Robotica (2014) volume 32, pp. 291–304. © Cambridge University Press 2014
doi:10.1017/S0263574713001306

AUV behavior recognition using behavior histograms, HMMs, and CRFs

Michael Novitzky†*, Charles Pippin†, Thomas R. Collins‡, Tucker R. Balch† and Michael E. West§

†*College of Computing, The Georgia Institute of Technology, Atlanta, GA, USA*
‡*Electrical Engineering, The Georgia Institute of Technology, Atlanta, GA, USA*
§*Georgia Tech Research Institute, Atlanta, GA, USA*

(Accepted December 19, 2013. First published online: February 10, 2014)

SUMMARY
This paper focuses on behavior recognition in an underwater application as a substitute for communicating through acoustic transmissions, which can be unreliable. The importance of this work is that sensor information regarding other agents can be leveraged to perform behavior recognition, which is activity recognition of robots performing specific programmed behaviors, and task-assignment. This work illustrates the use of Behavior Histograms, Hidden Markov Models (HMMs), and Conditional Random Fields (CRFs) to perform behavior recognition. We present challenges associated with using each behavior recognition technique along with results on individually selected test trajectories, from simulated and real sonar data, and real-time recognition through a simulated mission.

KEYWORDS: Autonomous underwater vehicles; Multi-robot systems; Behavior recognition; Auction-based methods.

1. Introduction

Multi-robot teams, in comparison with single robot solutions, can offer solutions that are more economical, robust to failure, and more efficient than single robot solutions.[1,2] A team of robots can work on tasks in parallel, perform distributed sensing and operate in multiple locations at once. Furthermore, multiple robots add redundancy to the system. Unfortunately, a tradeoff is that these teams must communicate and work together with the added uncertainty regarding the behaviors of robots. For instance, a team member may have trouble cooperating due to communication errors, because they are busy performing other tasks, or have conflicting goals.[3] Many different methods for performing distributed cooperation exist, including centralized optimization algorithms and game theoretic techniques. A centralized method requires at least one agent or a home base to make task or role assignments. Although this may be optimal when communication links are reliable, its efficacy degenerates with intermittent communication, and a central organizer makes the whole system come to a halt if it fails. Thus, a decentralized approach is much more viable because it is more robust to failures of communication. Auction-based algorithms generally have low communication requirements (where agents coordinate tasks through bid messages). Therefore, they can be well suited to environments with communication constraints. Auctions can perform computations in parallel and the methods take advantage of the local information known to each agent.[4,5]

However, this method can still degrade in overall efficiency as communication deteriorates.[6] Poor communication environments are encountered by autonomous underwater vehicles (AUVs) using traditional acoustic modems for underwater communication. The effectiveness of acoustic communications deteriorates in the presence of surface reflections, bottom reflections, ambient noise, and noise sources within the water column, such as emissions from other vessels. Sotzing and Lane[7] have demonstrated that using teammate prediction improves overall performance of a cooperative

* Corresponding author. E-mail: misha@gatech.edu

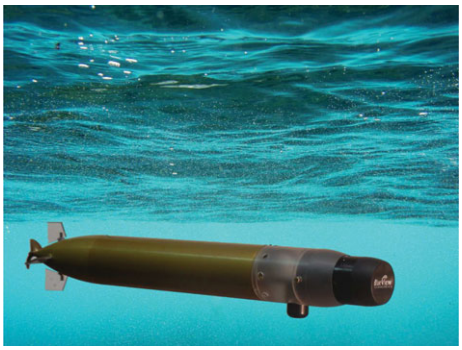


Fig. 1. (Colour online) Yellowfin Autonomous Underwater Vehicle – designed to be man-portable for oceanographic observation.

AUV system. This type of system still needs communication to be sufficient to avoid degradation as task predictions accumulate error over time without correction from teammates.

The motivation for this work is the need for multiple small AUVs to perform autonomous research operations in underwater environments. The Georgia Tech Research Institute has developed the Yellowfin AUV research platform, as seen in Fig. 1, and is planning future experiments with multiple Yellowfin platforms.[8] Because of the vehicle's size, power constraints, and operating environment, communication bandwidth is limited. The vehicle was fabricated with open standards and hardware when possible and uses open-source software components.

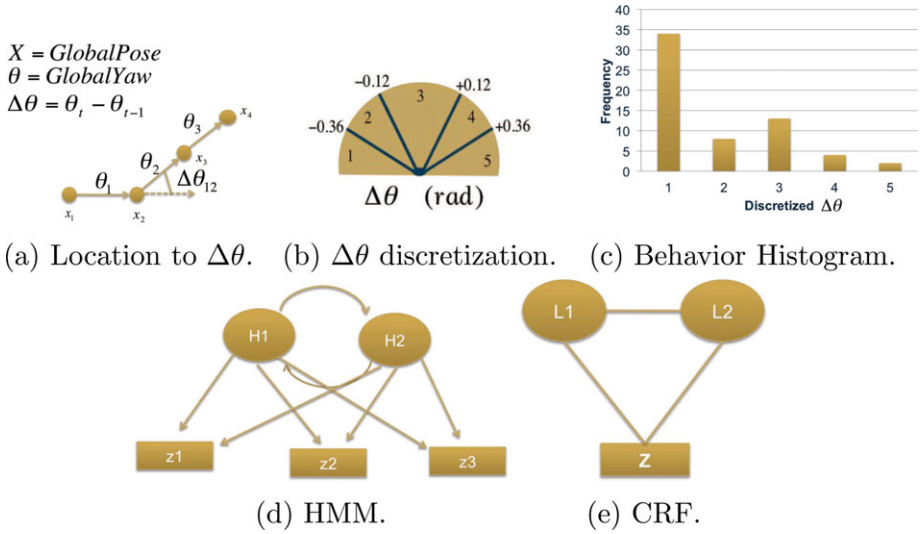
The ultimate purpose of this research is to create a system that can efficiently operate with as little explicit communication as possible, since this is the type of environment our own AUV will encounter.[8] We envision a system similar to that proposed by Novitzky[9] that will utilize auction-based methods augmented with prediction of teammate tasks during periods of poor communication. If the confidence in a prediction of a teammate's task is low, then an AUV can perform prediction verification, through behavior recognition, as suggested by Baxter *et al.*[10] Additionally, this handles the situation where an AUV may have been assigned a task only to discover another agent already performing the task but not communicating. The work in this paper specifically focuses on whether behavior recognition of an AUV is possible using Behavior Histograms, Hidden Markov Models (HMMs), or Conditional Random Fields (CRFs) and which is more suitable for the system described above.

