Bio-Inspired Multi-Robot Communication through Behavior Recognition

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

***Abstract*— This paper focuses on enabling multi-robot teams to cooperatively perform tasks without the use of radio or acoustic communication. One key to more effective cooperative interaction in a multi-robot team is the ability to understand the behavior and intent of other robots. This is similar to the honey bee "waggle dance" in which a bee can communicate the orientation and distance of a food source. In this similar manner, our heterogenous multi-robot team uses a specific behavior to indicate the location of mine-like objects (MLOs). Observed teammate action sequences can be learned to perform behavior recognition and task-assignment in the absence of communication. We apply Conditional Random Fields (CRFs) to perform behavior recognition as an approach to task monitoring in the absence of communication in a challenging underwater environment. In order to demonstrate the use of behavior recognition of an Autonomous Underwater Vehicle (AUV) in a cooperative task, we use trajectory based techniques for model generation and behavior discrimination in experiments using simulated scenario data. Results are presented demonstrating heterogenous teammate cooperation between an AUV and an Autonomous Surface Vehicle (ASV) using behavior recognition rather than radio or acoustic communication in a mine clearing task.**

I. INTRODUCTION

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 [1], [2]. However, this method can still degrade in overall efficiency as communication deteriorates [3]. In order to mitigate the effects of intermittent communication in a maritime environment, Sotzing and Lane [4] demonstrated that using teammate task prediction improves overall performance of a cooperative autonomous underwater vehicles (AUV) system. This is an important capability 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, the system developed by Sotzing and Lane 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.

This work was supported through internal research and development funding at the Georgia Tech Research Institute (GTRI)

T. R. Balch, is with Faculty of College of Computing, Georgia Institute of Technology, Atlanta, GA 30332 USA tucker@cc.gatech.edu

M. Novitzky, C. Pippin, T. R. Collins, and M. E. West are with the Georgia Tech Research Institute (GTRI), Atlanta, GA 30332 USA first.last@gtri.gatech.edu

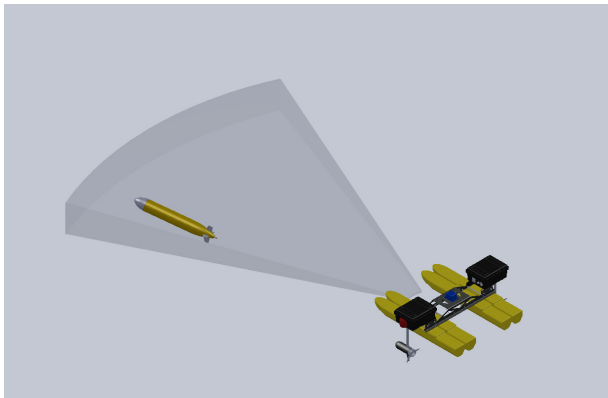


Fig. 1: A Kingfisher ASV using its sonar to observe a Yellowfin AUV. The ASV can use these observations to perform behavior recognition of the AUV.

The purpose of this research is to enable robot teams to cooperate in environments without communications. 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. Previously, we have demonstrated the ability to perform behavior recognition of a limited number of static behaviors using simulation and real sonar data [5][6]. The research presented in this paper extends our previous work by focusing on a mine clearing task which includes teammate behavior recognition so that implicit communication can be leveraged. Similarly to honey bees performing a "waggle dance" to indicate the direction and distance to a food source, our AUV will perform it's own dance behavior called the *InfinityPattern* to indicate the location of a mine-like object (MLO).

II. BACKGROUND AND RELATED WORK

As this work focuses on enabling a multi-robot team to work without the use of communication, it uses insect communication as inspiration. Insects use many modalities of communication: tactile, chemical, acoustic, and visual. The modality of particular interest is that of the honey bee's "waggle dance" [7]. During the waggle dance, a sequence of motions occur: arcing right, waggling, and arcing left.

