Robust Recognition of Physical Team Behaviors Using Spatio-Temporal Models

Gita Sukthankar
Robotics Institute
Carnegie Mellon University
5000 Forbes Ave.
Pittsburgh, PA
gitars@cs.cmu.edu

Katia Sycara
Robotics Institute
Carnegie Mellon University
5000 Forbes Ave.
Pittsburgh, PA
katia+@cs.cmu.edu

ABSTRACT

This paper presents a framework for robustly recognizing physical team behaviors by exploiting spatio-temporal patterns. Agent team behaviors in athletic and military domains typically exhibit an observable structure characterized by the relative positions of teammates and external landmarks, such as a team of soldiers ambushing an opponent or a soccer player moving to receive a pass. We demonstrate how complex team relationships that are not easily expressed by region-based heuristics can be modeled from data and domain knowledge in a way that is robust to noise and spatial variation. To represent team behaviors in our domain of MOUT (Military Operations in Urban Terrain) planning, we employ two classes of spatial models: 1) team templates that encode static relationships between team members and external landmarks; and 2) spatially-invariant Hidden Markov Models (HMMs) to represent evolving agent team configurations over time. These two classes of models can be combined to improve recognition accuracy, particularly for behaviors that appear similar in static snapshots. We evaluate our modeling techniques on large urban maps and position traces of two-person human teams performing MOUT behaviors in a customized version of Unreal Tournament (a commercially available first-person shooter game).

Categories and Subject Descriptors

I.5 [**Pattern Recognition**]: Miscellaneous; I.2.11 [**Distributed Artificial Intelligence**]: Multiagent Systems

General Terms

Algorithms

Keywords

multi-agent plan recognition, RANSAC, HMMs

AAMAS'06 May 8–12 2006, Hakodate, Hokkaido, Japan.

1. INTRODUCTION

In certain domains tasks are too complicated to be performed by individual agents and must be achieved through the coordinated efforts of a group of agents over a period of time. To analyze performance of these tasks, we need to extend existing behavior recognition formalisms to accommodate group behaviors. Due to the increase in number of actions generated, assuming that each agent is simultaneously executing actions, team behaviors have a more complicated *temporal* structure than single agent behaviors. However team plans involving physical movement also possess a distinctive *spatial* structure, characterized by the relative positions of teammates and external landmarks, that can be exploited to classify team behaviors. This paper presents a framework for constructing spatio-temporal models to robustly recognize physical team behaviors from position traces of the agents' movement.

We demonstrate our method in the domain of MOUT (Military Operations in Urban Terrain) team planning to recognize military team behaviors customarily performed by human soldiers while moving through an urban environment. Examples of team behaviors include building entry, perimeter guarding, opponent flanking, and formation movement, such as stacked and bounding overwatch.[1] Although this paper focuses on MOUT team behaviors, our algorithms are also applicable for recognizing team behaviors in other military, robotic, and athletic domains.

In this paper, we present two classes of spatial models and illustrate how they can be used to robustly recognize team behaviors in the presence of spatial variations, noise, clutter, and human variability in behavior execution.

team templates: models encoding static spatial relationships between team members and external landmarks. These models are composed of the relative positions of map entities and can be generalized to different geographic layouts using similarity transforms. To efficiently identify team templates over large map areas without using exhaustive search, we employ an efficient randomized search technique, RANSAC (Random Sampling and Consensus) [4].

spatially-invariant HMMs: a set of Hidden Markov Model classifiers applied over short overlapping time windows to identify temporal patterns in agent team configura-

[1]See Section 3 for more details on MOUT team behaviors

tions. By representing the agents' positions in a canonical reference frame, the classifier is robust to certain spatial variations.

We evaluate our modeling and recognition techniques on two different types of data: 1) large urban simulation battle maps and 2) position traces of two-person human teams performing designated sequences of MOUT behaviors in a customized version of Unreal Tournament (a commercially available first-person shooter game).

2. RELATED WORK

Previous work on team behavior recognition has been primarily evaluated within athletic domains, including American football [6], basketball [7], and Robocup soccer simulations [14, 15]. Recognition for military air-combat scenarios has been examined in the context of event tracking [18] and teammate monitoring [8], Finally, a general framework for multi-agent plan recognition (Hierarchical Multiagent Markov Processes) [16] has been demonstrated for a single pair of humans moving around a laboratory.

The research described in this paper focuses on the pattern classification aspect of team behavior recognition—how to systematically model and identify complex spatio-temporal patterns that are created by agent team behaviors. Most previous research in the area of team behavior recognition either leverages the existence of domain-specific features to classify spatial patterns [6, 7] or relies on identifying temporal patterns in behavior sequences [18, 8, 16], largely ignoring implicit spatial information. For many athletic behaviors, researchers have been able to exploit simple region-based or distance-based heuristics to build accurate, but domain-specific classifiers. Based on the premise that all behaviors always occur on the same playing field with a known number of entities, it is often possible to discretize the playing field into grids [15] or typed regions (goal, scrimmage line) [6] that can be used to classify player actions. Prior work on spatial representations for the MOUT domain [2, 11] only addresses the problem of behavior generation, not recognition.

