Proceedings of the 2015 Winter Simulation Conference
L. Yilmaz, W. K. V. Chan, I. Moon, T. M. K. Roeder, C. Macal, and M. D. Rossetti, eds.

DETECTING TEAM BEHAVIOR USING FOCUS OF ATTENTION

Bradley J. Wimpey
Craig Lennon
MaryAnne Fields

U. S. Army Research Laboratory
28000 Powder Mill Road
Adelphi, MD 20783, USA

ABSTRACT

An autonomous mobile robot, working with human teammates, should be equipped to intelligently react to changes in team behavior without relying on directives from human team members. To respond appropriately to changes in team behavior, the robot should detect when these situations occur, and correctly classify the new team behavior. We demonstrate a method for detecting and classifying behavior changes in a simulated team, using the team's focus of attention. The method draws from Kim et al. (2010), who developed an algorithm for propagating the motion of soccer players through a vector field in order to predict locations of future action in a soccer game. Using this propagation method, our implementation extends this work by extracting statistical features from the motion information, and, looking back over a window of prior feature values, detects changes in the team behavior and classifies group activity according to a set of possible behaviors.

1 INTRODUCTION

A robotic team member would be of great value to teams which operate in hazardous areas, including teams such as rescue and aid workers, law enforcement, and military units. However, outdoor ground robots in use today are often limited to either teleoperation, or to behaviors designed to operate on a predefined piece of terrain. If a mobile robot is to work as a teammate in hazardous areas, more autonomy is required, since the unit will encounter situations in which there will be more pressing needs than to give proper commands to the robot. In particular, a robotic teammate must be able to detect when emergency situations alter the goals of the team, in order to adjust its own goals autonomously. Detecting these situations reliably would require training a robot to interpret information from a variety of sources, including visual and auditory, and to fuse this information with its current knowledge of the world. One source of information the robot could use is the movement of its teammates. In this paper, motion information derived from team movement is used as an estimation of the focus of attention (FoA) of the team, with statistical information gathered from the FoA used as features for classification of the team's behavior. The method for generating the FoA draws from the work of Kim et al. (2010), where Kim et al. constructed motion fields to predict locations of future action in a soccer game, referring to these locations as "Points of Convergence." These Points of Convergence were the physical locations at which the team's motion, propagated through the motion field, had certain characteristics. As our interest is in a team which is not bounded by a playing field, we are less interested in the physical locations at which the motion field attains certain values, and more interested in the statistical features of these values independent of physical location. We refer to these values as the focus of attention of the team.

978-1-4673-9743-8/15/$31.00 ©2015 IEEE

2 RELATED WORK

There has been work conducted using sports data to classify team formations, detect activities, and classify activities, with varying levels of success. For example, the identification of team formations (Atmosukarto et al. 2013, Hess, Fern, and Mortensen 2007) and play types (Siddiquie, Yacoob, and Davis 2009), and the recognition of team activities (Direkoğlu and O'Connor 2012, Bialkowski et al. 2013). These methods generally use a visual feed (*e.g.* television) of players wearing uniforms and playing against a background designed for good television viewing. Such optimal viewing capabilities would not be available to our robot, especially in a military application. The robot might, however, have access to global positioning system (GPS) feeds of its teammates, or some other method of knowing their positions, as well as a priori maps of the terrain on which it is working. Further complicating the application of sports-tested detection methods to other applications is the implicit structure provided by the sports field, with bounded domain and clearly established and visible goals. Such structure is absent from the applications which interest us. With this in mind, we present sports-tested work related to our efforts.

Atmosukarto et al. (2013) focused on classifying offensive team formations in American football game plays. From a single frame taken before the play began, they detected the line of scrimmage, the side of the ball of the offensive team, and classified the offensive team formation. Siddiquie, Yacoob, and Davis (2009) classified American football play types. They did not track players or use player position information directly. Instead, they extracted spatiotemporal visual features from the field of play to form a visual vocabulary, then attempted to recognize plays by comparing visual vocabulary distributions over time. To classify events in field hockey, Bialkowski et al. (2013) used two representations of player positions: field occupancy map (histogram of player counts in subsections of the field) and a team centroid representation. Direkoğlu and O'Connor (2012) calculated position distributions of a handball team playing on a handball court. They extracted motion features from sequences of these distributions to recognize game activities.

