Behavior Recognition of Autonomous Underwater Vehicles Using Forward-Looking Sonar

Michael Novitzky, Charles Pippin, Thomas R. Collins, Tucker R. Balch and Michael E. West

Abstract Multi-robot teams can cooperate to perform multiple tasks simultaneously and have the potential to be more efficient and robust to failure when compared to single robot solutions. Current decentralized task assignment approaches, such as market based methods, are dependent on agent communication. However, in low communication environments, such as in underwater applications, effective cooperation can become more difficult and current methods for task assignment degrade. One key to more effective interaction is the ability to understand the behavior and intent of other robots. Observed teammate action sequences can be learned to perform behavior recognition and task-assignment in the absence of communication. This paper focuses on multi-robot teams working cooperatively in an underwater application. We apply Hidden Markov Models (HMMs) to perform behavior recognition as an approach to task monitoring in the absence of robust communication in a challenging underwater environment. In order to demonstrate the feasibility of performing behavior recognition of an Autonomous Underwater Vehicle (AUV), we use trajectory based techniques for model generation and behavior discrimination in experiments using simulated and real sonar data. Experimental results are presented and compared to a histogram based technique.

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 so-

Michael Novitzky and Tucker R. Balch
College of Computing, Georgia Institute of Technology, 801 Atlantic Drive, Atlanta, Georgia 30332 e-mail: misha@gatech.eduandtucker@cc.gatech.edu

Charles Pippin, Thomas R. Collins, and Michael E. West
Georgia Tech Research Institute, 430 N. 10th St. Atlanta, GA 30332 e-mail: first.last@gtri.gatech.edu

lutions [8, 6]. 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 because they have conflicting goals [2].

Many different methods for performing distributed cooperation exist, including centralized optimization algorithms and game theoretic techniques. Centralized methods can provide optimal solutions, but are less effective in poor communication environments due to a central point of failure. Decentralized methods have varying reliance on communication schemes, and can often handle intermittent communications. For instance, auction-based algorithms generally have low communication requirements (agents assign tasks using bids). Therefore, they are well suited to environments with communication constraints [7, 10]. However, this method can still degrade in overall efficiency as communication deteriorates [18]. In order to mitigate the effects of intermittent communication, Sotzing and Lane [19] demonstrated that using teammate task prediction improves overall performance of a cooperative autonomous underwater vehicles (AUV) system. This is important for AUVs as acoustic transmissions suffer from surface reflections, bottom reflections, ambient noise, and noise sources within the water column, such as emissions from other vessels. However, Sotzing and Lane's system still needs communication to be of relatively good quality, as without a sufficient amount of communication the system degrades as predictions accrue error over time without correction from teammate communication.

The purpose of this research is to enable robot teams to cooperate more effectively in the poorest of communication environments. Many current decentralized coordination methods, such as auctions or self-assignment, require teammates to broadcast their self-assigned task/roles along with costs. By defining a task/role as a robot performing a behavior in a certain location, behavior recognition can be used as task/role identification. This research focuses on teammate behavior recognition to incorporate observations and indirect communication into a task assignment framework.

2 Background and Related Work

There are several distinctions that are made in order to categorically study the recognition of teammate behaviors [20], including plan, activity, goal, and intent recognition. While researchers use some of these terms interchangeably, it is the focus of this research to perform activity recognition, which is the process of recognizing behaviors using only sensor data about physical movement of teammates.