3. MOUT DOMAIN

MOUT (Military Operations in Urban Terrain) scenarios involve moving teams of soldiers and vehicles through heavily-cluttered urban areas to accomplish high-level strategic objectives. A common task performed by human soldiers is building clearing, during which a firing-team of soldiers must enter a building and clear it of enemy occupants and hazards; the challenge is to explore unknown areas while constantly remaining in a defensible position against enemy attack. To achieve this, soldiers move through the urban terrain using set procedures that maintain a defensive position while still propelling the team forward.

The cognitive task analysis of building clearing given in [12] emphasizes the importance of spatial cues in the soldiers' decision making process. Features such as proximity to other buildings, opportunities for cover, open spaces, windows in the building, street layout, fortifications, height of buildings, locations of stairways, obstacles, potential booby-traps, and doors, are critical cues that expert human decision-makers must take into account while performing the task. Certain team behaviors are triggered by the spatial characteristics

Figure 1: MOUT scenario in customized Unreal Tournament environment from spectator viewpoint. A pair of human players control soldiers A and B as they execute the procedure for traversing a T-shaped intersection. The bot models and animations were modified to conform to the appearance of real human soldiers rather than the larger-than-life UT fantasy fighter models.

of the terrain; for instance, firing-teams have special procedures for moving through L-shaped and T-shaped street intersections. In these cases, we hypothesize that static spatial configurations, such as the ones described in our team template models, are highly predictive of the teams' actions.

However, there are some cases, particularly for smaller two-soldier subteams, in which static spatial configurations, by themselves, lack predictive power. The second part of the paper focuses on the three such behaviors, stacked movement, bounding overwatch, and buttonhook entry commonly used during the building clearing task. These behaviors are difficult to identify solely on the basis of static snapshots due to their spatial similarities (see Figure 3).

During stacked movement the purpose is to move the team in such a way that their gun angles completely span all possible areas of approach; the team moves slowly and in synchrony. For moving through open areas or intersections, this approach is less feasible since it's difficult to cover all possible threatened areas. In this case, the bounding overwatch behavior is used; one soldier moves forward while the other remains stationary. The buttonhook entry is similar to bounding overwatch; one soldier moves through the doorway hugging the wall while the other soldier waits and guards. After the entry is clear, the second soldier moves through the doorway hugging the opposite wall. To recognize these behaviors, our classifier needs to exploit the temporal information in behavior sequences in conjunction with spatial information on the position and velocities of team members in a way that is robust to geographic variations that modify the execution of the behavior.

4. METHODS

In this section, we present our methodology for developing spatial models and using them to classify instances of team behavior. Section 4.1 discusses the representation and

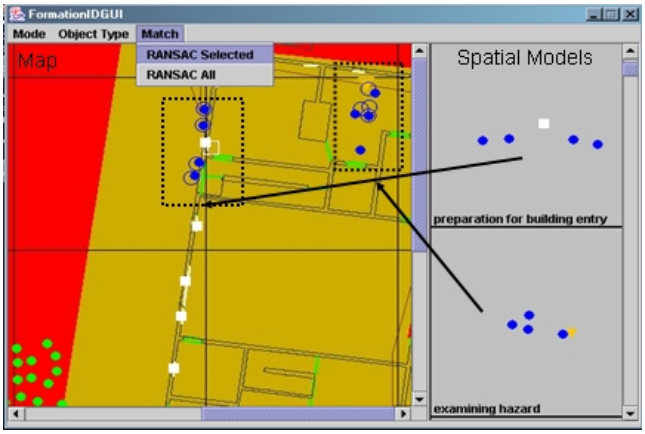

Figure 2: Spatial model authoring and matching system. The library of previously created spatial models are shown to the right of the screen; the left side of the GUI displays an annotated map to be analyzed. A fire team of soldiers (blue circles) examining a hazard (orange inverted triangle) are displayed on the map; a second fire team prepares to enter the building through the door (marked by the white rectangle). Our matching technique successfully associates both groupings of soldiers with the correct spatial models; hollow circles and rectangles show the locations of the entities as predicted by model projection.

authoring of team templates to encode static spatial relationships between team members and external landmarks. Section 4.2 provides an overview of the randomized search technique, RANSAC (Random Sampling and Consensus) that we use to efficiently and robustly identify these templates over large simulation map areas without resorting to exhaustive search. In Section 4.4 we describe the Unreal Tournament simulation environment that we use to collect data from pairs of human subjects performing the three types of statically-similar team behaviors (bounding overwatch, buttonhook entry, stacked movement). Finally, Section 4.5 presents our procedure for converting agent position traces into the canonical representation used by our Hidden Markov Model classifiers (Section 4.6).