There has also been work into the recognition (Luotsinen, Fernlund, and Boloni 2007, Sukthankar and Sycara 2006) and prediction (Kim et al. 2010, Kim, Lee, and Essa 2012) of team behaviors. Luotsinen, Fernlund, and Boloni (2007) modeled movement behaviors of teams of tanks using hidden Markov models (HMM). They learned these models of team behavior from movement examples, not prediction of movement as we utilize here. Sukthankar and Sycara (2006) developed two approaches, HMMs and template modeling and matching, to model and recognize MOUT (Military Operations in Urban Terrain) behaviors. Their template modeling and matching approach utilized spatial configurations of entities such as teammates, civilians, windows, doorways, and hazards. The templates attempted to capture formations and configurations that may occur during the execution of MOUT behaviors. The HMMs were used to label behaviors, using data from a two-man team of human players on a customized version of Unreal Tournament. As mentioned previously, Kim et al. (2010) extracted regions of interest to predict future locations of action within a soccer game. Kim, Lee, and Essa (2012) computed locations of regions of interest on a field, using moving camera footage of dynamic scenes. They computed a statistical model of motion flow on the field using Gaussian process regression (Kim, Lee, and Essa 2011). Their goal of finding important locations on the field differs from our work, which detects and classifies team behavior.

3 FOCUS OF ATTENTION

We are considering a situation in which the robotic team member (RTM) accompanies a coordinated group operating as a team. The RTM can estimate the location of its teammates without remaining in constant visual contact, perhaps by a relay of their GPS estimated locations, for example. Because the RTM may need to adjust its goals autonomously, without directives from the rest of the team, it will need to interpret and classify the group's movement, and detect when there are changes in the team's behavior. The focus of attention of the team, our interpretation of the Points of Convergence values of Kim et al. (2010), provides features for detecting changes in group behavior, and also for classifying that behavior.

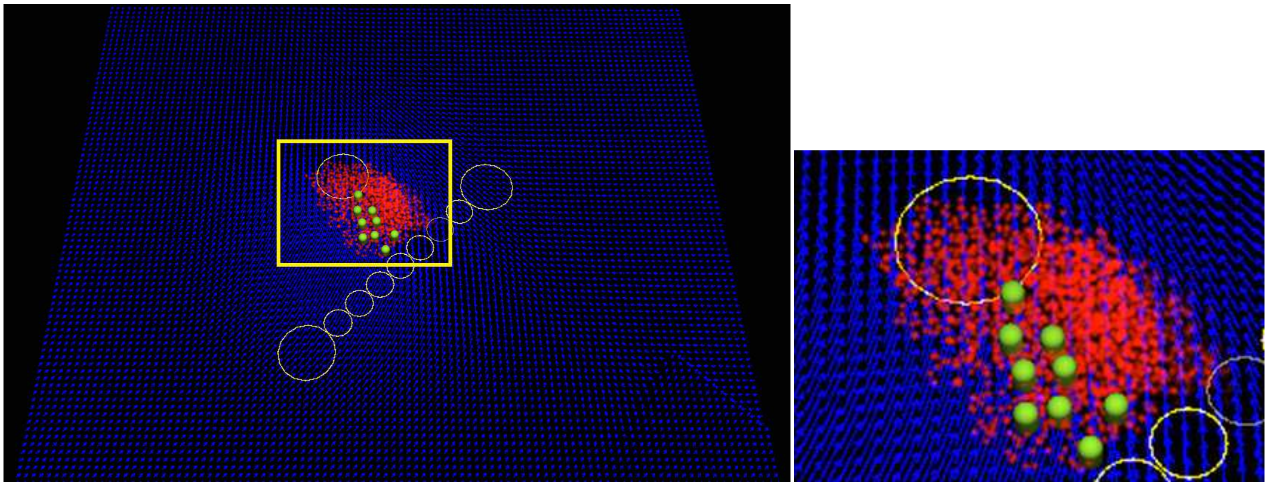


Figure 1: An illustration of our FoA Sandbox, using the Unity game engine as a simulator and visualization tool for movements of a team in the field of play. The small circles are waypoints to visit, forming the path for the group. The larger circles are the starting zone (large circle in the bottom-left), the goal (large circle in the upper-right), and in this example, the safe area (the large circle off of the path). In this illustration, a team (the green capsule-shaped objects) seeks cover off of the main path to a safe area. The motion field is represented as blue arrows, and a visualization of the FoA (values above a certain threshold) is represented with red particles.

