Conditional Random Fields for Behavior Recognition of Autonomous Underwater Vehicles

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

Abstract This paper focuses on multi-robot teams working cooperatively in an underwater application. Multi-robot teams working cooperatively to perform multiple tasks simultaneously have the potential to be more robust to failure and efficient when compared to single robot solutions. One key to more effective interaction is the ability to identify the behavior of other agents. However, the underwater environment presents specific challenges to teammate behavior identification. Current decentralized collaboration approaches, such as auction-based methods, degrade in poor communication environments. Sensor information regarding teammates can be leveraged to perform behavior recognition and task-assignment in the absence of communication. This work illustrates the use of Conditional Random Fields (CRFs) 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 AUV in the underwater domain, we use trajectory based techniques for model generation and behavior discrimination in experiments using simulated trajectories and real sonar data. Results are presented with comparison of a CRF method to one using Hidden Markov Models.

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.edu and e-mail: tucker@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 [7, 5]. 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 [2]. Many different methods for performing distributed cooperation exist, in-

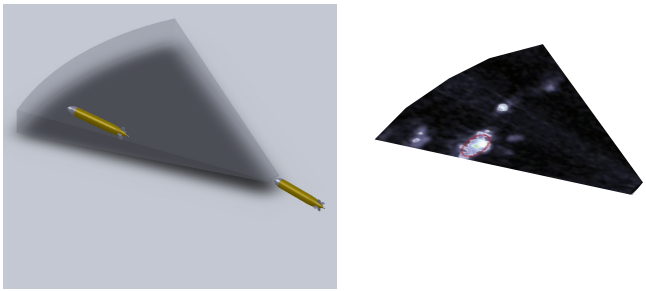


Fig. 1: An AUV using its forward-looking sonar to *Track & Trail* a leader AUV in order to perform behavior recognition.

cluding centralized optimization algorithms and game theoretic techniques. A centralized method requires at least one agent or a home base to make task/role assignments. Although this may be optimal when communication links are reliable, its efficacy degenerates with intermittent communication, and a central point of failure makes the whole system come to a halt. Thus, a decentralized approach is much more viable as 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 are 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 [6, 8].

However, this method can still degrade in overall efficiency as communication deteriorates [13]. Such poor communication environments are encountered by autonomous underwater vehicles (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. Sotzing and Lane [14] have demonstrated that using teammate prediction improves overall performance of a cooperative AUV system. Such a 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 ultimate purpose of this research is to create a system that can efficiently operate with as little explicit communication as possible as this is the type of environment our own AUV will encounter [17]. We envision a system similar to that

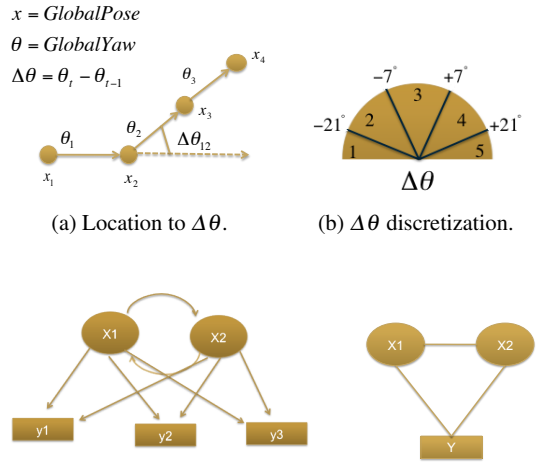
proposed by Novitzky [10] which will utilize auction-based methods along with prediction of teammate tasks during periods of poor communication. If the confidence in a prediction of a teammate's task is low, then an AUV can perform prediction verification through behavior recognition, as suggested in [3]. Additionally, this handles the situation where an AUV may have been assigned a task only to discover another agent already performing the task but not communicating.

2 Related Work

This work focuses on behavior recognition of autonomous mobile robots, specifically in the underwater domain. 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 in itself is still somewhat constraining 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 AUV location information from simulated sonar data. These trajectories were fed into maneuver recognition algorithm capable of identifying an AUV's actions such as *straight* and *veer-left* 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*. Their results also included that top-level goals are achieved via the AUV performing sub-goal behaviors.

More recently, Novitzky et al. [11] 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 UUV 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.

Of specific importance to this work is that performed by Vail et al. [16] 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.



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

(b) $\Delta\theta$ discretization.

(c) Hidden Markov Model (HMM)

(d) Conditional Random Field

Fig. 2: 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 two behavior recognition methods are Hidden Markov Models and Conditional Random Fields, as seen in (c), and (d) respectively.

Vail and Veloso [15] used CRFs for multi-robot domains. The authors experimented with two approaches to feature selection: grafting, and l_1 regularization. They 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, this 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 a CRF and a method using HMMs.