4.1 Team Template Representation

To model team behaviors, we developed a tool that enables the author to describe behaviors by designating a set of characteristic spatial relationships that commonly occur during the execution of a behavior. Once a library of spatial models has been constructed, they can be used to classify formations of MOUT entities on a 2D annotated urban map. If labeled training data of team behaviors exists, these team templates could potentially be learned from data using a supervised classifier; however all the experiments described in this paper were created with the authoring interface shown in Figure 2.

Each team template contains the following attributes:

behavior name: Behaviors are represented by collections of spatial models; however no particular temporal structure, or execution order, is attached to the collection. Each spatial model can only belong to a single behavior; we do not include models which appear in a large set of behaviors since they are unlikely to help in discriminating between multiple behaviors.

spatial position of relevant entities: Entities are represented by a single (x,y) coordinate of their centroid; larger entities are represented as groups of points connected by visibility constraints.

entity type: For our library of MOUT behaviors, we designated eleven types of entities including person (unknown), civilian, teammate, opponent, hard cover, soft cover, empty area, windows, intersections, doorways, hazards, and objectives.

One consideration in developing models is generalization—how well do models developed for one scenario match behaviors executed in a different spatial layout? Without generalization it becomes impractical to exhaustively enumerate all possible spatial relationships that can occur across different maps. To solve this problem, we define a set of legal transforms to project models to new spatial layouts and score the quality of the match. We define the set of legal transforms to be the class of similarity transforms (rotation, translation, and scaling); these can be parameterized in homogeneous coordinates as follows:

$$\mathbf{T} = \begin{bmatrix} s\cos(\theta) & s\sin(\theta) & x \\ -s\sin(\theta) & s\cos(\theta) & y \\ 0 & 0 & 1 \end{bmatrix}$$

where θ is the angle of rotation, s is a scale factor, x is the x-translation, and y is the y-translation. This formulation can easily be extended to model three dimensional transforms by increasing the matrix to 4×4. The next section describes a robust and efficient technique for searching the space of possible transforms.

4.2 Efficient Template Matching

Given a set of spatial models and valid transforms, the problem of determining which spatial models are applicable to the current map can be solved by searching the space of potential transforms and models to find all the combinations of model plus transform that result in a match of sufficient quality. A commonly-used approach is exhaustive template matching [1]. Each template is applied to all possible locations in the map using a sliding window; the distance function is calculated over the window area and those matches that score under the threshold are retained. This process is exhaustively repeated for a range of scales and rotations, over all models in the library. Unfortunately this process is time consuming, scales poorly to higher dimensional transforms, and is sensitive to noise, occlusion and misalignment.

Instead, we employ a statistically-robust technique, Random Sampling and Consensus (RANSAC) [4], to efficiently sample the space of transforms using hypotheses generated from minimal sample sets of point correspondences. The algorithm can be summarized as follows:

hypothesis generation: entities are drawn uniformly and at random from the annotated map and associated with randomly-selected entities of the same type in the model. Two pairs of corresponding entities are sufficient to uniquely specify a transform hypothesis. This data-driven method of generating hypotheses is much more efficient than uniformly sampling the space

of possible transforms or exhaustively searching a discretization of the transform space.

hypothesis testing: Given a transform hypothesis, we project all of the entities in the model to the coordinate frame of the map and assess the quality of the match based on both spatial similarity and type matching. This gives us the likelihood that the given hypothesis could have generated the observed data in the map.

For each spatial model, we use RANSAC to randomly generate and test a large number of plausible transforms and select those hypotheses (a combination of a model and a valid transform) with match quality better than a specified threshold.

Since our spatial transforms have four degrees of freedom, they can be fully specified by two pairs of point correspondences. First, we randomly select two entities from the model under consideration; then based on the types of the entities (e.g., civilians, hard cover, hazard) we randomly select candidate entities on the map with compatible object types. The positions of these entities is used as the minimal set to generate a transform hypothesis as follows.

Given the minimal set $\{(x_1, y_1), (x_2, y_2)\}$ from the model and the corresponding set of points $\{(X_1, Y_1), (X_2, Y_2)\}$ from the map, we generate a third virtual pair of correspondences $(x_3, y_3) \mapsto (X_3, Y_3)$ where

$$\begin{aligned} x_3 &= x_1 + y_2 - y_1 \\ y_3 &= y_1 + x_1 - x_2 \\ X_3 &= X_1 + Y_2 - Y_1 \\ Y_3 &= Y_1 + X_1 - X_2 \end{aligned}$$

From these three correspondences, we can directly recover $\mathbf{T}$ using matrix inversion.

$$\begin{bmatrix} t_{11} & t_{12} & t_{13} \\ t_{21} & t_{22} & t_{23} \\ t_{31} & t_{32} & t_{33} \end{bmatrix} = \begin{bmatrix} X_1 & X_2 & X_3 \\ Y_1 & Y_2 & Y_3 \\ 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} x_1 & x_2 & x_3 \\ y_1 & y_2 & y_3 \\ 1 & 1 & 1 \end{bmatrix}^{-1}$$

This is a solution to a general affine transform [5] given three pairs of point correspondences, however $\mathbf{T}$ is guaranteed to be a valid, orientation-preserving similarity transform due to our construction of the third point.