In order to test our methodology on the type of group behavior that interested us, we developed a simulation environment in the Unity game engine (Unity 2014). Within this environment, (the "FoA Sandbox"), we developed several team behaviors, the details of which are provided in Section 4. The FoA Sandbox allows us to script the team's behaviors, and visually assess the implementation, in order to provide ground truth for training and test data. Figure 1 shows an example of the type of behavior considered; a team with nine members follows a path, and then runs off that path to seek cover. In our sandbox, the team members move over terrain that is flat, and usually featureless, though in some cases has obstacles that need to be avoided. In the future, we will consider the effects of varying elevation and terrain type, but at present they move over undistinguished, level ground. From the simulated team members in the FoA Sandbox, we could estimate velocity information, generate the motion field, and calculate Points of Convergence, used as an estimation of the team's focus of attention (see Sections 3.1-3.2). From the FoA, statistical features used for classification are extracted (see Section 3.3). Once the features are extracted, they are used in two ways: as features for a classifier which provides a constant, noisy signal describing the behavior of the team, and as a time series which is divided into separate time segments based on changes in the mean and variance of the features. These time segments are considered periods in which one behavior is recognized to have occurred, and each segment is labeled by behavior type. This provides a temporal description of the team's behavior which is more stable and less noisy.

3.1 Motion Field Implementation

Unity uses a left-handed coordinate system, with the ground plane formed by the x–z plane, and the y-axis pointing up. We considered team movement only in the ground plane. The motion field is constructed from the interpolation of estimated velocity information of the individual team members (Kim et al. 2010). Individual teammate velocities are estimated in the x and z directions from a position difference over a small time window, and then smoothed independently in the x and z dimensions with a half-Gaussian kernel. The velocity information from the individual teammates is then interpolated, using a radial basis function (RBF), to form two 2D interpolated fields of the velocities in the x and z directions, respectively. In our implementation, we used an RBF interpolation algorithm from the ALGLIB numerical analysis library (Bochkanov 2014). We then calculated the orientation and magnitude of each visible vector to form a 2D motion field over our arena of interest. At each grid point in our arena of interest, we used the x and z components of the motion vector to calculate the orientation and magnitude of the motion vector. The magnitudes are then weighted to put greater emphasis on points near the teammates. First, as our team may split into subgroups depending on the situation, the teammates are clustered using the mean shift clustering algorithm. Then, a weight for each cell is drawn from a 2D Gaussian centered on each cluster. Specifically, for each vector in the motion field, the maximum of the cluster weights at that vector's location is used to weight the magnitude. Figure 1 shows a snapshot of a seek cover event from the FoA Sandbox, with the motion field represented as blue arrows.

3.2 Propagating Motion Forward

Points of Convergence are derived from values in an iteratively constructed importance table (IT) (Kim et al. 2010). The IT is initialized with magnitudes from the motion field, and during each iteration, the motion vector at each grid point in the field "pushes" the corresponding IT value to a new location (in our implementation, a vector's magnitude has to be greater than a threshold to be propagated to a new table cell). This process is repeated, propagating at locations where the IT value has changed for each iteration, until the IT either reaches a steady state or exceeds an iteration threshold. The FoA is the values of the cells in the IT above a threshold. We use particle emitters in Unity for visualization, as shown by the red areas in Figure 1, showing the cells that have exceeded a certain threshold. For greater detail on the motion field implementation and the Points of Convergence algorithm, the interested reader is directed to the work of Kim et al. (2010). The process described so far interpolates individual motion over the surrounding terrain, and projects it into future group motion through the motion field to form the IT.

3.3 Statistical Features Used for Classification

The values in the IT can be thought of as an array of variables, which must be summarized in order to reduce the dimensionality of the problem, and to separate the effects of team movement from the effects of the relative positions of team members during the movement. Consider the column (x) and row (z) of the ITs as positions in the x–z plane, with the value (f) of the IT cell (x,z) representing the third axis. We capture the center of the focus of attention by calculating the mean (μ_x, μ_z, μ_f) over the thresholded IT. We expect μ_x and μ_z to provide information about where the team's focus is as the team moves through the environment, and use FoA velocity information derived from them in our classification. We calculate the velocity of the center of the FoA by taking the change in μ_x and μ_z over time, and use the magnitude of this velocity ($|v|$) as a feature. Since the value of μ_f had a wide range of values over the various tables, we actually use the logarithm of this value, and use the same scaling on the variance of f. The variables $|v|$ and $\log(\mu_f + 1)$ provide us with information about how fast the center of group focus is changing, and about its intensity. In order to capture information about how that focus is distributed about its center, we add the variance of x,z and the covariance of x,z to our features. In addition to detecting how spread out the group's attention is, we also want to detect if it has separated into distinct clusters. Toward that end, we binarized the IT cell values, replacing those above the threshold with 1, and those below with 0. We then conducted density-based clustering (Hennig 2014) on the binarized data to develop clusters, using the number of clusters, and the proportion of cells in the three largest clusters, as the last of our classification variables. To summarize, from each table we extracted the variables: magnitude of $|v|$, log of μ_f and log of the variance of f, the variance of x,z and the covariance matrix of x,z, the number of clusters, and the proportion of cells in the three largest clusters. We retain a history of each variable, and classify each IT based not only on the features extracted from it, but also from a history of the last nine observations of each of these variables. This is done because the ITs are produced at such a rate that behaviors should change only after several ITs have been produced. Adding a history of each variable to the classification features improved overall classification success. If ITs were produced less frequently, however, we would reduce the amount of history considered, or use only the current IT.