Waggling is when the dancing bee walks in a straight line while oscillating its torso left and right. Current research indicates that the orientation of the waggle portion of the dance represents the angle between a food source and the sun. The length of the waggle indicates the energy required to reach the food source. While the waggle dance is a tactile form of communication between honeybees, scientists use computer vision techniques to aid biologists studying their behavior [8] [9] [10]. The typical workflow for these computer vision techniques is to label primitive actions which in turn makeup a behavior. Oh et al. [9] used labeled tracks of a honeybee performing the "waggle dance" so that a parametric switching linear dynamic system could learn and then accurately label each primitive motion. Work by Feldman and Balch [8] used a technique in which kernel regression labeled the primitive motions of a bee's trajectory. A Hidden Markov Model (HMM) was used to smooth the labeled sequence and subsequently identify the most likely behavior such as *dancer* or *active bee*.

Previous work on robot teammate observation has focused on identifying common resources [11] or communication broadcast [12]. Some of the earliest work by Parker [12] was built upon the ALLIANCE architecture. The author studied the effects of allowing robot teammates to observe each other's actions by varying team size and awareness level against the time and energy required to complete a task. Adding robots to a team decreased the execution time of a task by exploiting parallelism, regardless of the level of awareness. With respect to the energy measure, it was found that the energy metric improves with awareness regardless of the size of the team. This was because action recognition and awareness prevented duplication of efforts. Parker stated that communication broadcasts of current actions was an acceptable form of action recognition due to the lack of perception capable by the robots at the time.

The most similar work to that presented in this paper is by Kuniyoshi [11] in which the author used binocular vision for teammate and resource observation. Kuniyoshi coined the term *Cooperation by Observation* as defined as "Observing other agent's action, and choosing appropriate actions regarding the observed action and the current task situation." In essence, the author extends a behavior-based architecture to allow for dynamic matching of behavior resources and behavior types. Kuniyoshi tested this framework on three tasks: posing, unblocking, and passing. In each task, a teammate was observed along with a common resource. For example in the unblocking task, robot i is pushing a block and robot j determines an obstacle is in robot $i's$ path. Robot j then moves the obstacle so that robot i can deliver its block to the appropriate location.

For this paper, we will use intended recognition as described by Kanno et al. [13] as occurring when the observed agent is aware of the observer and actively cooperates. There are several distinctions that are made in order to categorically study the recognition of teammate behaviors [14], 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 from collecting data from sensors of physical movement. In this paper the term behavior recognition is used as our ASV and AUV are behavior based.

There are several approaches to activity recognition including logic-based, topologically invariant, and control theoretic techniques [15] [16] [17]. Probabilistic graphical models such as Hidden Markov Models (HMMs) and Conditional Random Fields (CRFs) are popular as they are useful for pattern recognition and have proven robust to uncertainty in many areas including speech recognition [18] [19] and handwritten script recognition [20]. 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 [21] 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 to several concurrent HMMs. A limitation of this system is that if an HMM was started at the wrong moment, it could miss the appropriate activation of the initial states. To alleviate this problem the authors ran an HMM for a specific behavior at intervals in order to capture the correct linear order of events.

Of specific importance to this work is that performed by Vail et al. [22] 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 [23] 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.

Behavior recognition of autonomous mobile robots is the focus of this research, specifically in the underwater domain similar to that of Baxter et al. [24], [25]. Baxter et al. [24] performed behavior recognition using HMMs on post-

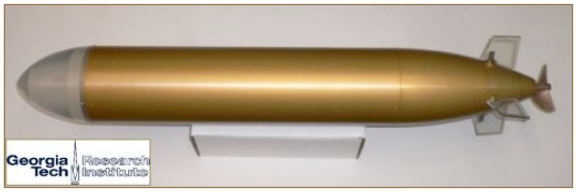


Fig. 2: Yellowfin Autonomous Underwater Vehicle - designed to be man-portable for oceanographic observation.

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 [25] 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 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 we present in this paper uses a much simpler and environmentally agnostic encoding method and using simulated location data for behavior classification.

The goal of this work is to create a system that can efficiently operate with as little explicit communication as possible. This paper investigates the feasibility of an ASV performing behavior recognition of an AUV through a sonar, as illustrated in Fig. 1. Although [25] has used simulated sonar data and [24] has used post-mission GPS trajectory analysis of an actual AUV for behavior verification, little or no research has attempted to perform cooperative behaviors based on behavior recognition in the underwater domain. Our method is presented using a simulation of an Autonomous Surface Vehicle (ASV) performing cooperative behaviors by using behavior recognition, with a CRF, of an Autonomous Underwater Vehicle (AUV) in a mine-clearing task.