2. Related Work

This work focuses on behavior recognition of autonomous mobile robots, specifically in the underwater domain. Baxter *et al.*[10] performed behavior recognition using HMMs on post-mission analysis of self-localization provided by an AUV. The post-mission analysis converted GPS pose trajectories to low-level actions such as *track-west* and *left u-turn east*. The main drawback of this method is that it claimed to be agnostic to the environment yet still required the use of cardinal direction, which in itself is still somewhat constraining to the compass orientation within an environment. The authors improved upon their discretization methods where they also enhanced HMMs to deal with behavior sequences of variable length.[11] They began with AUV location information from simulated sonar data. These trajectories were fed into a maneuver-recognition algorithm capable of identifying an AUV's actions such as *straight* and *veer-left*, thus making it more environmentally agnostic. While the algorithm was attempting to recognize top-level goals such as *mine-countermeasure* (MCM), *mine-countermeasure inspection* (MCMI), and *point inspection* (PI), it further divided the top-level goals into recognizable sub-goals which included *dive*, *track*, *right u-turn*, and *left u-turn* along with *inspection*. Their results also included the observation that top-level goals are achieved via the AUV performing sub-goal behaviors.

More recently, Novitzky *et al.*[12] performed exploratory work using HMMs to discriminate a small number of robot behaviors. The authors successfully applied their technique to a very limited amount of real data. Their data consisted of collected trajectories gathered while unmanned aerial vehicles (UAVs) performed search and track behaviors of surface targets and AUV trajectories collected via a forward-looking sonar. However, their data consisted of only a few behaviors and at most a handful of tracks for each behavior.

https://doi.org/10.1017/S0263574713001306 Published online by Cambridge University Press



(a) Location to $\Delta\theta$. (b) $\Delta\theta$ discretization. (c) Behavior Histogram.

(d) HMM. (e) CRF.

Fig. 2. (Colour online) In (a) an AUV's location over time is used to determine its global yaw. The change in global yaw from one time step to the next is encoded as an integer value, which represents a given range, as seen in (b). The three behavior-recognition methods are Behavior Histograms, Hidden Markov Models, and a Conditional Random Field, as seen in (c), (d), and (e), respectively.

Of specific importance to this work is that performed by Vail *et al.*[13] in which the authors compared the accuracy of CRFs and HMMs for activity recognition on robot systems. Their chosen domain was simulated robot *Tag*. In their simulation, two robots were passively moving from waypoint to waypoint while a third was the *Seeker* searching for a robot to *Tag*. As part of the analysis of CRFs and HMMs, the authors tested the accuracy with different observations such as raw positions only, including velocities, and chasing features. The authors also examined the effect of incorporating features, which violate the independence assumptions between observations. The results showed that a discriminatively trained CRF performed as well as or better than an HMM in their robot *Tag* domain.

Vail and Veloso[14] used CRFs for analyzing roles in multi-robot domains. The authors experimented with two approaches to feature selection and applied these methods to data recorded during RoboCup soccer small-size league games. The goal of their work was to create a classifier that can provide useful information to robots that are playing against a team whose roles are being classified. They found that using feature selection can dramatically reduce the number of features required by CRFs to achieve error rates that are close to or identical to the error rate achieved by the model with its full complement of features. Reducing the number of features dramatically speeds up online classification and training.

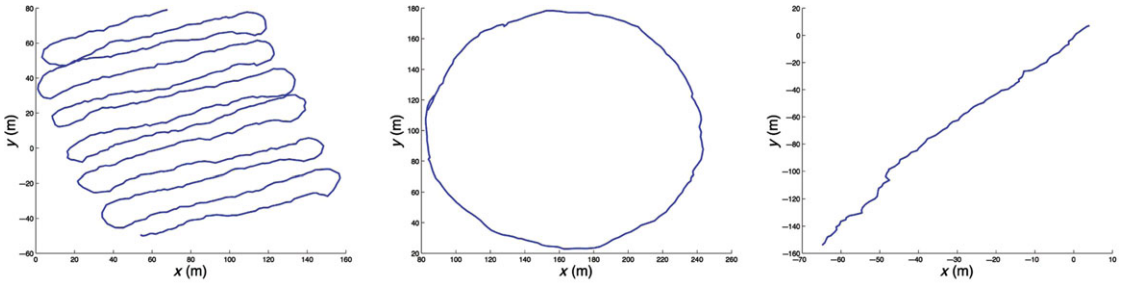
Unlike previous behavior recognition work in the AUV domain, our work does not rely on trajectories provided through post-mission analysis nor only through simulation. Furthermore, it performs behavior recognition of an AUV through the use of a simple discretization method, resulting in only one feature, on both simulated trajectories and actual sonar data comparing the results of a method using Behavior Histograms, HMMs, and a CRF.

While this work uses high-frequency sonar data to validate the ability to perform behavior recognition of autonomous underwater vehicles, it is feasible to use other sensor modalities. Airborne LIDAR has been used for bathymetry, underwater mine detection from helicopters, and shallow underwater target detection from surface vessels.[15–17] Thus, it is feasible that Airborne LIDAR could be used by a heterogenous teammate in the form of an autonomous aerial platform to detect the location of an AUV. Another source of teammate localization data underwater is through the use of hydrophone arrays that have been capable of localizing targets and AUVs.[18] Thus, a teammate capable of interfacing with a hydrophone array can perform behavior recognition from a remote location.

3. Trajectory Discretization

The encoding method used is agnostic to any environment. The only measurement required is the location $\mathbf{x} = (x, y)$ coordinates of an AUV in a fixed 2D plane, as seen in Fig. 2a. The motion model

https://doi.org/10.1017/S0263574713001306 Published online by Cambridge University Press



(a) Noisy *SearchPattern*. (b) Noisy *Loiter*. (c) Noisy *GoToWaypoint*.