3.4 Detecting Changes in Statistical Features

Although classifying based on a history of our statistical features produced some reduction in noise and better results, we found that we could further reduce prediction noise by first detecting the times of significant changes in the features, and then using these times to indicate the beginnings of periods of potentially different team behavior. This breaks time into segments, and we can label each time segment based on all of the IT labels for that time segment. This method of labeling allows for the use of priors in the labeling. We could use prior knowledge to label the segment based on a weighted vote of the labels of each IT within the segment. Such weighting might come from information gathered by other sources available to the robotic team member, such as perception based on visual or audio information. With such information unavailable at the present time, we used the most common label within the time segment as the label for the time segment.

We selected a segmentation method that detects time series change points without a priori knowledge of the number of change points to be detected, but we did not require that the method run online during the simulation. This method was the hierarchical divisive estimation algorithm (HDEA) (James and Matteson 2014), which we used to segment the high-dimensional time series composed of the time-sequenced features described in Section 3.3. Thus we extracted features from each IT, and classified the table according to those features (and the features of the previous nine), recording both the features and the IT label. The time series of features in this record was subsequently segmented by the HDEA. In testing, the segmentation was performed at the end of all team action, but the HDEA could be applied after any part of the action was completed in order to segment and label behaviors up to that point, *e.g.* it could be applied every few seconds to label behavior occurring over some previous window of time, or when significant events are detected by other perception algorithms. This method removes noise from the behavior signal, with small changes in the error rates, which we discuss in Section 4.1.

4 EXPERIMENTATION

In order to test our ability to classify team behaviors, we constructed simulation scenes depicting five types of behaviors. In each scene, the group is following a path, and performs a significant action: turning, avoiding an obstacle, reversing course, seeking cover off the path, or disbanding. Brief descriptions of the behaviors are provided below.

- Turn - The piecewise linear path followed by the team contains a significant change in slope between adjacent linear segments. An example turn scene is shown in Figure 2.
- Obstacle Avoidance - One of three types of obstacles block the path: oval object, rectangular object, or multiple objects (oval + rectangle). Once a teammate gets close enough to the obstacle, he plots a course around the obstacle. An example of obstacle avoidance is shown in Figure 2. The shortest path around the obstacle(s) (indicated with the green lines in Figure 2) is calculated using the $A^\star$ algorithm (Granberg 2014).
- Reverse Course - The group is repulsed, and each team member walks directly away from the center of the triggering waypoint.
- Seek Cover - Each team member sprints to a designated safe area. A series of frames displaying a seek cover event in the FoA Sandbox is shown in Figure 3.
- Disband - The group disperses, with each team member walking in the direction opposite his cluster's center.

When using degree measurements to describe directions within the FoA Sandbox, we consider the positive z (depth) axis as North (0°), and the positive x axis (horizontal) as East (90°). In order to train and test classification, we recorded data from 240 variations of the scenes, which are described by type and behavior in Table 1. As each scene played out, a human observer recorded the time at which the test behavior started and stopped (if it stopped). From this record, each IT was given a label based on the time at which it was created. For example, an IT from a turn behavior test would have one of the labels "straight" or "turn." These were the labels used for training and testing the classifier.

Table 1: Experimental scenes by type, number of variations, and behavior.

Type	Num.	Behavior
Turn	48	Start heading: 0, 45, 90, 135, 180, 225, 270, 315, and turn by ± 45, ± 90, ± 135 ($^\circ$).
Obstacle	48	Once close to obstacle, a teammate plots a course around/through the obstacle.
Reverse	48	Walk in the opposite direction from the center of the triggering waypoint.
Seek Cover	48	Sprint to a designated "safe zone" area.
Disband	48	Walk in the opposite direction from your cluster's center.