III. HARDWARE PLATFORM

The motivation for this work is the need for multiple small AUVs and ASVs to perform autonomous research operations in underwater environments. The Georgia Tech Research Institute has developed the Yellowfin AUV research platform, as seen in Fig. 2, and is planning future experiments with multiple Yellowfin platforms [26]. Because of the vehicle's size, power constraints, and operating environment, communication bandwidth is limited. The vehicle was fabricated

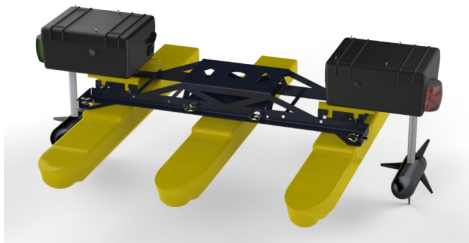


Fig. 3: Clearpath Robotics' Kingfisher M100 ASV is designed for environmental and civil engineers.

with open standards and hardware and uses open-source software components.

The heterogeneous teammate of the Yellowfin AUV is the Kingfisher M100 ASV made by Clearpath Robotics [27]. The Kingfisher, as seen in Fig. 3, is designed for environmental and civil engineers. It weighs 30 kg and its dimensions are 50x50x20.5 inches. While they are not the fastest vehicles, with a max speed of three miles per hour, they do provide a stable platform capable of keeping a station over a specified location. In addition, their size does allow for a 11.5 lbs payload which is used by more capable sensors.

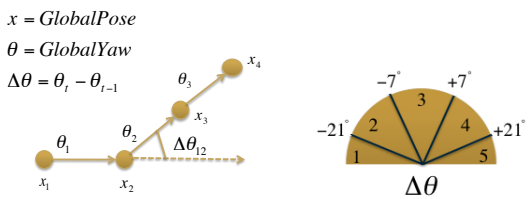
The cornerstone of our autonomy software for our marine vehicles is the open-source MOOS-IvP software suite [28]. MOOS-IvP allows for rapid deployment of autonomous vehicles along with being embraced by the autonomous marine vehicle community. Within MOOS-IvP is a simulation environment for our experiments from which our simulated data is generated.

IV. METHODS

A. 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. 4a. 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.

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



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

Fig. 4: 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).

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

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

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. 4b, 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 Conditional Random Field (CRF).

B. Conditional Random Field

Conditional random fields (CRFs) are undirected graphical models for structured classification [19]. 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.

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

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

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

2) 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.

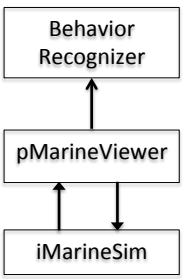


Fig. 6: Simulation workflow.

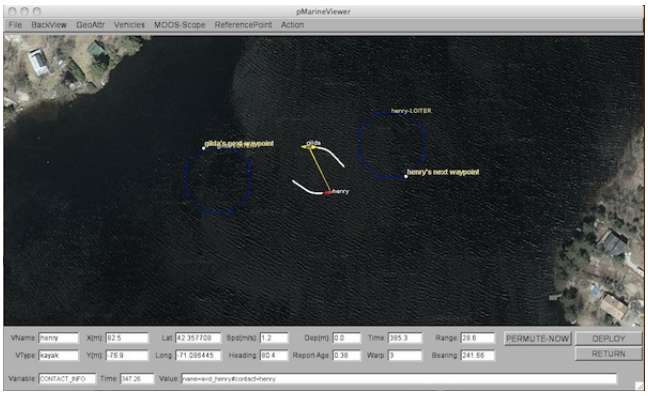
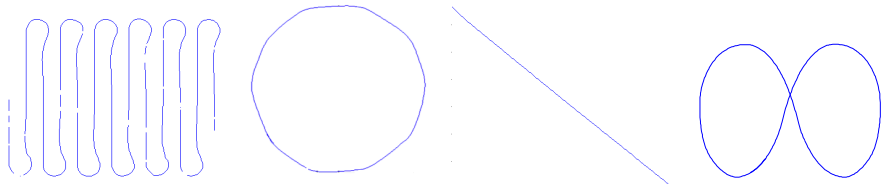


Fig. 7: An example mission running in pMarineViewer.