Fig. 3. (Colour online) Noisy versions of template trajectories are depicted in (a), (b), and (c).

of the AUV is assumed to be non-holonomic and always moving with a forward motion similar to a tricycle model. The yaw of the AUV is calculated from the vector of motion from one time-step to the next.

$$\Delta \mathbf{x}_{(t-1,t)} = \mathbf{x}_t - \mathbf{x}_{t-1} \tag{1}$$

$$\theta_t = \arctan(\Delta \mathbf{x}_{(t-1,t)}) \tag{2}$$

$$\Delta\theta_t = \theta_t - \theta_{t-1} \tag{3}$$

The encoding used in this research is the change in yaw between time steps. Possible changes in yaw are discretized according to bins. Each bin corresponds to a range of values. Bin 3, for example, represents a change in yaw between -0.12 and $+0.12$ rad. As seen in Fig. 2b, an AUV moving straight ahead is observed as having no change in yaw and thus encoded as a 3 while one turning by 0.26 rad is encoded as a 2. A series of these encodings are combined into a trajectory string for input into the Behavior Histograms, Hidden Markov Models (HMMs), or the Conditional Random Field (CRF).

4. Recognition Methods

This paper focuses on behavior recognition of an AUV performing the behaviors *GoToWaypoint*, *Loiter*, and *SearchPattern*, which can be seen in Fig. 3. These behaviors are created using simple rules integrated with sensor information. In normal operation, these behaviors could be coupled with others such as *AvoidObstacle* which may use cameras or sonar to detect such obstacles. The *GoToWaypoint* is the simplest of the three behaviors as the AUV proceeds from its current location to a defined waypoint. The *Loiter* behavior is performed while waiting for instructions and is created by the AUV following a set of waypoints placed along the perimeter of an octagon. The *SearchPattern* behavior is typically performed while trying to cover a rectangular area by following waypoints placed along evenly separated segmenting lines. In this paper, we assume the behaviors are performed atomically, meaning no other behaviors are running in parallel. In general, observations are labeled as $Z = \{z_1, ..., z_T\}$ where the index represents successive time steps. In our domain, z_t contains an integer value of the change in yaw of the AUV, described above. In the HMM method, each hidden state H may not have an explicit definition. In the CRF method, the labels L_t are drawn from one of the three behaviors. Behavior recognition is the ability for each method to determine one of the three behaviors as the most likely being observed. Accuracy is defined as the behavior recognition method properly identifying the behavior being observed.

4.1. Histogram matching

The baseline behavior recognition method uses histogram matching. After a given trajectory is discretized, as described previously, a histogram is created from the possible encodings. As an example, if a trajectory is discretized into five possible changes in yaw then a histogram of that

https://doi.org/10.1017/S0263574713001306 Published online by Cambridge University Press

(a) VideoRay ROV. (b) Inverted Sonar Image.

Fig. 4. (Colour online) The VideoRay ROV is seen in (a). An inverted sonar image of the VideoRay ROV along with false positive noise is seen in (b). The largest object is assumed to be the target ROV, while the smaller objects are noise.

trajectory will have five bins, as seen in Fig. 2c. The height of each bin will reflect the frequency of that change in yaw throughout the trajectory. Each histogram is normalized.

4.1.1. Template histogram. A template histogram is made for each behavior that we wish to recognize. A training set of trajectory instances are discretized and a normalized histogram is made for each trajectory. A template histogram is then created by taking the mean of all the training instances for each bin.

4.1.2. Recognition using histogram intersection. Histogram intersection is used to determine the match value for each behavior template trajectory histogram B_j to a given behavior trajectory instance histogram B_k:

$$Intersection : B_j, B_k = \sum_{i=1}^{n} min(B_j(i), B_k(i)) \tag{4}$$

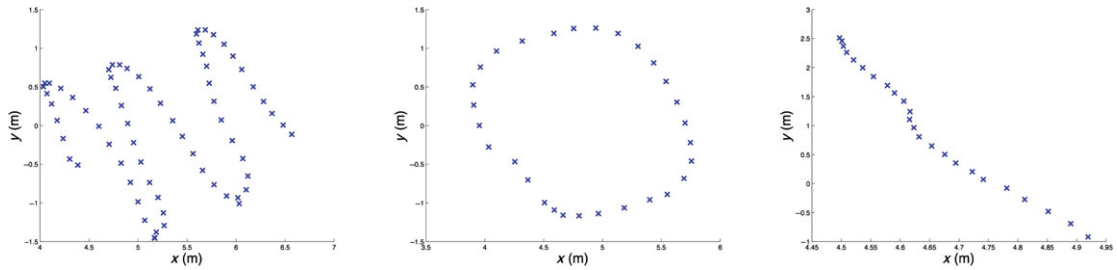
where each histogram has n total bins. A score of 1 means both histograms are an exact match, while a score of 0.5 is a partial match, and a score of 0 is a total mismatch. Behavior recognition is performed by taking the intersection of a behavior trajectory instance with each behavior template histogram. The given trajectory instance is classified as the behavior template histogram resulting with the highest histogram intersection.

4.2. Hidden markov model

In this approach to the recognition problem, each behavior is modeled using a separate Hidden Markov Model (HMM). Each HMM is first trained on example trajectories of a specific behavior. The trained HMM is then given test trajectories to determine the log-likelihood that the test trajectory was generated by that behavior.

4.2.1. Training. The Hidden Markov Model (HMM), as seen in Fig. 2d, is composed of hidden states and observations.[19] In Fig. 2d, the hidden states are labeled with $H_1 ... H_n$ while the observations are labeled $z_1 ... z_t$. A random process can be in any one of the hidden states and can emit any one of the observations. In this work the observations consist of the labeled changes in yaw, $\Delta\theta$. The number of hidden states for each HMM are empirically determined. An HMM must learn the transition probabilities between hidden states, $a_{i,j} = P(h_{t+1} = j | h_t = i)$, the probabilities that a hidden state may produce an observation, $b_{j,k} = P(z_t = k | h_t = j)$, and the initial state distribution, $\pi_j = P(h_1 = j)$. The compact notation $\lambda = (A, B, \pi)$ represents the complete parameter set of an HMM. The Baum-Welch algorithm estimates the maximum likelihood of the parameters, λ, when given a corpus of training data, Z, $\bar{\lambda} = max_\lambda P(Z|\lambda)$.

https://doi.org/10.1017/S0263574713001306 Published online by Cambridge University Press



(a) Sonar *SearchPattern*. (b) Sonar *Loiter*. (c) Sonar *GoToWaypoint*.