There are several approaches to recognition including logic-based and probabilistic-based methods. Early logic-based work in tracking individual activity used con-

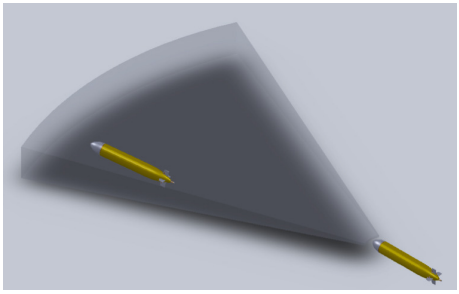


Fig. 1: An AUV using its forward-looking sonar to *track & trail* a leader AUV. The follower AUV can use sensor observations to perform behavior recognition of the leader AUV.

straint satisfaction techniques [21] in the *RESC* (REal-time Situated Commitments) system. This method was used for successful activity recognition of unmanned aerial vehicles (UAVs) in simulation. More recent work by Roduit et al. [17] used Point Distribution Models (PDMs) to compare robot trajectories. Hazan et al. [12] classify robot behaviors using topological methods rather than metric ones. They created fingerprints of robot behavior by describing the inner structure of the dynamics using topological objects such as unstable periodic orbits.

In particular, this paper uses Hidden Markov Models (HMM), as they are useful for pattern recognition and have proven robust to uncertainty in many areas including speech recognition [16] and handwritten script recognition [13]. The most prevalent use of behavior recognition with robots has been through the use of an overhead sensor, such as a camera.

Original work was presented by Han and Veloso [11] in which an agent's behavior was recognized using a Hidden Markov Model (HMM). The authors discretized input from a vision system overseeing a Robocup small-sized league robot interacting with a ball. The vision system reported the location and velocity of both the agent and the ball. These features were handed off to several concurrent HMMs looking for *go-to-ball* and *move-behind-ball* behaviors. Subsequent work by Feldman and Balch [9] used a similar technique with HMMs to recognize the behavior of honey bees. The vision algorithm tracked the honey bee paths which were given to a k-Nearest Neighbors (KNN) algorithm to determine one of several actions such as *waggle*. The normalized scores from the kNN were then fed into an HMM to determine which behavior was being displayed.

This work focuses on behavior recognition of autonomous mobile robots, specifically in the underwater domain similar to that of Baxter et al. [3, 4]. Baxter et al. [3] 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 is constrained to the compass

orientation within an environment. The authors improved upon their discretization methods in [4] where they also enhanced HMMs to deal with behaviors of variable length. They began with simulated AUV trajectories which were fed into maneuver recognition algorithm capable of identifying an AUV's actions of *straight*, *veer-left*, *veer-right*, and *dive* thus making it more environmentally agnostic. While the authors were searching for top-level goals such as *mine-countermeasure* (MCM), *mine-countermeasure inspection* (MCMI), and *point inspection* (PI), they further divided the top level goals into sub-goals which included *dive*, *track*, *right u-turn*, and *left u-turn* along with *inspection*. Baxter et al. indicate that top-level goals are achieved via the AUV performing several behaviors and were ultimately concerned with detecting high-level goals of variable length. The work in this paper uses a much simpler and environmentally agnostic encoding method and uses actual sonar data for behavior classification.

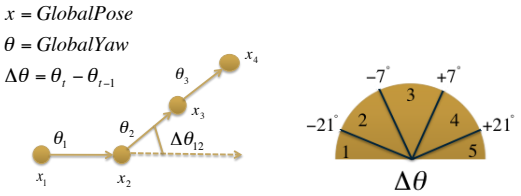
More recent work, [15] uses HMMs to discriminate a small number of robot behaviors. The authors successfully applied their technique to a very limited amount of real data. Their data consisted of collected trajectories gathered while unmanned aerial vehicles (UAVs) performed search and track behaviors of surface targets and UUV trajectories collected via a forward-looking sonar. However, the data consisted of only a few behaviors and at most a handful of tracks for each behavior.

The ultimate goal of this work is to create a system that can efficiently operate with as little explicit communication as possible. We envision a system similar to that proposed by [14] which will utilize auction-based methods along 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 in [3]. Additionally, this approach addresses the situation where an AUV may have been assigned a task only to discover another agent already performing the task but not communicating.

