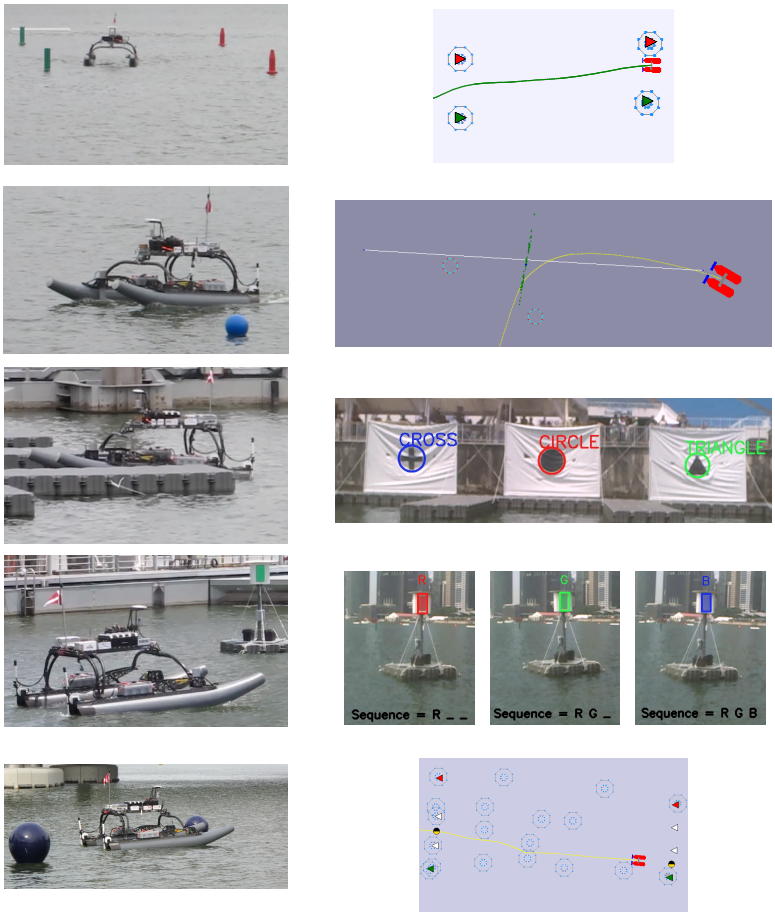

Fig. 15 Left column: snapshots of the WAM-V robot performing each of the five tasks (Task 1 at top to Task 5 at bottom). Right column: Row one, two and five snapshots from the pMarineViewer [2]. Row three and four show processed snapshots from the camera onboard.

determined before a timeout was reached, then we would move on to the final task and at least take a guess.

Unfortunately, we inputted the incorrect task number and this forced a guess to be reported when we *entered* Task 4 (the light buoy observation task), so as a result a guess was reported after docking even though post-processing of the camera data determined that we would have reported a correct sequence. This last-minute change deviated from our typically methodical approach to simulating and testing all code changes prior to deployment and really reinforced that if there is insufficient time to test a modification before deployment then it simply should not be made. We also struck a buoy in the obstacle field. It was later determined that this was due to a delay in our obstacle managing system. Although the buoy had been correctly detected, the behavior necessary to avoid it was not spawned in time to avoid collision.

Table 1 RobotX Final Rankings

1	MIT/Olin (USA)
2	KAIST (South Korea)
3	Queensland University of Technology (Australia)
4	Embry-Riddle Aeronautical University (USA)
5	National University of Singapore (Singapore)
6	Osaka University (Japan)

Despite these errors, we accumulated the highest point total in the final round. The final rankings are shown in Table 1.

8 Conclusion

This paper outlines the MIT/Olin team's approach and performance in the inaugural AUVSI RobotX competition. In the competition, each of the fifteen teams were provided with an identical marine vehicle frame and were responsible for building the propulsion, electronic, sensing, and autonomy systems required to complete a series of five tasks. Ultimately, the MIT/Olin team narrowly won first place in a very competitive field. The team's codebase and data are publicly available.

Acknowledgments

The RobotX competition was supported by AUVSI and ONR. Autonomy development and autonomy test facilities used for RobotX at MIT were supported in part by Battelle Maritime Systems and the ONR, Code 311. The project was also supported by the following grants ONR N00014-13-1-0588 and N00014-14-1-0214, APS-14-12 SLIN 0001, DARPA contract No. N66001-14-C-4031, UCSD award no 43019208, and NSF Award IIS-1318392. We would also like to thank our other sponsors that contributed to the project: MIT Sea Grant, Oliver and Merideth Seikel, Chick and Sandy Corrado, CENSAM, Singapore-MIT Alliance for Research and Technology (SMART), The US Military Academy, and Portwell, Inc. In addition we would like to acknowledge the contributions of the following individuals: Jonathan Garcia-Mallen, Chrys Chyssostomidis, Devynn Diggins, Mindy Tieu, Shivali Chandra, Nikolay Lapin, Justin Poh, Paul Titchener, Alex Kessler, Madelin Perry, Jay Woo, Zoher Ghadyali, William Warner, Victoria Coleman, and Pat Marion.

References

1. Bar-Shalom, Y.: Tracking and Data Association. Academic Press Professional, Inc., San Diego, CA, USA (1987)
2. Benjamin, M., Schmidt, H., Leonard, J.J.: `http://www.moos-ivp.org` (2014)
3. Benjamin, M.R.: Interval Programming: A Multi-Objective Optimization Model for Autonomous Vehicle Control. Ph.D. thesis, Brown University, Providence, RI (2002)
4. Benjamin, M.R., Schmidt, H., Newman, P.M., Leonard, J.J.: Nested Autonomy for Unmanned Marine Vehicles with MOOS-IvP. Journal of Field Robotics **27**(6), 834–875 (2010)
5. Hartley, R.I., Zisserman, A.: Multiple View Geometry in Computer Vision, second edn. Cambridge University Press, ISBN: 0521540518 (2004)
6. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. International Journal of Computer Vision **60**(2), 91–110 (2004)
7. Marine Advanced Research: `http://www.wam-v.com`
8. Maskell, S., Gordon, N.: A Tutorial on Particle Filters for On-line Nonlinear/Non-Gaussian Bayesian Tracking (2001)
9. Neumann, L., Matas, J.: Real-time scene text localization and recognition. In: Proc. IEEE Int. Conf. Computer Vision and Pattern Recognition (CVPR) (2012)
10. Oppenheim, A., ronald Schafer: Discrete-Time Signal Processing, 3rd edn. Prentice Hall (2009)
11. Paull, L., Saeedi, S., Seto, M., Li, H.: AUV navigation and localization: A review. Oceanic Engineering, IEEE Journal of **39**(1), 131–149 (2014)
12. Rosten, E., Drummond, T.: Machine learning for high-speed corner detection. In: Eur. Conf. on Computer Vision (ECCV), pp. 430–443. Springer (2006)
13. Sallen, R.P., Key, E.: A practical method of designing rc active filters. IRE Transactions on Circuit Theory **2**(1), 74–85 (1955)
14. Trees, H.V.: Optimal Array Processing. John Wiley & Sons (2002)