Fig. 5. (Colour online) Real trajectories captured by a BlueView forward-looking sonar of an AUV performing *SearchPattern*, *Loiter*, and *GoToWaypoint* are seen in (a), (b), and (c), respectively.

4.2.2. Testing. The probability of an observation sequence, Z, given an HMM trained on a behavior, λ,

$$P(Z|\lambda) = \sum_{all\,H} P(Z|H, \lambda)P(Z|\lambda) \tag{5}$$

is efficiently produced by the forward algorithm.[19] This produces a log-likelihood that a test trajectory instance was produced by the behavior it was trained upon.[19] A trial consists of an instance of a behavior trajectory being tested against each possible HMM. At each trial, the HMM producing the maximum log-likelihood is determined as the representative behavior of the trial. If the representative behavior matches the true test instance label, then it is logged as a positive identification. The accuracy of each trained HMM is the number of positive identifications over the entire corpus of similarly labeled instances.

4.3. Conditional random field

As seen in Fig. 2e, conditional random fields (CRFs) are undirected graphical models and are commonly used for structured classification.[20] CRFs are built from a vector of weights and a vector of features. Features take the form $f_i(t, l_{t-1}, l_t, Z)$ where i is an index into the feature vector f and t is an offset into the sequence, l_{t-1} and l_t are values of the label pair at time $t-1$ and t, respectively. Z represents the entire observation sequence across all values of t. In both the following subsections,

$$\Phi_Z = \sum_L \prod_{t=1}^{T} exp(w^T f(t, l_{t-1}, l_t, Z)) \tag{6}$$

is a normalizing constant.

4.3.1. Training. Training of CRFs is performed by finding a weight vector w^* that maximizes the conditional log-likelihood of labeled training data:

$$\mathcal{L}(L|Z; w) = w^T f(t, l_{t-1}, l_t, Z) - log(\Phi_Z) \tag{7}$$

$$w^* = \arg\max_w \mathcal{L}(L|Z; w) \tag{8}$$

4.3.2. Testing. The conditional probability of a label sequence given an observation sequence is computed from the weighted sum of the features as:

$$P(L|Z) = \frac{1}{\Phi_Z} \prod_{t=1}^{T} exp(w^T f(t, l_{t-1}, l_t, Z)) \tag{9}$$

https://doi.org/10.1017/S0263574713001306 Published online by Cambridge University Press

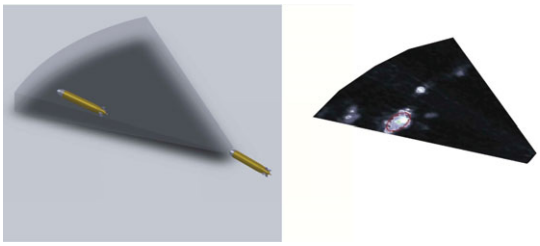


Fig. 6. (Colour online) An AUV using its forward-looking sonar to *Track & Trail* a leader AUV in order to perform behavior recognition.

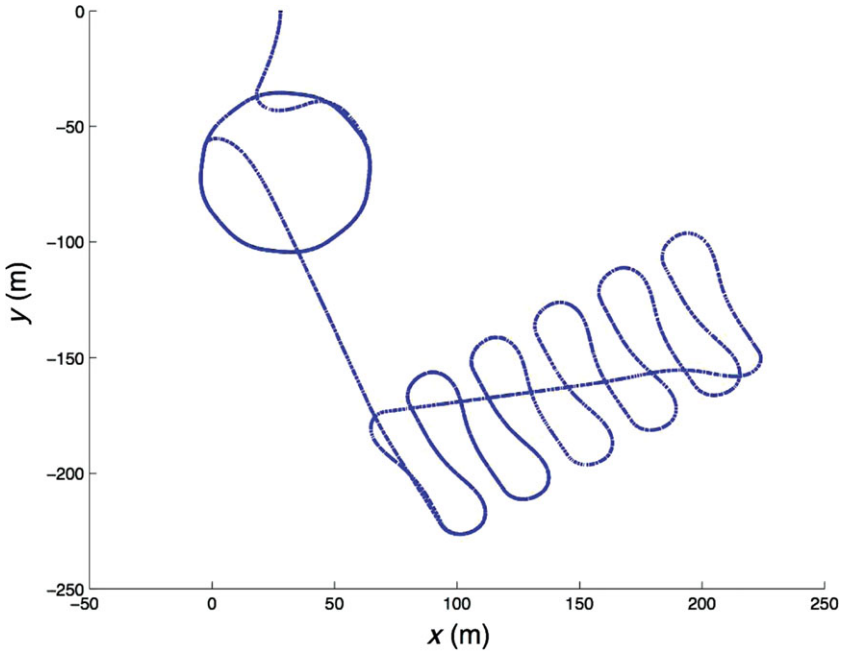


Fig. 7. (Colour online) Simulated survey mission in which the AUV loiters, proceeds to a location, and then performs *SearchPattern* of a specific area. The three behavior-recognition techniques label the behaviors in real-time.

The most likely behavior label l is assigned to each observation for each test instance presented to the trained CRF.

5. Experiments

We performed experiments in two scenarios: a stationary observer and *Tracking & Trailing* observer that followed the vehicle of interest. The stationary experiments were first performed using trajectory data gathered through simulation and then using a stationary forward-looking sonar. In order to test our method with two AUVs, trajectory data was gathered in simulation with one *Tracking & Trailing* a leader vehicle. The final set of experiments was performed with the three behavior recognition techniques receiving data in real time from a simulated mission.

5.1. Stationary observer

5.1.1. Simulation. MOOS-IvP, a widely used behavioral architecture for real and simulated AUVs, is used to generate the simulated trajectory data.[21] The behaviors *GoToWaypoint*, *Loiter*, and *SearchPattern* are run within iMarineSim and viewed through pMarineViewer, which are tools included in MOOS-IvP. The locations of the AUVs are recorded as each behavior is executed, providing a template trajectory. In order to create more realistic results, the template trajectories undergo rotation and translation transformations and are injected with Gaussian noise. Variations of

https://doi.org/10.1017/S0263574713001306 Published online by Cambridge University Press

Table I. Accuracy of simulated stationary behavior recognition.