4.1 Results

We classified each IT using K-Nearest Neighbors (Venables and Ripley 2002). The variables listed in Section 3.3 were scaled to $[0,1]$ and Euclidean distance metric was applied. We tested the success of the classifier using 5-fold cross-validation (Hastie, Tibshirani, and Friedman 2009), summarizing the results over all 5 folds in a confusion matrix (see Table 2). In this matrix, the row names indicate the true class, and the column names indicate the predicted class. Over 5 folds of cross-validation, each scene was part of the test set once, during which each IT from that scene was labeled. The total number of ITs with a given combination of true and predicted label were recorded in Table 2. For example, the cell in the row "straight" and in the column "reverse" shows that there were 15 ITs of true label "straight" that were classified as "reverse." From this table, we can discern how likely it is that an IT of a given true label will be classified as having any other label. For example, we can observe that IT classification tends to confuse "turn" and "obstacle" with "straight," but generally recognizes "disband" and "reverse," and often recognizes "seek cover."

We likewise compared the labels assigned to time segments with the true labels for those time segments. These results are summarized in the confusion matrix in Table 3, and the confusion of categories is similar. It is satisfying that the most visually distinctive events (disband and reverse) are the easiest to recognize. It appears, however, that turns may be too subtle to be reliably distinguished from straight movement based on FoA. The "obstacle" class is also difficult to distinguish from "straight."

If we had only one class with which we were concerned (*e.g.* class x), and a null hypothesis that the time segment was not of class x, then we would examine: (1) how likely are we to commit a type 1 error,

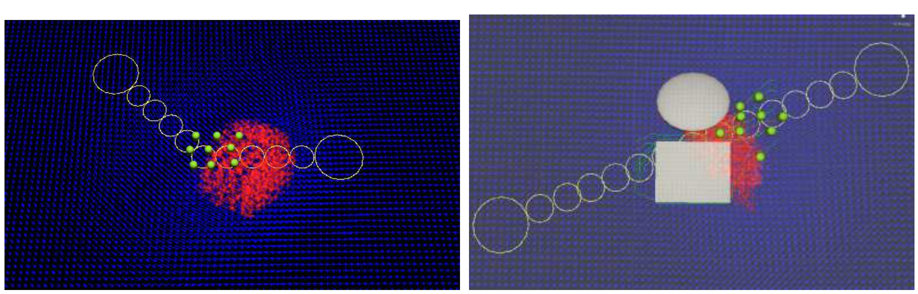
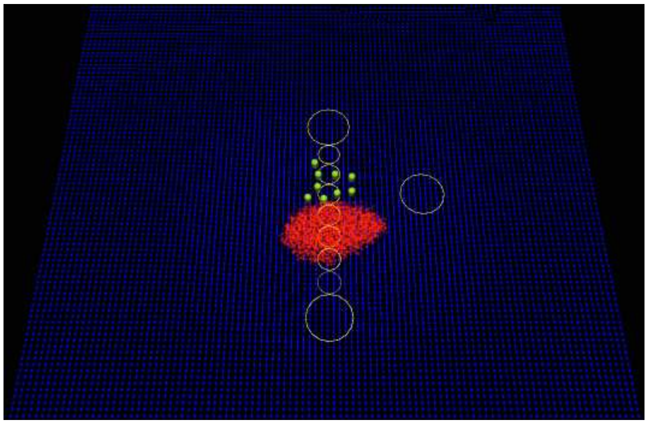
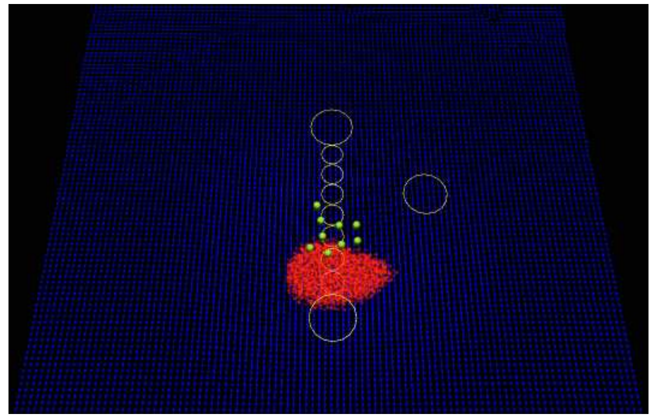