V. SIMULATION

The simulation environment for our experiments is provided by iMarineSim and pMarineViewer which are part of the MOOS-IvP open source autonomy package [28]. The pMarineViewer module is a GUI-based tool, as seen in Fig. 7, 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. The simulator tools allow for verification of vehicle behaviors and interactions. The experiments are performed using trajectory data gathered through simulation. The behaviors *GoToWaypoint*, *Loiter*, *SearchPattern*, and *InfinityPattern* are run within iMarineSim and viewed through pMarineViewer which shares its plotted trajectories with our behavior recognition module, as seen in Fig. 6. The locations of the AUVs are recorded as each behavior is performed. For these experiments, the perception algorithm makes the simplifying assumptions that there is only one relevant object in the scene, the Yellowfin, and that it will always be in the FOV of the sonar.

In this mine-clearing scenario, our heterogenous team consists of a Yellowfin AUV and our Kingfisher ASV, as described above. While the Yellowfin quickly performs a *SearchPattern* behavior over the designated area of operation, the Kingfisher watches silently. The Yellowfin AUV is quick and therefore can cover a large area much faster than the Kingfisher ASV. However, due to Yellowfin's dynamics it must be continuously in motion lest it sink to the bottom. As it moves quickly through the water column, it uses its BlueView forward-looking sonar to detect mine-like objects (MLOs). Because acoustic communications are restricted, the Yellowfin communicates the potential discovery of an (MLO) by performing an *InfinityPattern*, where the center of the infinity marks the spot of the MLO. Once the Kingfisher observes the *InfinityPattern* it quickly calculates the center of the behavior and proceeds to investigate whether the MLO is truly a mine to be cleared. Ten trials are performed, results are seen in Table I, of an MLO in a different location in a 40 meter by 40 meter area. In each of the trials, elapsed time is measured from a Yellowfin AUV detecting an MLO to the



(a) *SearchPattern* (b) *Loiter* (c) *GoToWaypointr* (d) *InfinityPattern*

Fig. 5: Simulated trajectories of an AUV performing *SearchPattern*, *Loiter*, *GoToWaypoint*, and *InfinityPattern* are seen in (a), (b), (c), and (d), respectively.

time when a Kingfisher ASV arrives at it's location.

VI. RESULTS

A. Training

Training of the CRF was performed with 600 static trajectories of each behavior: *SearchPattern*, *Loiter*, *GoToWaypoint*, and *InfinityPattern*, as seen in Fig. 5. After training, the CRF was verified against static trajectories of each template behavior. The CRF was able to distinguish each behavior with 100% accuracy.

B. Testing

As seen in Table II, it took the Kingfisher ASV an average of 42.5 seconds to reach the MLO after being detected by the Yellowfin AUV when using acoustic communication. This includes the time for the Yellowfin to communicate the MLO's location and travel time of the Kingfisher ASV to that location. In the no acoustic communication with behavior recognition scenario, it took the ASV an average of 201.5 seconds to identify the *InfinityPattern* behavior and travel to the MLO's location.