Behavior	Training	Testing	Histogram	HMM	CRF
SearchPattern	600	400	84.25%	100.00%	100.00%
Loiter	600	400	92.25%	99.25%	100.00%
GoToWaypoint	600	400	65.00%	96.00%	100.00%

Table II. Confusion matrices: simulated stationary data.

		Recognized behavior		
	Actual Behavior	*SearchPattern*	*Loiter*	*GoToWaypoint*
Histogram	*SearchPattern*	337	0	63
	Loiter	31	369	0
	GoToWaypoint	121	19	260
HMM	*SearchPattern*	400	0	0
	Loiter	0	397	3
	GoToWaypoint	16	0	384
CRF	*SearchPattern*	400	0	0
	Loiter	0	400	0
	GoToWaypoint	0	0	400

each behavior trajectory template are created as they undergo a random assignment of transformations, including changes in rotation and translation along with an injection of cumulative Gaussian noise with random assignments of standard deviation ranging from 0 to 0.25, as seen in 4a, 4b, and 4c. This will demonstrate that our methods are agnostic to the environment as they are robust to rotations and translations and environmental noise. The global change in yaw for these experiments is discretized into seven bins with a spread of four degrees per bin.

5.1.2. Real sonar data. For our real sonar data experiments, a surrogate vehicle called the VideoRay ROV is used instead of the Yellowfin AUV due to space limitations in our testing tank, the Acoustic Water Tank facility of the Georgia Tech Woodruff School of Mechanical Engineering. The testing tank is 7.62 meters deep, 7.62 meters wide, and 10.36 meters long. The VideoRay is a modified Pro 4 model ROV from Video Ray LLC. that includes the addition of Yellowfin subsystems such as the WHOI acoustic micro-modem and a BlueView forward looking sonar, as seen in Fig. 4a. The experiments were conducted with a BlueView forward-looking sonar positioned statically in a corner while it recorded the location of a human piloted VideoRay ROV. Throughout the experiment, the VideoRay ranged between 1 and 10 meters from the BlueView sonar. For these experiments, the perception algorithm makes the simplifying assumptions that there is only one relevant object in the scene, the VideoRay, and that it will always be in the FOV of the sonar. The VideoRay operators were asked to perform multiple runs of three behaviors, *GoToWaypoint*, *Loiter*, and *SearchPattern*.

The BlueView forward-looking sonar provides an image with intensity values corresponding to the acoustic response of a surface, as seen in Fig. 4b. The more intense a pixel, the more likely that an object exists at that location. In order to smooth the ROV's trajectory, only every ith frame is used, here $i = 10$. This reduces the number of outliers significantly as the sonar data is extremely noisy. Edges are found in the sonar image, which are used to create contours. The contour with the largest area is assumed to be the ROV, as we assume that only the ROV is in the image and the smaller contours are noise. The API of the BlueView sonar then produces the range and bearing of the center pixel relative to the sonar itself. Range and bearing are then converted to x and y coordinates to produce trajectories, as seen in Fig. 5. In this form, the discretization process converts location to global yaw then to change in yaw as described above. In this experiment, the global change in yaw is discretized into five bins with a spread of four degrees each.

https://doi.org/10.1017/S0263574713001306 Published online by Cambridge University Press

Table III. Accuracy of sonar behavior recognition.

Behavior	Training	Testing	Histogram	HMM	CRF
SearchPattern	21	12	75.00%	100.00%	75.00%
Loiter	23	16	68.75%	68.75%	68.75%
GoToWaypoint	14	10	90.00%	100.00%	80.00%

Table IV. Confusion matrices: stationary sonar data.

		Recognized behavior		
	Actual Behavior	*SearchPattern*	*Loiter*	*GoToWaypoint*
Histogram	*SearchPattern*	9	1	2
	Loiter	4	11	1
	GoToWaypoint	1	0	9
HMM	*SearchPattern*	12	0	0
	Loiter	5	11	0
	GoToWaypoint	0	0	10
CRF	*SearchPattern*	9	1	2
	Loiter	5	11	0
	GoToWaypoint	0	2	8

5.2. Track & Trail

In order to test our methods with a non-stationary observer, testing was performed on simulated data with one AUV performing *Track & Trail* of a leader performing a behavior, as seen in Fig. 6. As in the experiments above, the MOOS-IvP simulator is used to generate template trajectories of a leading AUV performing *GoToWaypoint*, *Loiter*, and *SearchPattern* while an observing AUV performs *Track & Trail*.

In order to use the change in yaw method of encoding, the template trajectories of each vehicle are used to produce the pose (x, y, θ) of the trailing vehicle along with range and bearing to the lead vehicle. Using this information allows the trailing AUV to reconstruct the leading AUV's trajectory, which will be discretized for use in behavior recognition. In order to create more realistic results, the original measurements of the trailing AUV's location (x, y, θ) along with range and bearing to the leader AUV are injected with Gaussian noise, similar to those seen in 4a, 4b, and 4c. This more accurately represents the uncertainty an AUV will have of its own location and the uncertainty of the location of the target AUV present in sonar data. In this experiment, the global change in yaw is discretized into seven bins with a spread of four degrees each.

5.3. Real-time recognition

In order to test our methods performing recognition in real-time, an example survey mission was simulated in MOOS-IvP, as seen in Fig. 7. The mission consisted of an AUV performing a repeated survey of an area. First, the AUV performs the *Loiter* behavior as it awaits the start mission order. In transit to the survey location, the AUV performs the *GoToWaypoint* behavior. Once it arrives at the appropriate coordinates, the AUV begins the *SearchPattern* behavior of the area. At the termination of the *SearchPattern* behavior, the AUV performs *GoToWaypoint* behavior to return to the start of the survey coordinates to perform a second *SearchPattern* behavior.