This paper investigates the feasibility of an AUV performing behavior recognition of another AUV through a forward-looking sonar, as illustrated in Fig. 1. Although [4] has used simulated sonar data and [3] has used post-mission GPS trajectory analysis of an actual AUV for behavior verification, little or no research has attempted to perform behavior recognition using an actual forward-looking sonar. Our method is validated both in simulation and using actual data from a BlueView forward-looking sonar.

3 Methods

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



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

Fig. 2: In (a) an AUV's location over time is used to determine it's 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).

$$\Delta x_{(t-1,t)} = x_t - x_{t-1} \tag{1}$$

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

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

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 in our example represents a change in yaw between -7 and 7 degrees. For instance, as seen in Fig. 2b, an AUV moving straight ahead is observed as having a 0° change in yaw and thus encoded as a *3* while one turning by 45° is encoded as a *2*. A series of these encodings are combined into a trajectory string for input into the Hidden Markov Model (HMM) or the histogram matching methods.

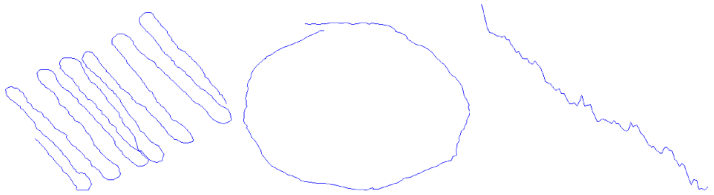
3.2 Histogram Matching: The baseline behavior recognition method uses histogram matching. After a given trajectory is discretized, as described previously, a histogram is made of the possible encodings. As an example, if a trajectory is discretized into five possible changes in yaw then a histogram of that trajectory will have five bins. The height of each bin will reflect the frequency of that change in yaw throughout the trajectory. Each histogram is normalized.

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

3.2.2 Recognition using Histogram Intersection: Histogram intersection is used to determine the match value for each behavior template trajectory histogram H_j to a given behavior trajectory instance histogram H_k :

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

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



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

Fig. 3: Simulated noisy trajectories of an AUV performing *SearchPattern*, *Loiter*, and *GoToWaypoint* are seen in (a), (b),and (c).

trajectory instance with each behavior template histogram. The given trajectory instance is classified as the behavior template histogram resulting with the highest histogram intersection.

3.3 Hidden Markov Models: In this research 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.

3.3.1 Training: The Hidden Markov Model (HMM) is composed of hidden states and observable states [16]. The hidden states are labeled with $x_1 ... x_n$ while the observation states are labeled $y_1 ... y_n$. A random process can be in any one of the hidden states and can emit any one of the observable states. In this work the observable states consist of the labeled changes in yaw, $\Delta\theta$. The number of hidden states are empirically determined. An HMM must learn the transition probabilities between hidden states and the probabilities that a hidden state may produce an observation. The Baum-Welch algorithm estimates the maximum likelihood of the parameters when given a corpus of training data.

3.3.2 Testing: Testing an HMM trained on a behavior is produced by the forward algorithm. An HMM can be used to determine the negative log-likelihood that a test trajectory instance was produced by the behavior it was trained upon [16]. A trial consists of an instance of a behavior trajectory being tested against each possible HMM. At each trial the HMM producing the maximum negative 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 Experimental Setup

The simulation environment for our experiments is provided by iMarineSim and pMarineViewer which are part of the MOOS-IvP open source autonomy package



(a) YellowRay ROV (b) Template *Loiter*

Fig. 4: The YellowRay ROV is seen in (a). An inverted sonar image is seen in (b) of the YellowRay ROV along with false positive noise. The largest object is assumed to be the target AUV.

[5]. The pMarineViewer module is a GUI-based tool that renders 2D overhead maps of the vehicles performing behaviors. The iMarineSim is a single-vehicle simulator that updates vehicle state based on actuator values. Actuator values are produced by the IvP Helm, which is a coordinator over activated behaviors. IvP Helm behaviors used extensively in this work include *waypoint following*, *obstacle avoidance*, and *Track&Trail*. The simulator tools allow for verification of vehicle behaviors and interactions.