3 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.

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

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

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

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, for example, represents a change in yaw between -7 and 7 degrees. 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 -15° 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.

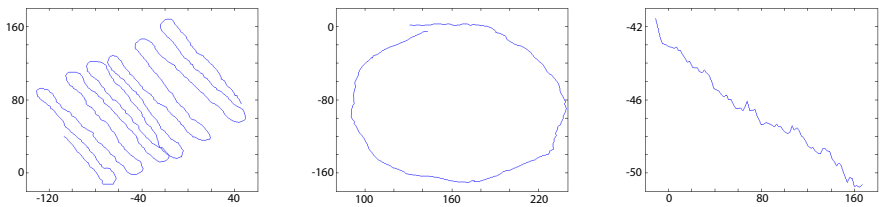
4 Discrimination Methods

In general, observations are labeled as $Y = \{y_1, ..., y_T\}$ and states are labeled as $X = \{x_1, .., x_T\}$, where the index represents successive time steps. In our domain y_t contains an integer value of the change in yaw of the AUV, described above. In the HMM method each hidden state x may not have an explicit definition. In the CRF method the labels x_t are drawn from one of the three behaviors.

4.1 Hidden Markov Model: 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.

4.1.1 Training: The Hidden Markov Model (HMM), as seen in Fig. 2c, is composed of hidden states and observable states [12]. In Fig. 2c 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.

4.1.2 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



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

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

trajectory instance was produced by the behavior it was trained upon [12]. 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.

4.2 Conditional Random Field: As seen in Fig. 2d, Conditional random fields (CRFs) are undirected graphical models for structured classification [9]. CRFs are built from a vector of weights and a vector of features. Features take the form $f_i(t, x_{t-1}, x_t, Y)$ where i is an index into the feature vector f and t is an offset into the sequence, x_{t-1} and x_t are values of the label pair at time $t-1$ and t respectively. Y represents the entire observation sequence across all values of t.

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

$$l(X|Y;w) = w^T f(t, x_{t-1}, x_t, Y) - log(Z_Y) \tag{4}$$

$$w^* = \arg\max_y l(X|Y;w) \tag{5}$$

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

$$P(X|Y) = \frac{1}{Z_Y} \prod_{t=1}^{T} exp(w^T f(t, x_{t-1}, x_t, Y)) \tag{6}$$

$$Z_Y = \sum_{X'} \prod_{t=1}^{T} exp(w^T f(t, x'_{t-1}, x'_t, Y)) \tag{7}$$

The most likely label x is assigned to each test instance presented to the trained CRF.

(a) YellowRay ROV (b) Inverted Sonar Image

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

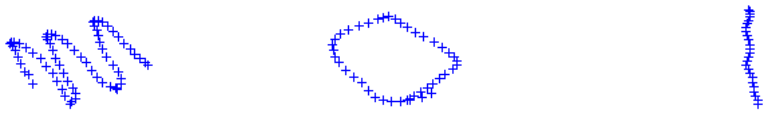
5 Experiments

5.1 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 AUVs, trajectory data was gathered in simulation with one *Tracking & Trailing* a leader vehicle.

5.1.1 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.

5.1.2 Real Sonar Data: For our real sonar data experiments, a surrogate vehicle called the YellowRay ROV is used instead of the Yellowfin AUV due to space limitations 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] which has been modified to act as a viable surrogate of the Yellowfin. This 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 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

[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.

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. The center of the largest blob in the sonar image is found by first calculating the edges. Contours are created using the edges and 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 blobs 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 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.

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 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 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.

6 Results

The results of three different sources of data are analyzed. The accuracy of the Hidden Markov Model (HMM) and Conditional Random Field (CRF) methods are considered for each data source.

6.1 Stationary Observer:

Table 1: Accuracy of Simulated Stationary Behavior Recognition.

Behavior	Training	Testing	HMM	CRF
SearchPattern	600	400	100%	100%
Loiter	600	400	99.25%	100%
GoToWaypoint	600	400	96%	100%

Table 2: Confusion matrix for the HMM method applied to simulated stationary data.

		SearchPattern	*Loiter*	*GoToWaypoint*
HMM	*SearchPattern*	400	0	0
	Loiter	0	367	3
	GoToWaypoint	16	0	384

6.1.1 Simulation: As seen 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 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 performed better for *SearchPattern*, *Loiter*, and *GoToWaypoint* as it was able to discriminate the behaviors with 100% accuracy. The HMM method had the most difficulty in discriminating *GoToWaypoint* as it recognized 16 instances of that behavior as *SearchPattern*.