To score a hypotheses, we transform the location of every entity in the model to the map using the transform $\mathbf{T}$. Each model entity contributes a positive vote for the given hypothesis if the distance from its predicted location to the closest map entity of compatible type falls below a specified threshold. The quality of a hypothesis is defined as the normalized sum of these individual votes.

Since RANSAC stochastically searches the space of possible transforms it is not guaranteed to find the best match. However the following formula can be used to determine how many iterations are necessary to achieve the best match with a specified probability of success [19]:

$$m = \left\lceil \frac{\log(1-P)}{\log[1-(1-\epsilon)^s]} \right\rceil$$

P is the target probability (e.g., $P = 0.99$ means the best match is found 99% of the time). s is the number of elements required to define the minimal set ($s = 2$ since a similarity transform requires 2 pairs of point correspondences). ϵ is the expected fraction of outliers in the data set. In traditional RANSAC applications ϵ is typically only about 0.1 (10% of the points are expected to be invalid). For our application, the fraction of outliers refers to the number of map annotations that do not match a single model; since each map actually contains multiple models in addition to entities that do not match any model the fraction of expected outliers is approximately 0.95. From the formula above, this indicates that the number of RANSAC iterations required to reliably find the best match is 1840. Note that executing the given number of iterations does not guarantee a particular level of classification accuracy; it merely ensures that there is only a 0.01% probability that a similarity transform exists that would achieve a higher score than the one returned by RANSAC. In Section 5.1, we present classification results for identifying team behaviors from 2D annotated maps of simulated urban layouts using our team template model and RANSAC.

4.3 Spatio-temporal Classification

There are some cases, particularly for smaller two-soldier subteams, in which static spatial configurations, by themselves, lack predictive power. To recognize these behaviors, our classifiers need to exploit the *temporal* information in behavior sequences in conjunction with spatial information on the position and velocities of team members. Since team behaviors can be executed in a variety of terrains, the classifiers must be robust to deviations in behavior execution caused by the team's response to local terrain features. However, arbitrarily introducing similarity transforms in the middle of a behavior sequence can destroy the spatio-temporal pattern created by the team's movements. To address the problem, we developed spatially-invariant classifiers, by transforming our position data into a canonical reference frame defined by the team's motion, and applying a set of Hidden Markov Model classifiers to recognize three statically similar team behaviors (bounding overwatch, buttonhook entry, stacked movement).

4.4 Human Data Collection

To evaluate our spatio-temporal classifiers, we collected data from a pair of human players using our modified Unreal Tournament game interface to manipulate "bots" through a small urban layout while performing a particular sequence of team behaviors (see Figure 1). Note that the subjects were not playing Unreal Tournament, but using Unreal Tournament to execute sequences of commonly used MOUT team maneuvers. To directly monitor the performance of human players, we customized Unreal Tournament (UT) using the game development language *Unrealscript*. Many of the original UT game classes were written in *Unrealscript* and thus can be directly subclassed to produce modified versions of the game (known as mods); for example, Gamebots [9] is an example of a mod that allows external programs to control game characters using network sockets.

We developed our own `TrainingBot` mod that allows us to save the state of all the bots in the scenario; currently we save each player's ID number, position $(x, y.z)$, and rotation (θ,ϕ) every 0.15 seconds. This information is useful for both offline behavior analysis and for a separate replay mode that allows us to create bots that follow the paths recorded by the original players.

4.5 Canonical Representation

Due to the continuous nature of the domain, automatically determining the exact transition points between team behaviors is a difficult problem. While approaching and entering buildings, the players continue moving their bots, changing team behaviors as appropriate for the physical layout. We address this issue by dividing the traces into short, overlapping time windows during which we assume that a single behavior is dominant; these windows are classified independently as described in Section 4.6. To recognize team behaviors performed in different physical layouts, it is important for our classifier to be rotationally- and translationally-invariant; we achieve this by transforming the data in each window into a canonical coordinate frame as described below.

More formally, we define:

- $a \in 1, \ldots, A$ is an index over A agents;
- j is an index over W overlapping windows;
- $t \in 1, \ldots, T$ is an index over the T frames in a given window;
- $\mathbf{x}_{a,j,t}$ is the vector containing the (x, y) position of agent a at frame t in window j.