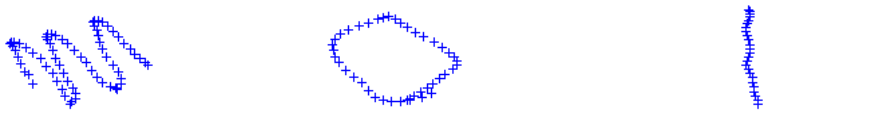
4.1 Stationary Observer: The experiments were first performed using trajectory data gathered through simulation and then using a stationary forward-looking sonar. In order to test our method with two Yellowfin vehicles, trajectory data was gathered in simulation with one *tracking & trailing* a leader vehicle.

4.1.1 Simulation: MOOS-IvP is used to generate the simulated trajectory data. The behaviors *GoToWaypoint*, *Loiter*, and *SearchPattern* are run within iMarineSim and viewed through pMarineViewer. 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 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, as seen in 3a, 3b, and 3c. This will demonstrate that our methods are agnostic to the environment as they are robust to rotations and translations and environmental noise.

4.1.2 Real Sonar Data: For our real sonar data experiments the YellowRay ROV is used, shown in FIG. 4a,in our testing tank[1].

The testing tank is 7.62 meters deep, 7.62 meters wide, and 10.36 meters long. The YellowRay is a Video Ray ROV [1]. This includes the addition of subsystems

[1] Acoustic Water Tank Facility , Woodruff School of Mechanical Engineering, Georgia Tech



(a) Sonar *Search-Pattern* (b) Sonar *Loiter* (c) Sonar *GoToWaypoint*

Fig. 5: 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.

such as the WHOI acoustic micro-modem, a BlueView forward looking sonar, and buoyancy to counteract our modifications, 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 YellowRay ROV. Throughout the experiment the Yellowray ranged between 1 to 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 YellowRay, and that it will always be in the FOV of the sonar. The YellowRay 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. Image processing techniques are used to find the center of the largest blob in the sonar image, as we assume that only the ROV is in the image. The API of the BlueView sonar then produces the range and bearing of the center pixel relative to the sonar itself. Range and bearing is then converted to x and y coordinates to produce trajectories, as seen in Fig. 5. In this form, discretization can take place, converting location to global yaw then to change in yaw as described above.

4.2 *Track&Trail*: In order to determine the ability of deploying our behavior recognition methods on a system consisting of a mobile observer, testing was performed on simulated data with one AUV performing *Track&Trail* of a leader performing a behavior. 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*. Each AUV runs its own MOOSDB community with their own instances of iMarineSim while both are viewed through pMarineViewer.

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, to produce trajectories similar

to those seen in 3a, 3b,and 3c. 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.

5 Results

The accuracy of both the histogram and Hidden Markov Model (HMM) methods are presented for each of the three data sources. Additionally, the confusion matrices for both methods are presented in a single table for each source of data. Each row corresponding to each behavior test instance being recognized as a column behavior.

5.1 Stationary Observer

The simulation results are shown in Table 1, each method was trained on a specific behavior using a corpus of 600 instances of trajectories generated by that behavior. A total of 400 trajectories from each behavior, for a total of 1200 instances, were presented to both methods. As is seen in Table 1, the histogram intersection technique was able to discriminate *SearchPattern*, *Loiter*, and *GoToWaypoint* with 84.25%, 92.25%, and 65% accuracy, respectively. The HMMs performed better for *SearchPattern*, *Loiter*, and *GoToWaypoint* as they were able to accurately discriminate trials by 100%, 99.25%, and 96%, respectively. The histogram method, as seen in Table 2, had poor classification results with 63 instances of *SearchPattern* being identified as *GoToWaypoint*. The histogram method performed poorly when recognizing instances of *GoToWaypoint* with 121 being identified as *SearchPattern* and only 19 identified as *Loiter*.