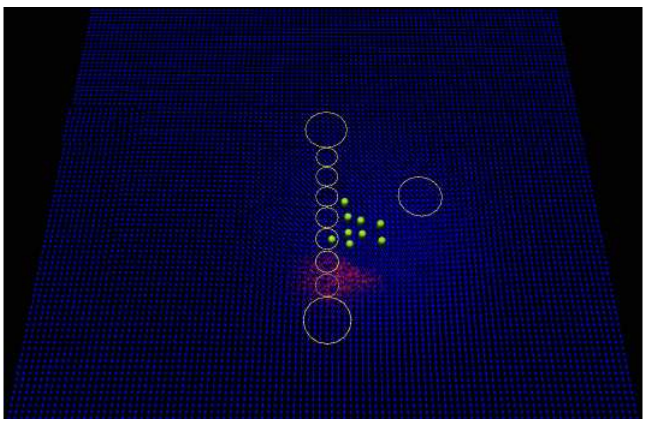
Figure 2: (left) An example turn in the path. The group starts with a heading of 135° (heading Southeast) and turns to a heading of 90° (heading East). (right) The group avoiding a "multiple object" obstacle. The shortest paths for each member to avoid the obstacle are shown in green.



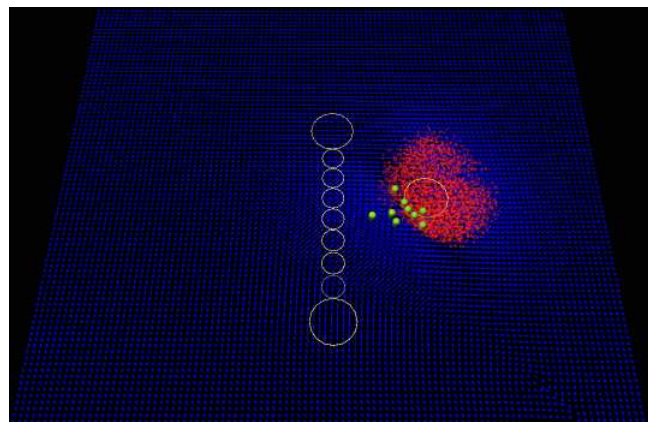
Frame 56



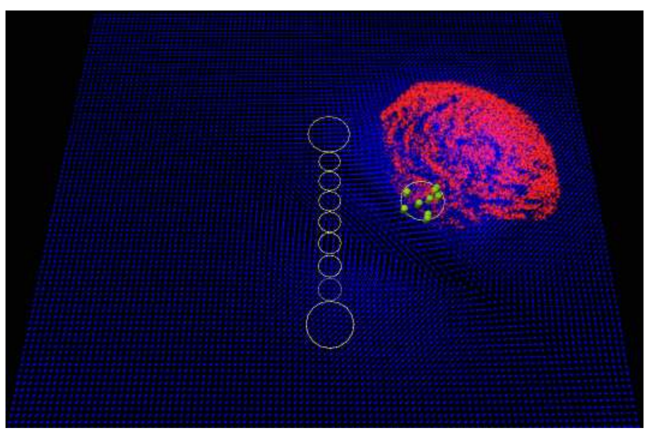
Frame 103



Frame 120



Frame 132



Frame 157

Figure 3: A sequence of frames from a seek cover event (captured at 10fps). The group is traveling from North to South when it receives a seek cover signal. The seek cover signal is triggered at frame 109, and the group then sprints toward cover in the safe zone (represented by the offset circle on the right side of the arena). Because of the sprint, the focus of attention gets pushed far beyond the safe zone.

Table 2: Confusion matrix for 5-fold CV of KNN applied to individual importance tables (ITs). Cells indicate number of ITs of true class (row) classified as predicted class (column).

True / Pred.	Disband	Obstacle	Reverse	Seek Cover	Turn	Straight
Disband	5389	83	34	4	0	167
Obstacle	240	3756	84	49	127	2779
Reverse	58	84	4429	28	2	47
Seek Cover	14	132	24	2543	31	394
Turn	2	73	11	67	1254	959
Straight	29	421	15	33	114	21051

Table 3: Confusion matrix for averaging of labels over segments of the time series. Cells indicate number of segments of true class (row) classified as predicted class (column).