The centroid of the positions of the agents in any given frame can be calculated as:

$$\mathbf{C}_{j,t} = \frac{1}{A} \sum_{\forall a} \mathbf{x}_{a,j,t}.$$

We describe the configuration of the agent team at any given time relative to this centroid to achieve translation invariance. However, rather than rotating each frame independently we define a shared canonical orientation for all of the frames in a window. This is important because it allows us to distinguish between similar formations moving in different directions (e.g., agents moving line abreast vs. single file). One standard technique for defining a canonical orientation is to use the principal axis of the data points for that window, which can be calculated using principal component analysis (PCA). However for efficiency we have empirically determined that we can achieve similar results by defining the canonical orientation as the displacement of the team centroid over the window: $\mathbf{d}_j = \mathbf{C}_{j,T} - \mathbf{C}_{j,1}$.

We rotate all of the data in each window so as to align its canonical orientation with the x-axis, using the rotation matrix $\mathbf{R}_j$. Thus the canonical coordinates, $\mathbf{x}'$, can be calculated as follows: $\mathbf{x}'_{a,j,t} \equiv \mathbf{R}_j \mathbf{x}_{a,j,t} - \mathbf{c}_{j,t}$. Our current recognition technique (described in Section 4.6) also relies on observations of agents' velocity as a feature which we locally compute as: $\mathbf{v}_{a,j,t} \equiv ||x'_{a,j,t+1} - x'_{a,j,t}||$.

4.6 HMM Classification

For each canonically-transformed window in our trace, our goal is to select the best behavior model. We perform this classification task by developing a set of hidden Markov models (HMMs), one for each behavior b, and selecting the model with the highest log-likelihood of generating the observed data. Our models ($\{\lambda_b\}$) are parameterized by the following:

- N, the number of hidden states for the behavior;
- $\mathbf{A} = \{a_{ij}\}$, the matrix of state transition probabilities, where $a_{ij} = Pr(q_{t+1} = j | q_t = i), \forall i, j$ and q_t denotes the state at frame t;
- $\mathbf{B} = \{b_i(o_t)\}$, where $b_i(o_t) = \mathcal{N}(\mu_i, \Sigma_i)$. The observation space is continuous and approximated by a single multivariate Gaussian distribution with mean, μ_i and a covariance matrix, Σ_i, for each state i;
- $\pi = \{\pi_i\}$, the initial state distribution.

For our problem, given A agents in a team, the observations at time t and window w are the tuple:

$$o_t = (\mathbf{x}'_{1,w,t}, v_{1,w,t}, \ldots, \mathbf{x}'_{A,w,t}, v_{A,w,t}).$$

We determine the structure for each behavior HMM based on our domain knowledge. For instance, the stacked behavior can be described using only two states ($N = 2$), whereas we represent the more complicated bounding overwatch behavior using six states connected in a ring. Each hidden state captures an idealized snapshot of the team formation at some point in time, where the observation tuple (in canonical coordinates) is well modeled by a single Gaussian. Rather than initializing the HMMs with random parameters, we use reasonable starting values. These can be polished using expectation-maximization (EM) [3] on labeled training data.

To determine the probability, $Pr(o_{1\ldots T}|\lambda_b)$, of generating the observed data with the model λ_b, we employ the forward algorithm [13] as implemented in the Hidden Markov Model toolbox [10]. We classify each window segment with the label of the model that generated the highest log-likelihood.

5. RESULTS

We evaluate our methods using two sets of experiments: 1) formation recognition in simulated 2-D overhead maps of urban areas annotated with the location of MOUT entities (see Section 4.1). 2) behavior identification from activity traces of two-person human teams performing sequences of MOUT behaviors (see Section 4.4). The former assesses the accuracy of the RANSAC-based method while the latter examines the performance of spatially-invariant HMMs. The two methods are complementary, and Section 6 discusses strategies for combining them to further improve recognition accuracy.

5.1 Team Template Matching

To test the robustness of our team template matching approach, we add clutter to the maps and distort formations by perturbing the positions of MOUT entities. Figure 4 reports the precision (fraction of correctly-classified results) and recall (fraction of formations that were detected) of our classifier under different conditions of clutter and location perturbation. Note that our approach independently matches each template against the data; thus, all matches that exceed the threshold score are reported as detections. The precision/recall curves are generated by varying this threshold parameter. There is no intrinsic restriction against assigning the same map entity to different templates — this enables us to create templates corresponding to a team and its component sub-teams, and to simultaneously recognize both.