The time between the Yellowfin AUV detecting an MLO and the Kingfisher recognizing the *InfinityPattern* varied. This variation occurred due to the location of the MLO with respect to the *SearchPattern*. If the *SearchPattern* smoothly transitioned into the *InfinityPattern* then recognition was quick. If the transition required a turn or a loop before initiating the *InfinityPattern* then recognition took longer. As examples the quickest recognition of the *InfinityPattern* took 127 seconds while the longest recognition time took 238 seconds. In general, the *InfinityPattern* was recognized on its first iteration. However, in one trial it took two full iterations of the *InfinityPattern* before it was recognized. This is all due to the fact that there was varying amounts of confusion while the Yellowfin AUV was transitioning to the *InfinityPattern*. If the transition between *SearchPattern* and *InfinityPattern* occurred during a long leg of the *SearchPattern* then confusion would result with the leg being identified as *GoToWaypoint*. If a small loop was required to transition between *SearchPattern* and *InfinityPattern* then there would be confusion with *Loiter* or even *SearchPattern*. However, these confusions would resolve themselves as more of the *InfinityPattern* would appear. If the transition between behaviors naturally looked like it could have come from an *InfinityPattern* then recognition was quick. On the other hand, if the transition was in stark contrast then it would require a second iteration of the *InfinityPattern* so that only that pattern was observable to the algorithm. If the observation window was too small the *InfinityPattern* was mistakenly identified as the *Loiter* behavior as only a portion of the infinity is visible. At certain points during the *InfinityPattern* the behavior recognition algorithm also detected portions of *GoToWaypoint*.

TABLE I: Arrival time of the Kingfisher ASV.

Trial	w/ Comms	w/ Beh. Rec.
1	32	158
2	34	203
3	40	195
4	43	167
5	45	177
6	51	207
7	55	219
8	44	279
9	46	205
10	35	205

VII. CONCLUSION AND FUTURE WORK

The work presented here demonstrates the feasibility of performing heterogenous cooperation without explicit communication through behavior recognition. In this implementation, the behavior of importance being recognized was the *InfinityPattern* which was used to indicate the location of a mine-like object (MLO) which is comparable to the honey bee's "waggle dance" to indicate food. While it is much faster for the heterogeneous team in these experiments to communicate an MLO's location acoustically, that may not be feasible in a communication restricted environment. The most restrictive portion of the no communication/behavior recognition arrival time was the length of time required to recognize the *InfinityPattern* as in each trial the trip time for the ASV was the same. This indicates that the *InfinityPattern* may not be the best behavior to indicate the location of an MLO. Future work includes further investigation of more optimal parameters for both discretization and for the behavior CRF as accuracy can be improved. This may

TABLE II: Average arrival time of the Kingfisher ASV.

	w/ Comms	w/ Beh. Rec.
Average	42.5	201.5

include the use of more features than just the change in yaw of the Yellowfin AUV. Ultimately, these methods should be verified with more behaviors than the ones used in these experiments, as they are a small sample representation. In order to verify that this method scales, experiments in simulation will include a larger number of ASVs and AUVs. We are currently gathering real GPS trajectories produced by the Kingfisher ASV performing the above mentioned behaviors so that the presented technique can be tested on real-world data.

VIII. ACKNOWLEDGMENTS

This work was conducted through internal research and development funding by the Georgia Tech Research Institute (GTRI).

REFERENCES

[1] M. Dias and A. Stentz, "A free market architecture for distributed control of a multirobot system," in *6th International Conference on Intelligent Autonomous Systems (IAS-6)*, July 2000, pp. 115–122.

[2] B. Gerkey and M. Mataric, "Sold!: Auction methods for multirobot coordination," *Robotics and Automation, IEEE Transactions on*, vol. 18, no. 5, pp. 758–768, 2002.

[3] S. Sariel, T. Balch, and N. Erdogan, "Naval mine countermeasure missions," *Robotics & Automation Magazine, IEEE*, vol. 15, no. 1, pp. 45–52, 2008.

[4] C. Sotzing and D. Lane, "Improving the coordination efficiency of limited-communication multi-autonomus underwater vehicle operations using a multiagent architecture," *Journal of Field Robotics*, vol. 27, no. 4, pp. 412–429, 2010.

[5] M. Novitzky, C. Pippin, T. R. Collins, T. R. Balch, and M. E. West, "Conditional random fields for behavior recognition of autonomous underwater vehicles," *Distributed autonomous robotic systems*, 2012.

[6] M. Novitzky, C. Pippin, T. R. Collins, T. Balch, and M. E. West, "Behavior recognition of autonomous underwater vehicles using forward-looking sonar," *Distributed autonomous robotic systems*, 2012.

[7] K. Von Frisch, "The dance language and orientation of bees." 1967.

[8] A. Feldman and T. Balch, "Representing Honey Bee Behavior for Recognition Using Human Trainable Models," *Adaptive Behavior*, vol. 12, no. 3-4, pp. 241–250, Dec. 2004.

