Identifying Physical Team Behaviors from Spatial Relationships

Gita Sukthankar
Katia Sycara
Robotics Institute
Carnegie Mellon University
5000 Forbes Avenue
Pittsburgh, PA 15213
412-268-8283, 412-268-6795
{gitars,katia+}@cs.cmu.edu

Keywords:
spatial reasoning, plan recognition, physical models of human movement, MOUT

ABSTRACT: *An important part of classifying MOUT (Military Operations in Urban Terrain) team behaviors is recognizing subtle spatial relationships between physical entities: opponents waiting in ambush, teammates organizing around a rendez-vous point, and potentially dangerous cul-de-sacs. In this paper, we present a RANSAC (Random Sampling and Consensus) based algorithm for identifying spatial relationships of MOUT entities based on a model library; possible configurations are scored based on a similarity function that incorporates information on entity type matching, transform validity, spatial proximity, and preservation of visibility constraints. Configurations can include both static entities (doors, buildings, hazards) and dynamic ones (opponents, teammates, and civilians). The output from our algorithm is used as a state feature for our behavior recognition system to recognize team behaviors from sequences of state transitions. We demonstrate that our algorithm is robust to spatial variations, generalizes across scenarios, and can be executed in real-time.*

1. Introduction

Although there has been some work on the problem of combining spatial representations with cognitive models to generate realistic human behavior, less attention has been devoted to the inverse problem of creating spatial models to recognize high-level behaviors from sequences of low-level physical movements. Ideally the same model can be used to generate human-like behavior and reused as part of a recognition system to evaluate the human trainee's performance. This approach is often employed in model-tracing tutoring systems such as the MOUT tutor described in (Livak, 2004); however we believe that efficient recognition of complex behaviors is best achieved by creating models optimized for the recognition task.

The capability to correctly infer the human trainee's intention from sequences of actions is useful in many applications. For certain teamwork tasks, synthetic teammates must anticipate their human teammates' intentions to function as effective partners, particularly in the absence of communication. In adversarial domains, computer-generated opponents must be able to predict the actions of their human adversary to simulate the experience of competing against a domain expert. Although this is commonly achieved in computer games by allowing the computer to "cheat" (providing the computer with extra information which is unavailable to the human), this approach can sabotage the training experience. Also plan recognition is an important prerequisite toward building "coaching" systems capable of giving detailed commentary about a human trainee's performance.

Standard recognition techniques, such as (Pynadath, 1999), exploit temporal regularities in action sequences generated during plan execution to eliminate unlikely explanations. In physical domains, plan execution also generates distinctive spatial characteristics that can be used to facilitate behavior recognition; in previous work (Sukthankar & Sycara, 2005), we examined the use of geographic features and motion information to recognize physical behaviors generated by a single simulated agent.

Due to the increase in number of actions generated, assuming that each team member is simultaneously executing actions, team behaviors have a more complicated temporal structure than single agent behaviors. However, team plans executed in physical domains often possess a distinctive spatial structure which can be exploited to identify team behaviors. This spatial structure includes the relative physical position of the team members and the physical position of team members in relation to static entities such as buildings, hazards, and doorways.

In this paper, we present a model for representing the spatial characteristics of physical human team behaviors and an algorithm for rapidly identifying currently applicable models from a library of previously created models. We demonstrate our approach by modeling and recognizing the physical behaviors in a simulated MOUT (Military Operations in Urban Terrain) scenario of a firing team moving through an urban area; recognition is performed by analyzing snapshots of an annotated 2D overhead map.

2. MOUT Domain: Building Clearing

Looking at the MOUT cognitive task analysis as given in (Phillips, McCloskey, McDermott, Wiggins, & Battaglia, 2001), we notice that spatial cues dominate all the task-focused decision requirements of building clearing MOUT operations. The six *task-focused* decision requirements include: 1) securing the perimeter; 2) approaching the building; 3) entering the building; 4) clearing the building; 5) maintaining and evaluating security; 6) evacuating the building.

For these tasks, 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 human decision-makers must use to determine how to achieve the building clearing task. Dynamic spatial features such as civilian activity and presumed enemy location also factor into several stages of the decision making process.

Out of the five *task-independent* decision requirements spatial cues are used for three of them: 1) maintain the enemy's perspective; 2) maintain big picture and situation awareness; 3) project into the future. For these three decision requirements, important spatial cues include: location of hallways, stairwells, general building layout, teammates' locations, and last known enemy positions.