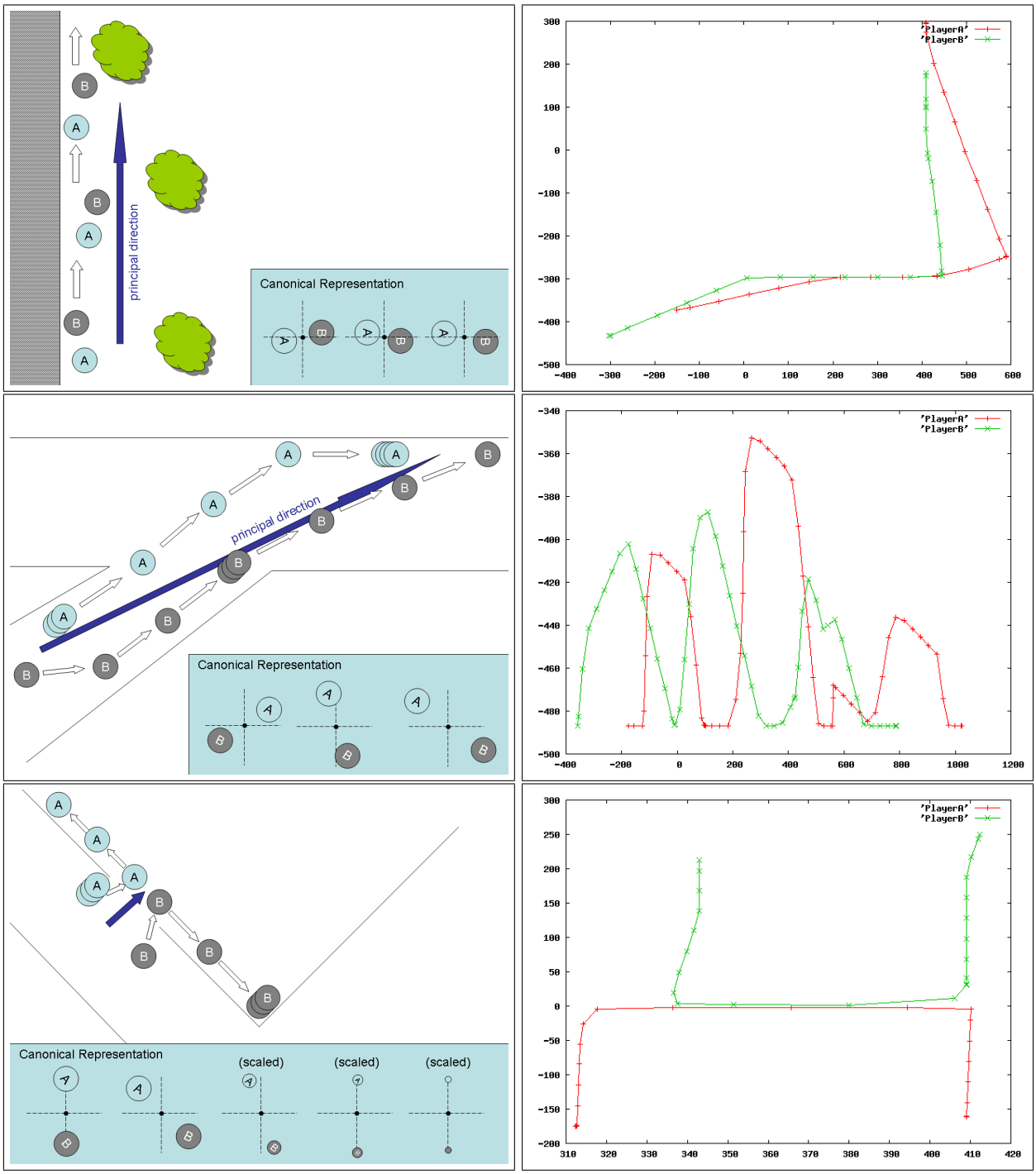

Figure 3: Team Behaviors: Stacked Formation (top), Bounding Overwatch (middle), Buttonhook Entry (bottom). Schematics for each behavior, along with the canonical representation for several frames, are depicted in the left column. A sample raw trace for each behavior is shown in the right column; the coordinates of the axes are in Unreal Tournament length units.

In each experiment, we randomly place fifty MOUT formations on the urban map and report the precision and recall averaged over ten RANSAC searches. The left panel of Figure 4 shows the effects of adding clutter (spurious MOUT entities of the appropriate type) to the map, without increasing the number of RANSAC iterations. The percentage of clutter is measured against the total number of MOUT entities in the formations on the map (thus 100% clutter denotes a 1:1 ratio between spurious and desired MOUT entities). The results show that, as expected, the RANSAC-based approach is very resistant to the presence of spurious entities on the map, and that precision/recall are both very high even at the extreme clutter levels. The right panel of Figure 4 shows precision/recall results for experiments where the locations of each of the MOUT entities were perturbed with iid Gaussian noise. As expected, the performance degrades as noise is added since the spatial configuration of the formation ceases to resemble the formation represented by the idealized model. However, we note that the technique is successful at identifying an acceptable number (80%) of the formations under reasonable noise conditions.