True / Pred.	Disband	Obstacle	Reverse	Seek Cover	Turn	Straight
Disband	418	1	1	0	0	29
Obstacle	5	279	5	1	5	240
Reverse	4	3	346	1	0	11
Seek Cover	0	6	0	212	1	48
Turn	0	0	2	4	107	107
Straight	4	13	7	4	15	1434

and (2) how likely are we to commit a type 2 error? With multiple classes and no null hypothesis, (1) becomes: for each class x, what is the true class distribution of those IT/segments labeled as x? Likewise, (2) becomes: for each class x, what is the distribution of predicted class for those IT/segments of true class x? To answer the first question with respect to the ITs, we can take the confusion matrix in Table 2 and divide each cell by the sum of its column. The result is Table 4, in which the left side of each cell contains the percent of ITs labeled as the column class which are actually of each row class. The right side of each cell contains the same calculation performed for each time segment, *i.e.* dividing the cells of Table 3 by the column sums. For example, the cell of row "reverse" and column "disband" shows that $1.0\%/\,0.9\%$ of class labels of type "disband" were given to ITs/segments of class "reverse." This provides a way to see the costs and benefits of removing the noise via time-based segmentation and labeling. Overall, the reduction in noise comes at little cost, with an improvement in predicting the "obstacle" class, but with worse prediction of the "straight" class. In general, if an IT/segment was labeled as being of class x, then the label was correct at least $82.9\%/\,76.7\%$ of the time. To answer question (2), we take the confusion matrices in Tables 2-3 and divide each cell by the sum of its row. The result (Table 5) is the percent of ITs/segments of (true) row class which are labeled as the column class. Again, labeling by time segments results in little change. We do, however, see from this table how unreliable our detection of the "turn" and "obstacle" classes are for both individual ITs and time segments. Approximately half of the "turn" and "obstacle" classes are detected, and most of those that are not are labeled as "straight." Again, we are reminded the FoA may not be the best technique for detecting subtle events such as turns or movement around obstacles.

5 CONCLUSION AND FUTURE WORK

A robotic teammate may encounter emergency situations in a hazardous environment. When those emergency situations occur, it must detect when the team's behavior has changed so that it may update its own goals. We have presented an approach to such detection using focus of attention. Classification based on FoA

Table 4: IT confusion matrix divided by column sums, showing what percent of the predicted class (column) was actually a member of each true class (row).

T/P	Disband	Obstacle	Reverse	Seek Cover	Turn	Straight
Disband	94.0 / 97.0	1.8 / 0.3	0.7 / 0.3	0.1 / 0.0	0.0 / 0.0	0.7 / 1.6
Obstacle	4.2 / 1.2	82.6 / 92.4	1.8 / 1.4	1.8 / 0.5	8.3 / 3.9	10.9 / 12.8
Reverse	1.0 / 0.9	1.8 / 1.0	96.3 / 95.8	1.0 / 0.5	0.1 / 0.0	0.2 / 0.6
Seek Cover	0.2 / 0.0	2.9 / 2.0	0.5 / 0.0	93.4 / 95.5	2.0 / 0.8	1.6 / 2.6
Turn	0.0 / 0.0	1.6 / 0.0	0.2 / 0.6	2.5 / 1.8	82.1 / 83.6	3.8 / 5.7
Straight	0.5 / 0.5	9.3 / 4.3	0.3 / 1.9	1.2 / 1.8	7.5 / 11.7	82.9 / 76.7
Total	100	100	100	100	100	100

Table 5: IT confusion matrix divided by row sums, showing what percent of the true class (row) was labeled as each predicted class (column).

T/P	Disband	Obstacle	Reverse	Seek Cover	Turn	Straight	Total
Disband	94.9 / 93.1	1.5 / 0.2	0.6 / 0.2	0.1 / 0.0	0.0 / 0.0	2.9 / 6.5	100
Obstacle	3.4 / 0.9	53.4 / 52.1	1.2 / 0.9	0.7 / 0.2	1.8 / 0.9	39.5 / 44.5	100
Reverse	1.2 / 1.1	1.8 / 0.8	95.3 / 94.8	0.6 / 0.3	0.0 / 0.0	1.0 / 3.0	100
Seek Cover	0.4 / 0.0	4.2 / 2.2	0.8 / 0.0	81.0 / 79.4	1.0 / 0.4	12.6 / 18.0	100
Turn	0.1 / 0.0	3.1 / 0.0	0.5 / 0.9	2.8 / 1.8	53.0 / 48.6	40.5 / 48.6	100
Straight	0.1 / 0.3	1.9 / 0.9	0.1 / 0.5	0.2 / 0.3	0.5 / 1.0	97.2 / 97.1	100

calculations taken over short time intervals (*e.g* less than 1 second) were noisy, but results were improved by including a history of such calculations in our classification, and by classifying segments of time. As indicated in Section 4.1, our method shows promising behavior classification results for some distinctive behaviors, but has difficulty detecting subtle differences in behavior. Our method allows for the use of prior knowledge in the labeling of behavior, however, and incorporating reliable information from other perception systems on the robot should improve the results. We intend to extend this work to 3D, examining simulated, dismounted movement over non-level terrain, and examining the effects of different terrain types on classification.