6.1.2 Sonar Data:

Table 3: Accuracy of Sonar Behavior Recognition.

Behavior	Training	Testing	HMM	CRF
SearchPattern	21	12	100%	75%
Loiter	23	16	68.75%	68.75%
GoToWaypoint	14	10	100%	80%

Table 4: Confusion matrices for the CRF and HMM methods applied to stationary sonar data.

		SearchPattern	*Loiter*	*GoToWaypoint*
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

As seen in Table 3, 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 both methods for testing. The HMM discrimination method had the best accuracy of 100%, 68.75% and 100%, respectively. The CRF performed worse than the HMM method with discrimination of *SearchPattern*, *Loiter*, and *GoToWaypoint* with accuracy of 75%, 68.75%, and 80%, respectively. The HMM method only had false positives with five instances of *Loiter* being identified as *SearchPattern*, as seen in Table 4. The CRF method suffered similarly to the HMM method in discriminating *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 discriminating *GoToWaypoint* as it only mis-identified two instances as *Loiter*.

The CRF method may be susceptible to both the way in which the location of the ROV is calculated and the resolution of the changes in yaw. In order to smooth the location of the ROV only every ith location is used. This reduces the number of outliers significantly as the sonar data is extremely noisy resulting in the ROV location to jump back and forth. Even though the trajectory data is somewhat smoothed by using only every ith location there were still some outliers. Additionally, the CRF may have benefited from more resolution in the change in yaw encoding. However, more experiments must be performed.

6.2 *Track & Trail*: 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 HMM discrimination method had accuracy of 97.25% with *SearchPattern*, 94.75% with *Loiter*, and

Table 5: Accuracy of Simulated *Track & Trail* Behavior Recognition.

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

Table 6: Confusion matrices for CRF and HMM methods applied to the *Track & Trail* data.

		SearchPattern	*Loiter*	*GoToWaypoint*
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

a much higher accuracy of 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 6, the CRF method only had at worst two mis-discriminations of *SearchPattern* versus the HMM method which had a best case of only 11 mis-discriminations.

7 Conclusion

The work presented here demonstrates the feasibility of performing behavior recognition of an AUV in situ. In general, using Hidden Markov Models (HMM) resulted in sufficient performance. 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 discriminating the *Loiter* behavior in the real sonar data set. 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 the 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. Further investigation of noisier simulated data is necessary to determine the failure point of both the Hidden Markov Model and Conditional Random Field methods. In these experiments a true distinction begins to show in the *Track & Trail* data. 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 is to obtain a larger corpus of real data. To truly test the feasibility of behavior recognition of one AUV *Tracking & Trailing* a leader should be performed with real forward-looking sonars on real AUVs while performing recognition in real-time.

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. Cao, Y., Fukunaga, A., Kahng, A.: Cooperative mobile robotics: Antecedents and directions. Autonomous robots **4**(1), 7–27 (1997)
6. 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)
7. Dudek, G., Jenkin, M., Milios, E., Wilkes, D.: A taxonomy for multi-agent robotics. Autonomous Robots **3**(4), 375–397 (1996)
8. Gerkey, B., Mataric, M.: Sold!: Auction methods for multirobot coordination. Robotics and Automation, IEEE Transactions on **18**(5), 758–768 (2002)
9. Lafferty, J., McCallum, A., Pereira, F.: Conditional random fields: Probabilistic models for segmenting and labeling sequence data (2001)
10. Novitzky, M.: Improvement of multi-auv cooperation through teammate verification. In: Workshops at the Twenty-Fifth AAAI Conference on Artificial Intelligence (2011)
11. 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)
12. Rabiner, L.: A tutorial on hidden markov models and selected applications in speech recognition. Proceedings of the IEEE **77**(2), 257–286 (1989)
13. Sariel, S., Balch, T., Erdogan, N.: Naval mine countermeasure missions. Robotics & Automation Magazine, IEEE **15**(1), 45–52 (2008)
14. Sotzing, C., Lane, D.: Improving the coordination efficiency of limited-communication multi-autonomus underwater vehicle operations using a multiagent architecture. Journal of Field Robotics **27**(4), 412–429 (2010)
15. Vail, D., Veloso, M.: Feature selection for activity recognition in multi-robot domains. In: Proceedings of AAAI, vol. 2008 (2008)
16. Vail, D., Veloso, M., Lafferty, J.: Conditional random fields for activity recognition. In: Proceedings of the 6th international joint conference on Autonomous agents and multiagent systems, pp. 1–8. ACM (2007)
17. West, M., Novitzky, M., Varnell, J., Melim, A., Sequin, E., Toler, T., Collins, T., Bogle, J.: Design and development of the yellowfin uuv for homogenous collaborative missions (2010)