In this paper, we tackle the inverse problem—from a set of spatial cues, we want to recognize which behaviors are being executed. Given the key importance of spatial cues for the task decision requirements we believe that accurately representing and generalizing spatial patterns is key to the problem of efficient behavior recognition.

3. Related Work

There has been some recent work on combining spatial formalisms with cognitive architectures to generate tactical team movement. Best and Lebiere (2003) created an authoring interface for developing spatial team plans to be executed with the ACT-R architecture; correspondingly in SOAR, Pearson and Laird (2004) developed an example-driven authoring tool to automatically create SOAR productions from spatial diagrams of MOUT scenarios drawn by SMEs (subject matter experts). Sukthankar *et al.* (2004) created an interface for modifying motion capture data to create synthetic MOUT soldiers with realistically variable movement but did not create a general spatial representation for directing movement. Although these spatial descriptions could be reused to recognize behaviors in a limited range of situations, we believe that they lack the power to generalize to large deviations in spatial layout engendered by the use of a radically different map layout or missing spatial elements produced by occlusion or incomplete detection. Our spatial representation is designed to efficiently match behaviors to models that were originally designed for significantly different spatial layouts and to robustly match behaviors even if key spatial entities are hidden.

4. Method

Our method requires an initial phase of constructing one or more spatial models to correspond to each physical behavior; we designed an authoring graphical user interface (GUI) to facilitate the model creation process (see Figure 1). Once a library of spatial models has been constructed, they can be used to classify formations of MOUT entities on a 2D annotated map as being characteristic of a particular team behavior. The same technique can also be used to classify spatial groupings of static entities (e.g., doors and walls) to be used as decision cues for the building clearing task.

4.1. Spatial Representation of MOUT Entities

To model MOUT 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 the behavior. The model 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 ap-

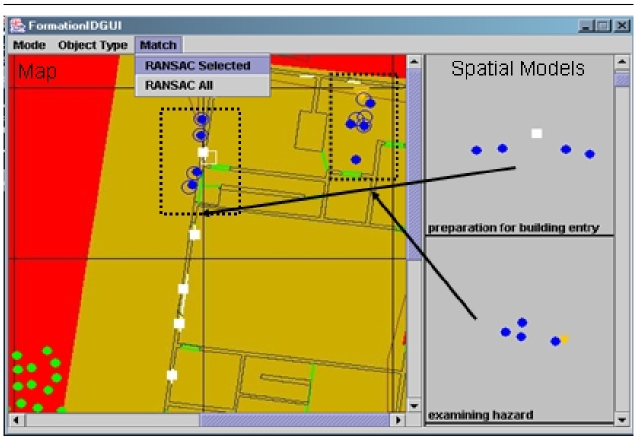

Figure 1: Spatial model authoring and matching system. The library of previously created spatial models are shown to the right of the screen; the left side of the GUI displays an annotated map to be analyzed. A fire team of soldiers (blue circles) examining a hazard (orange inverted triangle) are displayed 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. Note that the positions of the entities as predicted by the projection of the hazard examination model are not precisely localized, even though the model is correctly situated. A method for enhancing the position estimation is discussed in Section 5.4.

pear 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 a visibility constraint (see below).

entity type: For our library of MOUT behaviors, we designated eleven types of entities along with a relationship hierarchy which we use in scoring the compatibility between entity matches (See Section 4.3.2). Some of the types (e.g., objectives and hazards) do not refer to a specific physical type of object or area but are used to designate the role that the object plays in the world. Entity types include: person (unknown), civilian, teammate, opponent, hard cover, soft cover, empty area, windows, intersections, doorways, hazards, and objectives.

pairwise constraints between entities: For certain behaviors, lines of visibility (or lack of visibility) between entities are an important part of the spatial relationships. We model these visibility relationships as line segments between map entities that are either visibility preserving (cannot cross occlusions) or enforce lack of visibility between entities (must cross an occlusion).

scaling limitations: Certain models are only valid at a limited range of scales. For instance, a model representing a formation of foot soldiers would remain valid if the separation between soldiers was rescaled from 3m to 10m; however if the separation between soldiers increased to 1 km, the same model should no longer be valid (even if visibility constraints were satisfied).

4.2. Using Spatial Transforms to Generalize Models

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.

For this domain, 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:

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

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.

4.3. Robustly Matching Models to Maps

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 (Ballard & Brown, 1982). Each templates is applied to all possible locations in the map using a sliding window; the distance function is calculated over the window area and 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, RANSAC (Random Sampling and Consensus) (Fischler & Bolles, 1981), to efficiently sample the space of transforms using hypotheses generated from minimal sample sets of point correspondences. RANSAC 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 generated the observed data in the map. Our likelihood estimate is robust to missing data (entities in the model that are missing on the map) and to outliers.