Each behavior recognition method had slight modifications for real-time recognition. The CRF produces a histogram of each behavior ID and if a single behavior was above a threshold of 70% then it was labeled as that behavior. If no behavior was above 70% then the CRF system returned *no label*. The HMM method produces a negative log-likelihood that a learned behavior HMM produced the trajectory found in the behavior window. The larger the negative log-likelihood, closer to zero, the more likely that behavior produced the trajectory seen in the window. The HMM method was sensitive to the number of hidden states used to represent a behavior. A manual tuning process resulted in all the HMMs having the same number of states. The behavior histogram method produced an

https://doi.org/10.1017/S0263574713001306 Published online by Cambridge University Press

Table V. Accuracy of simulated *Track & Trail* behavior recognition.

Behavior	Training	Testing	Histogram	HMM	CRF
SearchPattern	600	400	97.50%	97.25%	99.50%
Loiter	600	400	96.25%	94.75%	99.75%
GoToWaypoint	600	400	56.25%	95.25%	99.75%

Table VI. Confusion matrices: simulated *Track & Trail* data.

		Recognized behavior		
	Actual Behavior	*SearchPattern*	*Loiter*	*GoToWaypoint*
Histogram	*SearchPattern*	390	0	10
	Loiter	8	385	7
	GoToWaypoint	116	59	225
HMM	*SearchPattern*	389	0	11
	Loiter	1	379	20
	GoToWaypoint	13	6	381
CRF	*SearchPattern*	398	2	0
	Loiter	1	399	0
	GoToWaypoint	0	1	399

intersection value for each trained behavior. A value of one meant that the behavior in the window was a perfect match for the behavior histogram and anything less meant it was less like the trained behavior histogram. In the simulated survey mission each behavior was labeled by hand, including an extra label for transitions between behaviors. The results for the real-time behavior recognition experiments are the percentage of correctly labeled behaviors versus those incorrectly labeled.

6. Results

The results of three different sources of data are analyzed and presented below. The accuracy of the Behavior Histogram, Hidden Markov Model (HMM), and Conditional Random Field (CRF) methods are considered for each data source. Accuracy is defined as the behavior recognition method correctly identifying the behavior with its true label.

6.1. Stationary observer

6.1.1. Simulation. As seen in Table I, each method was trained on a specific behavior using a corpus of 600 instances of trajectories generated by running the behavior offline. A total of 400 trajectories from each behavior, for a total of 1200 instances, were presented to the three methods for classification. The trajectories were generated by inserting Gaussian noise with a standard deviation of 0.25 on the location. As is seen in Table I, the Behavior Histogram method was able to achieve an accuracy of 84.25%, 99.25%, and 65% for *SearchPattern*, *Loiter*, and *GoToWaypoint*, respectively. The HMMs performed well for *SearchPattern*, *Loiter*, and *GoToWaypoint* as they were able to accurately discriminate trials by 100%, 99.25%, and 96%, respectively. The CRF had 100% accuracy on all three behaviors. As seen in Table II, the confusion matrix for the Behavior Histogram shows that 63 instances of *SearchPattern* were confused with *GoToWaypoint*, while *GoToWaypoint* was confused with *SearchPattern* and *Loiter*. For the three behaviors, the HMM method had the most difficulty in discriminating *GoToWaypoint*, recognizing 16 instances of that behavior as *SearchPattern*.

6.1.2. Sonar data. As seen in Table III, each method was trained on real sonar data while an ROV performed a specific behavior using a corpus of 21 instances for *SearchPattern*, 23 instances for *Loiter*, and 14 instances for *GoToWaypoint*. A total of 38 instances were presented to each method for testing. The HMM discrimination method had the best accuracy of 100%, 68.75%, and 100%, respectively. The Behavior Histogram method had an accuracy of 75%, 68.75%, and 90%. The

https://doi.org/10.1017/S0263574713001306 Published online by Cambridge University Press

Table VII. Percentage confusion matrices: real-time simulated survey mission (Window Size 40, HMM with 7 states).

		Recognized behavior			
	Actual Behavior	*SearchPattern*	*Loiter*	*GoToWaypoint*	*NoLabel*
Histogram	*SearchPattern*	91.49	0.00	8.51	0.00
	Loiter	8.16	90.31	0.00	1.53
	GoToWaypoint	8.74	26.21	65.05	0.00
	Transition	61.46	0.00	0.00	38.54
HMM	*SearchPattern*	95.82	0.00	0.00	4.18
	Loiter	21.63	75.46	0.00	2.91
	GoToWaypoint	52.43	2.52	0.00	45.05
	Transition	58.33	0.00	0.00	41.67
CRF	*SearchPattern*	93.14	0.36	5.21	1.29
	Loiter	4.80	86.79	0.00	8.42
	GoToWaypoint	14.47	24.85	50.97	9.71
	Transition	53.33	4.90	0.00	41.77

Table VIII. Percentage confusion matrices: real-time simulated survey mission (Window Size 55, HMM 7 states).

		Recognized behavior			
	Actual Behavior	*SearchPattern*	*Loiter*	*GoToWaypoint*	*NoLabel*
Histogram	*SearchPattern*	86.08	0.00	13.92	0.00
	Loiter	5.10	85.71	0.00	9.18
	GoToWaypoint	15.53	35.92	48.54	0.00
	Transition	61.46	0.00	0.00	38.54
HMM	*SearchPattern*	94.33	0.00	0.00	5.67
	Loiter	22.55	66.94	0.00	10.51
	GoToWaypoint	42.14	1.75	0.00	56.12
	Transition	55.52	0.00	0.00	44.48
CRF	*SearchPattern*	84.90	0.00	10.88	4.23
	Loiter	4.80	78.47	1.33	15.41
	GoToWaypoint	16.89	24.56	41.94	16.60
	Transition	47.29	0.00	2.92	49.79

CRF performed worse than both the other methods with recognition of *SearchPattern*, *Loiter*, and *GoToWaypoint* with accuracy of 75%, 68.75%, and 80%, respectively. As seen in Table IV, the Behavior Histogram method mis-identified one instance of *SearchPattern* with *Loiter* and two as *GoToWaypoint*. Its worst performance was misidentifying four instances of *Loiter* as *SearchPattern* and one instance as *GoToWaypoint*. The HMM method only had errors in recognizing *Loiter* with five instances being identified as *SearchPattern*. The CRF method performed similarly to the HMM method in misinterpreting *Loiter* as *SearchPattern*. Additionally, the CRF method identified one instance of *SearchPattern* as *Loiter* and two instances as *GoToWaypoint*. The CRF method's best performance on the real sonar data was in recognizing *GoToWaypoint* as it only mis-identified two instances as *Loiter*.

https://doi.org/10.1017/S0263574713001306 Published online by Cambridge University Press

Table IX. Percentage confusion matrix: real-time simulated survey mission (HMM 7 states variable window sizes 70,70,25).