5.2 Spatially-Invariant HMMs

To evaluate our spatio-temporal classification method, we developed HMM models for three behaviors typically employed by two-person firing teams during the building clearing task: stacked movement, bounding overwatch, and buttonhook entry (see Figure 3). Note that these three behaviors look very similar in static snapshots and can only be robustly recognized by observing spatio-temporal traces. Position data was simultaneously recorded from two subjects at 0.15 second intervals using our `TrainingBot` mod (see Section 4.4). Players executed team behaviors in predesignated sequences, transitioning smoothly from one behavior into the next, adapting each behavior as needed to the local physical layout (turning corridors, entering rooms). The traces were divided into overlapping 20 frame (3 second) windows, which were transformed into a canonical coordinate frame as described in Section 4.5 (illustrated in the inset of Figure 3). The window size was empirically selected base on the observed average speed of the MOUT soldiers.

By using real data collected from human players rather than simulated traces, we can evaluate the robustness of our approach to realistic deviations during behavior execution. Figure 3 (right) shows a raw trace for each behavior; note that consecutive executions of the same behavior exhibit significant variation, particularly noticeable in the bounding overwatch behavior. Each behavior was also performed in a variety of local physical layouts. Table 1 presents the classification results (confusion matrix) for the three modeled behaviors; the accuracy of the HMM approach is good, particularly for the stacked formation. Buttonhook entry is sometimes confused with bounding overwatch, as may be expected from similarities in the canonical representation shown in Figure 3.

Table 1: Confusion matrix for HMM behavior classification. The ground truth is given in the left column; the classification result is given in the top row. The spatially-invariant Hidden Markov Model approach achieves good accuracy. Buttonhook entry is often confused with bounding overwatch, as may be expected from similarities in the canonical representation as shown in Figure 3.

	stacked	bounding	buttonhook
stacked	90%	10%	0%
bounding	14%	67%	19%
buttonhook	0%	33%	67%

6. DISCUSSION

The two approaches presented in this paper, team templates and spatially-invariant HMMs, are complementary. Team templates are easy to author and can be rapidly executed over large, cluttered maps. Spatially-invariant HMMs are more difficult to design, but are well-suited for classifying statically-indistinguishable behaviors. We believe that combining outputs from both models should lead to improved recognition accuracy. Team templates are effective at determining the spatial context of an observed behavior, enabling improved discrimination between behaviors that display a similar temporal structure. For instance, although a buttonhook entry and a bounding overwatch often appear similar to the spatially-invariant HMM, the presence of context can enable us to disambiguate them since the former behavior is typically performed to enter a door or window, while the latter is used to move the formation down a corridor or street.

We are interested in applying our algorithms to the recognition of large team formations [17]. The efficient team template matching strategy described in the first half of the paper scales well to larger team sizes. Introducing more points into the team template is actually beneficial for two reasons: 1) the minimal set required to calculate the transform is independent of team size; 2) including additional model entities improves the robustness of the matching since a greater number of entities can contribute votes in support of the best hypothesis, and it is less probable that a configuration of spurious entities will be a good match.

In principle, the spatially-invariant HMM approach can also be applied to larger teams; however constructing models for teams with more members can become difficult since having more team members creates many more possible configurations (even in our canonical representation). For our particular domain, separating larger teams (of 8 or 16) into two-person subteams which can be be analyzed separately is a feasible approach. We are investigating strategies for combining multiple classifiers to recognize team behaviors; using repeated executions of static template matching followed by the use of more specialized classifiers designed for detecting particular behaviors. Larger team behaviors would be detected by combining the outputs of specialized classifiers (e.g., detecting invariants) rather than using a single highly-complex classifier.

7. CONCLUSION AND FUTURE WORK

This paper presents two methods for recognizing physical team behaviors from spatio-temporal movement patterns. We demonstrate that our techniques are robust to the effects of local spatial variations, clutter, noise, and human variability during behavior execution. In future work, we are particularly interested in the following problems: 1) incorporating the outputs of our classifiers into a framework