ACKNOWLEDGMENTS

Research was sponsored by the Army Research Laboratory and was accomplished under Cooperative Agreement Number W911NF-12-2-0019. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Army Research Laboratory or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation herein.

REFERENCES

Atmosukarto, I., B. Ghanem, S. Ahuja, K. Muthuswamy, and N. Ahuja. 2013, June. "Automatic Recognition of Offensive Team Formation in American Football Plays". In *IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 991–998. Portland, Oregon, USA.

Bialkowski, A., P. Lucey, P. Carr, S. Denman, I. Matthews, and S. Sridharan. 2013, June. "Recognising Team Activities from Noisy Data". In *IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 984–990. Portland, Oregon, USA.

Sergey Bochkanov 2014. “ALGLIB”. Software available at http://www.alglib.net.

Direkoğlu, C., and N. O’Connor. 2012, October. “Team Activity Recognition in Sports”. In *Proceedings of the 12th European Conference on Computer Vision, Part VII*, 69–83. Florence, Italy.

Aron Granberg 2014. “The A* Pathfinding Project”. Software available at http://arongranberg.com/astar/.

Hastie, T., R. Tibshirani, and J. Friedman. 2009. *The Elements of Statistical Learning: Data Mining, Inference and Prediction*. Second ed. Springer.

Hennig, C. 2014. *fpc: Flexible procedures for clustering*. R package version 2.1-9.

Hess, R., A. Fern, and E. Mortensen. 2007, October. “Mixture-of-Parts Pictorial Structures for Objects with Variable Part Sets”. In *Proceedings of the IEEE International Conference on Computer Vision*, 1–8. Rio de Janeiro, Brazil.

James, N. A., and D. S. Matteson. 2014. “ecp: An R Package for Nonparametric Multiple Change Point Analysis of Multivariate Data”. *Journal of Statistical Software* 62 (7): 1–25.

Kim, K., M. Grundmann, A. Shamir, I. Matthews, J. Hodgins, and I. Essa. 2010, June. “Motion Fields to Predict Play Evolution in Dynamic Sport Scenes”. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 840–847. San Francisco, California, USA.

Kim, K., D. Lee, and I. Essa. 2011, November. “Gaussian Process Regression Flow for Analysis of Motion Trajectories”. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 1164–1171. Barcelona, Spain.

Kim, K., D. Lee, and I. Essa. 2012, June. “Detecting Regions of Interest in Dynamic Scenes with Camera Motions”. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 1258–1265. Providence, Rhode Island, USA.

Luotsinen, L., H. Fernlund, and L. Boloni. 2007, May. “Automatic Annotation of Team Actions in Observations of Embodied Agents”. In *Proceedings of the Sixth International Joint Conference on Autonomous Agents and Multi-Agent Systems (AAMAS)*, 32–34. Honolulu, Hawaii, USA.

Siddiquie, B., Y. Yacoob, and L. Davis. 2009. “Recognizing Plays in American Football Videos”. Technical report, University of Maryland.

Sukthankar, G., and K. Sycara. 2006, May. “Robust Recognition of Physical Team Behaviors Using Spatio-Temporal Models”. In *Proceedings of the Fifth International Joint Conference on Autonomous Agents and Multi-Agent Systems (AAMAS)*, 638–645. Hakodate, Japan.

Unity 2014. Software available at http://unity3d.com.

Venables, W. N., and B. D. Ripley. 2002. *Modern Applied Statistics with S*. Fourth ed. New York: Springer. ISBN 0-387-95457-0.

AUTHOR BIOGRAPHIES

BRADLEY J. WIMPEY is a Postdoctoral Research Fellow with the U.S. Army Research Laboratory. He holds a Ph.D. in Computer Science from the University of Georgia. bradley.j.wimpey.ctr@mail.mil.

CRAIG LENNON is a Mathematician/Statistician at the United States Army Research Laboratory. craig.t.lennon.civ@mail.mil.

MARYANNE FIELDS is an Applied Mathematician at the United States Army Research Laboratory. mary.a.fields22.civ@mail.mil.