	Recognized behavior			
Actual	*SearchPattern*	*Loiter*	*GoToWaypoint*	*NoLabel*
SearchPattern	92.73	0.00	0.08	7.19
Loiter	20.87	60.97	0.00	18.16
GoToWaypoint	31.07	0.87	32.91	35.15
Transition	58.54	0.00	0.00	41.46

Table X. Percentage confusion matrix: real-time simulated survey mission (HMM 7 states variable window sizes 70,70,10).

	Recognized behavior			
Actual	*SearchPattern*	*Loiter*	*GoToWaypoint*	*NoLabel*
SearchPattern	90.75	0.00	4.51	4.74
Loiter	20.87	60.97	0.00	18.16
GoToWaypoint	28.93	0.87	49.71	20.49
Transition	56.35	0.00	2.19	41.46

6.2. Track & Trail

As seen in Table V, using the change in yaw of the leading vehicle as a discretization method resulted in sufficient accuracy as most of the accuracy was above 90%. The results are from inserting Gaussian noise with a standard deviation of 0.75 on the location (x, y, θ) of the trailing vehicle, range, and bearing to the leader. The Behavior Histogram method had an accuracy of 97.5%, 96.25%, and 56.25% of *SearchPattern*, *Loiter*, and *GoToWaypoint*, respectively. The HMM discrimination method had accuracy of 97.25% with *SearchPattern*, 94.75% with *Loiter*, and 95.25% with *GoToWaypoint*. The CRF discrimination method had the best accuracy of discrimination of *SearchPattern*, *Loiter*, and *GoToWaypoint* with accuracy of 99.50%, 99.75%, and 99.75%, respectively. As seen in Table VI, the CRF method only had at the worst case two mis-recognitions of *SearchPattern* versus the HMM method which had a best case of only 11 mis-recognitions and the Behavior Histogram method had 175 mis-recognitions of *GoToWaypoint*.

6.3. Real-time recognition

As seen in Tables VII–X, the three behavior recognition methods were able to recognize the behaviors in real-time during the simulated survey mission with varying success. The discretization of the change in yaw for each run was performed every two seconds with nine bins each representing eight degrees of change in yaw. Experiments were performed with varying behavior window sizes. In Table VII, all three behavior recognition techniques were tested with a behavior window size of 40 observations. The HMM method fared the worst of the three methods as it was unable to recognize the *GoToWaypoint* behavior while also only recognizing 75.46% of the *Loiter* behavior. The CRF performed better as it was able to identify 93.14%, 86.79%, and 50.97% of *SearchPattern*, *Loiter*, and *GoToWaypoint* behaviors, respectively. The behavior histogram method fared the best overall as it was above 90.0% accuracy for both the *SearchPattern* and *Loiter* behaviors but only had a 65.05% accuracy for the *GoToWaypoint* behavior. Increasing the behavior window size to 55 observations for all three methods decreased their performance across the board, as seen in Table VIII.

The HMM method proved to be fragile in relation to the behavior window size. Several experiments were performed with varying behavior window sizes for each behavior HMM. As seen in Table IX, by adjusting the behavior window size of the trajectory presented to the *SearchPattern*, *Loiter*, *GoToWaypoint* HMMs to 70, 70, and 25, respectively, increased the accuracy of the HMM method in

https://doi.org/10.1017/S0263574713001306 Published online by Cambridge University Press

recognizing *GoToWaypoint* from 0.0% to 32.91%. The accuracy in determining the *GoToWaypoint* behavior increased to 49.71% by decreasing the behavior window size even more to 10 observations, as seen in Table X. There was a slight decrease in the accuracy of the HMM method in recognizing both *SearchPattern* and *Loiter* in the 70,70,25 and 70,70,10 window size conditions compared to 40 and 55 window sizes that is a reflection of the window size and interplay between the negative log-likelihood produced by each behavior HMM.

7. Conclusion

The work presented here demonstrates the ability of the three behavior recognition systems to work relatively well when static testing data is provided. The Hidden Markov Models (HMM) method resulted in sufficient performance with static data as most of the recognition accuracy was above 90%. Using Behavior Histograms resulted in the worst performance of the three methods in both the simulated stationary data and the simulated *Track & Trail* data while performing slightly better than the CRF method in the sonar data. It is not surprising that the baseline method, Behavior Histogram, is the worst performer when compared to the other two methods as it only takes the frequency of the changes in yaw into account making it less suitable for time series data. Using the Conditional Random Field method resulted in better performance than the HMM method when there was ample training data available, as in the simulated stationary observer data or the simulated *Track & Trail* data. However, the CRF method performed poorly in discrimination of the real sonar data. Due to the small sample size of real sonar data it may be an indication of under-training the CRF. However, all the methods struggled with discriminating the *Loiter* behavior in the real sonar data set. It is possible that the sonar-captured *Loiter* behavior should be further separated into *counter-clockwise* and *clockwise Loiter* as that could be the reason for the methods performing poorly. This is in contrast to the simulated *Loiter* trajectories, which only performed them in the *counter-clockwise* direction.

The three behavior recognition techniques did not fare as well during real-time recognition experiments as an ideal system would have above 90% accuracy for all the behaviors. In general, the behavior histogram method performed the best. This may be due to the lack of noise in the simulated mission trajectory. Both the CRF and HMM methods poorly recognized the *GoToWaypoint* behavior. When using one behavior window size for all behaviors, the CRF method performed with more accuracy overall than the HMM method. In general, the HMM method is more difficult to tune because the number of hidden states must be determined in conjunction with the behavior window size for each behavior for best performance. The poor overall results for all the behavior recognition methods may be due to the method in which the simulated mission trajectory is labeled. The real-time data was labeled according to which behavior the vehicle was running at the time of discretization. This does not correspond to what a human can distinguish by looking at the behavior window. This will be a focus of our future research.