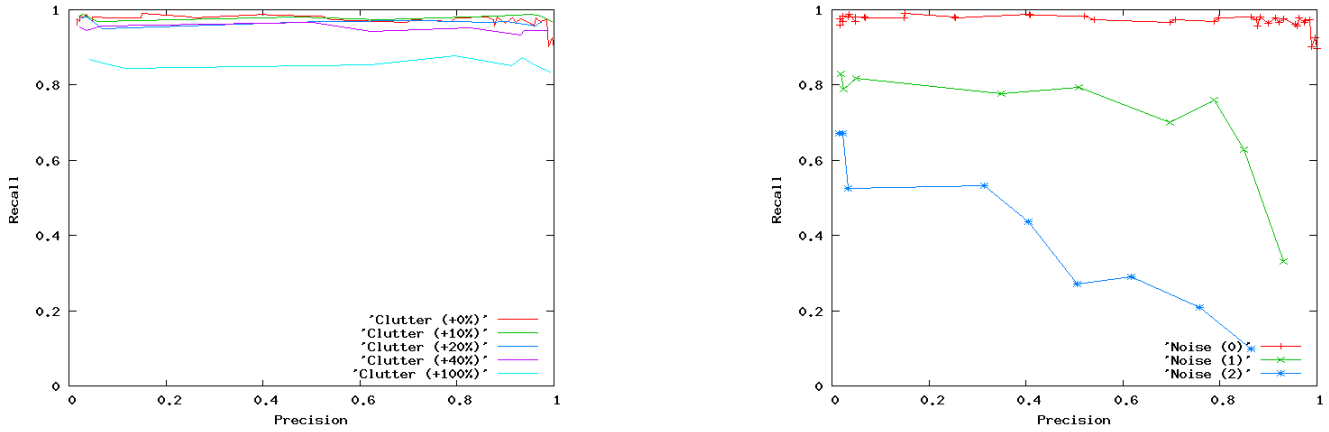

Figure 4: Precision and recall curves for matching team templates using the RANSAC-based method. Fifty formations were placed on an urban map using randomized similarity transforms. Each run employed 100,000 iterations and the precision/recall averaged over 10 trials is shown. The left panel shows the effect of adding spurious MOUT entities (clutter) while maintaining the same number of iterations. The right panel shows the results of distorting observed formations by perturbing the location of each MOUT entity on the map with iid Gaussian noise.

for symbolic team plan recognition; 2) evaluating our models on MOUT data from human domain experts; and 3) evaluating alternate temporal classifier formalisms.

8. ACKNOWLEDGEMENTS

The authors thank Rahul Sukthankar for proof-reading the document and Reid van Lehn for his work on software development. This work has been supported by AFOSR grant F49620-01-1-0542.

9. REFERENCES

[1] D. Ballard and C. Brown. *Computer Vision*. Prentice-Hall, 1982.

[2] B. Best and C. Lebiere. Spatial plans, communication, and teamwork in synthetic MOUT agents. In *Proc. Behavior Representation in Modeling and Simulation Conference (BRIMS)*, 2003.

[3] R. Duda, P. Hart, and D. Stork. *Pattern Classification*. Wiley & Sons, Inc, 2001.

[4] M. Fischler and R. Bolles. Random sample consensus: A paradigm for model fitting with application to image analysis and automated cartography. *Communications of the ACM*, 24(6), 1981.

[5] R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2000.

[6] S. Intille and A. Bobick. A framework for recognizing multi-agent action from visual evidence. In *Proc. AAAI*, 1999.

[7] M. Jug, J. Pers, B. Dezman, and S. Kovacic. Trajectory based assessment of coordinated human activity. In *Proc. International Conference on Computer Vision Systems (ICVS)*, 2003.

[8] G. Kaminka and M. Tambe. Robust agent teams via socially attentive monitoring. *Journal of Artificial Intelligence Research*, 12:pp.105–147, 2000.

[9] G. Kaminka, M. Veloso, S. Schaffer, C. Sollitto, R. Adobbati, A. Marshall, A. Scholer, and S. Tejada. Gamebots: A flexible test bed for multiagent team research. *Communications of the ACM*, 45(1), 2002.

[10] K. Murphy. The Bayes Net Toolbox for Matlab. *Computing Science and Statistics*, 33, 2001.

[11] D. Pearson and J. Laird. Redux: Example-drive diagrammatic tools for rapid knowledge acquisition. In *Proc. Behavior Representation in Modeling and Simulation Conference (BRIMS)*, 2004.

[12] J. Phillips, M. McCloskey, P. McDermott, S. Wiggins, and D. Battaglia. Decision-centered MOUT training for small unit leaders. Technical Report 1776, U.S. Army Research Institute for Behavioral and Social Sciences, 2001.

[13] L. Rabiner. A tutorial on Hidden Markov models and selected applications in speech recognition. *Proc. IEEE*, 77, 1989.

[14] P. Riley and M. Veloso. On behavior classification in adversarial environments. In L. Parker, G. Bekey, and J. Barhen, editors, *Distributed Autonomous Robotic Systems 4*. Springer-Verlag, 2000.

[15] P. Riley and M. Veloso. Recognizing probabilistic opponent movement models. In A. Birk, S. Coradeschi, and S. Tadorokoro, editors, *RoboCup-2001: Robot Soccer World Cup V*. Springer Verlag, 2002.

[16] S. Saria and S. Mahadevan. Probabilistic plan recognition in multiagent systems. In *Proc. nternational Conference on AI and Planning Systems (ICAPS)*, 2004.

[17] G. Sukthankar. Thesis proposal: Activity recognition for physically-embodied agent teams. Technical Report CMU-RI-05-44, Robotics Institute, Carnegie Mellon, 2005.

[18] M. Tambe. Tracking dynamic team activity. In *Proc. AAAI*, 1996.

[19] G. Zu and Z. Zhang. *Epipolar Geometry in Stereo, Motion, and Object Recognition*. Kluwer, 1996.