The results for the experiments using the sonar data are shown in Table 3. Each method was trained on real sonar data while an ROV performed a specific behavior

Table 1: Accuracy of Simulated Stationary Behavior Recognition.

Behavior	Training	Testing	Histogram	HMM
SearchPattern	600	400	84.25 %	100%
Loiter	600	400	92.25 %	99.25%
GoToWaypoint	600	400	65 %	96%

Table 2: Confusion matrices for the simulated stationary data.

		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

Table 3: Accuracy of Sonar Behavior Recognition.

Behavior	Training	Testing	Histogram	HMM
SearchPattern	21	12	75%	100%
Loiter	23	16	68.75 %	68.75%
GoToWaypoint	14	10	90%	100%

Table 4: Confusion matrices for stationary sonar data.

		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

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 both methods for testing. The histogram intersection method was able to discriminate instances of *SearchPattern*, *Loiter*, and *GoToWaypoint* with an accuracy of 75%, 68.75%, and 90%, respectively. The HMM discrimination method had better accuracy of 100%, 68.75% and 100%, respectively. The HMM method only had false positives with five instances of *Loiter* being identified as *SearchPattern*, as seen in Table 4. The histogram method predominantly suffered with four instances of *Loiter* being identified as *SearchPattern* and one as *GoToWaypoint*. *SearchPattern* was also misidentified as *Loiter* for one instance and *GoToWaypoint* for two of them.

5.2 *Track&Trail*

Table 5: Track&Trail Results.

Behavior	Training	Testing	Histogram	HMM
SearchPattern	600	400	97.5%	97.25 %
Loiter	600	400	96.25%	94.75 %
GoToWaypoint	600	400	56.25 %	95.25 %

As seen in Table 5, using the change in yaw of the leading vehicle as a discretization method resulted in sufficient accuracy. 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 histogram intersection method was able to discriminate instances of *SearchPattern*, *Loiter*, and *GoToWaypoint* with an accuracy of 97.5%, 95.25%, and 56.25%, respectively. The HMM discrimination method, however, had equivalent accuracy of 97.25% with *SearchPattern*, 94.75% with *Loiter*, and a much higher accuracy of 95.25% with *GoToWaypoint*. The worst

Table 6: Confusion matrices for the *Track&Trail* data.

		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

performance was by the histogram method with recognition of *GoToWaypoint* with 116 instances being recognized as *SearchPattern* and 59 instances as *Loiter*, as seen in Table 6.

6 Conclusion and Future Work

The work presented here demonstrates the feasibility of performing behavior recognition of an AUV in situ. Using Hidden Markov Models (HMM) resulted in more accurate behavior discrimination when compared to the histogram intersection technique. This is because histograms don't capture the sequence of actions, but only the frequency of each action without regard to the order in which they occurred. What is interesting is the results obtained against the real sonar data, which was not preprocessed or conditioned, both methods struggled in discriminating the *Loiter* behavior. It is possible that the sonar-captured *Loiter* behavior should be further separated into *left* and *right Loiter* as that could be the reason for both methods performing poorly, since they are very similar to *SearchPattern* behavior. This is in contrast to the simulated *Loiter* trajectories which only performed them in the left direction.

Future work includes further investigation of more optimal parameters for both discretization and for the behavior HMMs as accuracy can be improved. Ultimately, these methods should be verified with more behaviors than the ones used in these experiments, as they are a small sample representation. The next step for verifying the ability to recognize behaviors with one AUV tracking & trailing a leader should be performed with real forward-looking sonars on real AUVs.

Acknowledgements 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 their valuable comments and aiding in data gathering. This work was conducted through internal research and development funding by the Georgia Tech Research Institute (GTRI).