Discretization parameters for the change in yaw observations play a crucial role in the success of behavior recognition in these experiments. For example, an experiment that discretizes the global change in yaw with five bins each with a spread of four degrees has a limited resolution. Any change in yaw greater than six or less than negative six degrees is placed into bins one and five, respectively. Thus, if a crucial distinction between two behaviors occurs beyond these terminal edge bins they will not be properly discriminated. The discretization parameters were empirically determined for each experiment. While simple, global changes of yaw by time as an observation method is very susceptible to changes in speed. Alternatives are to use an observation of changes in yaw by distance or describe higher level primitives such as *straight*, *left turn*, and *right turn* as observations for our behavior recognition methods.

Future work includes further investigation of discretization and behavior recognition methods along with a larger data set. It may be possible that each behavior recognition method requires different change in yaw discrimination parameters for improved accuracy. Alternatively, higher level motion primitives may increase accuracy and robustness. Alternative recognition methods may be more appropriate. Handwriting recognition has similarities with behavior recognition with a long history and may yield improved methods.[22] Recognition should be verified with more behaviors than the ones used in these experiments, as they are a small sample representation. Future research will include adding noise to the simulated real-time missions and a real-time mission with data gathered from actual vehicles.

https://doi.org/10.1017/S0263574713001306 Published online by Cambridge University Press

Acknowledgments

The authors thank the reviewers with their useful comments and edits. The authors thank their colleagues Paul Robinette and Andrew Melim along with the faculty and staff of the Georgia Tech Mechanical Engineering Acoustic Tank for aiding in data gathering. The authors thank Aaron Bobick, Magnus Egerstedt, Brian Hrolenok, and Jim Rehg for their valuable comments. This work was conducted through internal research and development funding by the Georgia Tech Research Institute (GTRI).

References

1. G. Dudek, M. Jenkin, E. Milios and D. Wilkes, "A taxonomy for multi-agent robotics," *Auton. Robots* **3**(4), 375–397 (1996).
2. Y. Cao, A. Fukunaga and A. Kahng, "Cooperative mobile robotics: antecedents and directions," *Auton. Robots* **4**(1), 7–27 (1997).
3. R. C. Arkin, *Behavior-Based Robotics*, ch. 9. (MIT Press, Cambridge, MA, 1998).
4. M. Dias and A. Stentz, "A free market architecture for distributed control of a multirobot system," *Proceedings of the 6th International Conference on Intelligent Autonomous Systems (IAS-6)* (Jul. 2000) pp. 115–122.
5. B. Gerkey and M. Mataric, "Sold!: Auction methods for multirobot coordination," *IEEE Trans. Robot. Autom.* **18**(5), 758–768 (2002).
6. S. Sariel, T. Balch and N. Erdogan, "Naval mine countermeasure missions," *IEEE Robot. Autom. Mag.* **15**(1), 45–52 (2008).
7. C. Sotzing and D. Lane, "Improving the coordination efficiency of limited-communication multi-autonomus underwater vehicle operations using a multiagent architecture," *J. Field Robot.* **27**(4), 412–429 (2010).
8. M. West, M. Novitzky, J. Varnell, A. Melim, E. Sequin, T. Toler, T. Collins and J. Bogle, "Design and Development of the Yellow fin uuv for Homogenous Collaborative Missions," *Association for Unmanned Vehicle Systems International* (2010).
9. M. Novitzky, "Improvement of Multi-AUV Cooperation Through Teammate Verification," *Automated Action Planning for Autonomous Mobile Robots: Papers from the 2011 AAAI Workshop (WS-11-09)*, San Francisco, CA (2011) pp. 72–73.
10. R. Baxter, D. Lane and Y. Petillot, "Behaviour Recognition for Spatially Unconstrained Unmanned Vehicles," *Proceedings of the IJCAI* **9**, 1–8 (2009).
11. R. H. Baxter, D. M. Lane and Y. Petillot, "Recognising Agent Behaviour During Variable Length Activities," *Proceedings of The 19th European Conference on Artificial Intelligence* (2010).
12. M. Novitzky, C. Pippin, T. Balch, T. Collins and E. West, "Behavior Recognition of an AUV Using a Forward-Looking Sonar," **In**: *Workshops at the Robotics: Science and Systems*, (Los Angeles, CA, 2011).
13. D. Vail, M. Veloso and J. Lafferty, "Conditional Random Fields for Activity Recognition," *Proceedings of the 6th International Joint Conference on Autonomous Agents and Multiagent Systems*, ACM (2007) pp. 1–8.
14. D. Vail and M. Veloso, "Feature Selection for Activity Recognition in Multi-Robot Domains," *Proceedings of AAAI,* Pittsburgh (2008).
15. J. Irish and J. Lillycrop, "Scanning laser mapping of the coastal zone: the shoals system," *ISPRS J. Photogramm. Remote Sens.* (1999).
16. "Mine warfare platforms, programs and systems," *Naval Expeditionary Warfare Directorate (N85): Naval expeditionary warfare vision 2010* (2010).
17. M. Tulldahl and M. Pettersson, "Lidar for Shallow Underwater Target Detection," *Proceedings of the SPIE, Electro-Optical Remote Sensing, Detection, and Photonic Technologies and Their Applications* (2007).
18. J. Dzielski, M. DeLorme, A. Sedunov, P. Sammut and M. Tsionskiy, "Guidance of an Unmanned Unederwater Vehicle Using a Passive Acoustic Threat Detection System," *IEEE Waterside Security Conference (WSS)* (2010).
19. L. Rabiner, "A tutorial on hidden Markov models and selected applications in speech recognition," *Proceedings of the IEEE* **77**(2), 257–286 (1989).
20. J. Lafferty, A. McCallum and F. Pereira, "Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data," *International Conference on Machine Learning* (2001).
21. M. Benjamin, H. Schmidt, P. Newman and J. Leonard, "Nested autonomy for unmanned marine vehicles with MOOS-IvP," *J. Field Robot.* **27**(6), 834–875 (2010).
22. R. Plamondon and S. N. Srihari, "Online and off-line handwriting recognition: a comprehensive survey," *IEEE Trans. Pattern Anal. Mach. Intell.* **22**(1), 63–84 (2000).