[9] S. Oh, J. Rehg, T. Balch, and F. Dellaert, "Learning and inference in parametric switching linear dynamic systems," in *Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on*, vol. 2. IEEE, 2005, pp. 1161–1168.

[10] T. Balch, F. Dellaert, A. Feldman, A. Guillory, C. Isbell, Z. Khan, A. Stein, and H. Wilde, "How multirobot systems research will accelerate our understanding of social animal behavior," *Proceedings of the IEEE*, vol. 94, pp. 1445–1463, 2006.

[11] Y. Kuniyoshi, "Behavior matching by observation for multi-robot cooperation," in *Robotics Research-International Symposium*, vol. 7. MIT Press, 1996, pp. 343–352.

[12] L. Parker, "The effect of action recognition and robot awareness in cooperative robotic teams," in *Intelligent Robots and Systems 95.'Human Robot Interaction and Cooperative Robots', Proceedings. 1995 IEEE/RSJ International Conference on*, vol. 1. IEEE, 1995, pp. 212–219.

[13] T. Kanno, K. Nakata, and K. Furuta, "A method for team intention inference," *International Journal of Human-Computer Studies*, vol. 58, no. 4, pp. 393–413, 2003.

[14] G. Sukthankar, "Activity recognition for agent teams," Ph.D. dissertation, CMU, 2007.

[15] M. Tambe and P. Rosenbloom, "Resc: An approach for real-time, dynamic agent tracking," in *International Joint Conference on Artificial Intelligence*, vol. 14, 1995, pp. 103–111.

[16] A. Hazan, F. Davesne, V. Vigneron, and H. Maaref, "Topological characterization of mobile robot behavior," in *Intelligent Robots and Systems, 2006 IEEE/RSJ International Conference on*. IEEE, 2007, pp. 4157–4162.

[17] Y. Demiris, "Prediction of intent in robotics and multi-agent systems," *Cognitive processing*, vol. 8, no. 3, pp. 151–158, 2007.

[18] L. Rabiner, "A tutorial on hidden markov models and selected applications in speech recognition," *Proceedings of the IEEE*, vol. 77, no. 2, pp. 257–286, 1989.

[19] J. Lafferty, A. McCallum, and F. Pereira, "Conditional random fields: Probabilistic models for segmenting and labeling sequence data," 2001.

[20] R. Nag, K. Wong, and F. Fallside, "Script recognition using hidden markov models," in *Acoustics, Speech, and Signal Processing, IEEE International Conference on ICASSP*, vol. 11, April 1986, pp. 2071 – 2074.

[21] K. Han and M. Veloso, "Automated robot behavior recognition," in *Robotics Research-International Symposium*, vol. 9, 2000, pp. 249–256.

[22] D. Vail, M. Veloso, and J. Lafferty, "Conditional random fields for activity recognition," in *Proceedings of the 6th international joint conference on Autonomous agents and multiagent systems*. ACM, 2007, pp. 1–8.

[23] D. Vail, M. Veloso, and Lafferty, "Feature selection for activity recognition in multi-robot domains," in *Proceedings of AAAI*, 2008.

[24] R. Baxter, D. Lane, and Y. Petillot, "Behaviour recognition for spatially unconstrained unmanned vehicles," in *Proceedings of the IJCAI*, vol. 9, 2009, pp. 1–8.

[25] R. H. Baxter, D. M. Lane, and Y. Petillot, "Recognising agent behaviour during variable length activities," in *Proceedings of The 19th European Conference on Artificial Intelligence*, 2010.

[26] M. West, M. Novitzky, J. Varnell, A. Melim, E. Sequin, T. Toler, T. Collins, and J. Bogle, "Design and development of the yellowfin uuv for homogenous collaborative missions," 2010.

[27] "Clearpath robotics," 2012. [Online]. Available: http://www.clearpathrobotics.com/

[28] M. Benjamin, H. Schmidt, P. Newman, and J. Leonard, "Nested autonomy for unmanned marine vehicles with moos-ivp," *Journal of Field Robotics*, vol. 27, no. 6, pp. 834–875, 2010.