References

1. Videoray (2011). URL `http://www.videoray.com/`
2. Arkin, R.C.: Behavior-Based Robotics, chap. 9. Cambridge, Mass., MIT Press (1998)
3. Baxter, R., Lane, D., Petillot, Y.: Behaviour recognition for spatially unconstrained unmanned vehicles. In: Proceedings of the IJCAI, vol. 9, pp. 1–8 (2009)
4. Baxter, R.H., Lane, D.M., Petillot, Y.: Recognising agent behaviour during variable length activities. In: Proceedings of The 19th European Conference on Artificial Intelligence (2010)
5. Benjamin, M., Schmidt, H., Newman, P., Leonard, J.: Nested autonomy for unmanned marine vehicles with moos-ivp. Journal of Field Robotics **27**(6), 834–875 (2010)
6. Cao, Y., Fukunaga, A., Kahng, A.: Cooperative mobile robotics: Antecedents and directions. Autonomous robots **4**(1), 7–27 (1997)
7. Dias, M., Stentz, A.: A free market architecture for distributed control of a multirobot system. In: 6th International Conference on Intelligent Autonomous Systems (IAS-6), pp. 115–122 (2000)
8. Dudek, G., Jenkin, M., Milios, E., Wilkes, D.: A taxonomy for multi-agent robotics. Autonomous Robots **3**(4), 375–397 (1996)
9. Feldman, A., Balch, T.: Modeling honey bee behavior for recognition using human trainable models. In: Modeling Other Agents from Observations (Workshop at AAMAS) (2004)
10. Gerkey, B., Mataric, M.: Sold!: Auction methods for multirobot coordination. Robotics and Automation, IEEE Transactions on **18**(5), 758–768 (2002)
11. Han, K., Veloso, M.: Automated robot behavior recognition. In: ROBOTICS RESEARCH-INTERNATIONAL SYMPOSIUM-, vol. 9, pp. 249–256 (2000)
12. Hazan, A., Davesne, F., Vigneron, V., Maaref, H.: Topological characterization of mobile robot behavior. In: Intelligent Robots and Systems, 2006 IEEE/RSJ International Conference on, pp. 4157–4162. IEEE (2007)
13. Nag, R., Wong, K., Fallside, F.: Script recognition using hidden markov models. In: Acoustics, Speech, and Signal Processing, IEEE International Conference on ICASSP, vol. 11, pp. 2071 – 2074 (1986)
14. Novitzky, M.: Improvement of multi-auv cooperation through teammate verification. In: Workshops at the Twenty-Fifth AAAI Conference on Artificial Intelligence (2011)
15. Novitzky, M., Pippin, C., Balch, T., Collins, T., West, E.: Behavior recognition of an auv using a forward-looking sonar. In: Workshops at the Robotics: Science and Systems. Los Angeles, CA (2011)
16. Rabiner, L.: A tutorial on hidden markov models and selected applications in speech recognition. Proceedings of the IEEE **77**(2), 257–286 (1989)
17. Roduit, P., Martinoli, A., Jacot, J.: A quantitative method for comparing trajectories of mobile robots using point distribution models. In: Intelligent Robots and Systems, 2007. IROS 2007. IEEE/RSJ International Conference on, pp. 2441–2448. IEEE (2007)
18. Sariel, S., Balch, T., Erdogan, N.: Naval mine countermeasure missions. Robotics & Automation Magazine, IEEE **15**(1), 45–52 (2008)
19. Sotzing, C., Lane, D.: Improving the coordination efficiency of limited-communication multi-autonomous underwater vehicle operations using a multiagent architecture. Journal of Field Robotics **27**(4), 412–429 (2010)
20. Sukthankar, G.: Activity recognition for agent teams. Ph.D. thesis, CMU (2007)
21. Tambe, M., Rosenbloom, P.: Resc: An approach for real-time, dynamic agent tracking. In: INTERNATIONAL JOINT CONFERENCE ON ARTIFICIAL INTELLIGENCE, vol. 14, pp. 103–111 (1995)