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. Below, we describe the two phases of our algorithm in greater detail.

4.3.1. Hypothesis Generation

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 of 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 (Hartley & Zisserman, 2000) 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.

For this domain, we assume that models are valid for all rotations and translations but are limited to a valid range of scales. If the candidate point correspondences yield a transform outside the valid scale range, it is discarded before the hypothesis testing phase.

4.3.2. Hypothesis Testing

We score each sampled hypothesis as follows (illustrated in Figure 2). First 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. Additionally, we enforce pairwise constraints between entities, such as visibility and occlusion, as specified by the model. If the constraints are violated, we penalize the quality of the hypothesis.

For our library of building clearing MOUT behaviors, we designated eleven different types of entities and defined three compatibility functions. Entity types include: person (unknown), civilian, teammate, opponent, hard cover, soft cover, empty area, windows, intersections, doorways, hazards, and objectives. Compatibility functions are used to determine how matches between two entities of different types affect the quality score. We used three types of compatibility functions: 1) exact match—the type of the model entity and map entity must match exactly to contribute positively toward the quality score; 2) categorical match—the map entity and model entity have to belong to the same general category, e.g., person, cover, empty space; 3) functional

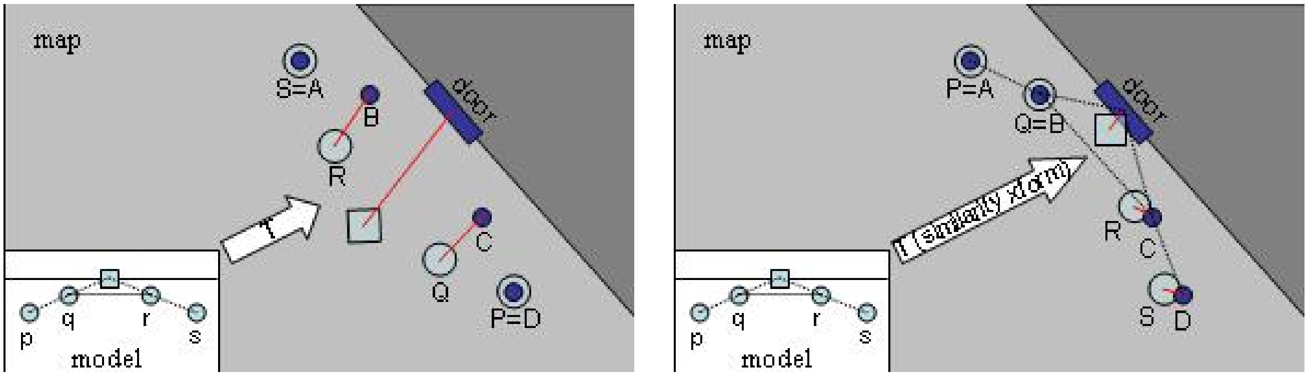

Figure 2: Examples of hypothesis generation and testing using RANSAC. For the building entry maneuver model shown here, we show two different sampled minimal correspondence sets. In (left), $\{P \mapsto D, S \mapsto A\}$, while in (right) $\{P \mapsto A, Q \mapsto B\}$. Given these correspondences, we derive the the transform, **T**. Applying **T** to each of the model entities, gives us their predicted locations on the map. Each predicted location casts a vote in favor of **T** if there exists a map entity of compatible type within range. The first hypothesis (left) receives no additional votes since the distances to compatible entities is too great. The second hypothesis (right) is consistent with the map and receives many votes. Pairwise constraints, shown as dotted lines, are then verified. In this case, they denote desired visibility constraints between MOUT soldiers, and we confirm that the second hypothesis does not violate them. Although not shown in these simple examples, RANSAC is robust to large numbers of outliers and to missing data.

match—the map entity has to serve the same functional purpose as the model entity. For our experiments, we specified that the random point correspondences used to generate candidate hypotheses must be drawn from the set of exact matches; the more relaxed categorical match criteria was used during to score the quality of the hypothesis.

Our current implementation supports two types of pairwise constraints, visibility and occlusion. A visibility constraint specifies that two entities on the map must have unobstructed line of sight (or fire), while the occlusion constraint specifies that no unobstructed line of sight should exist between those entities. Once a hypothesis has passed the initial screening stage, we can verify that the constraints are obeyed using standard line intersection algorithms against obstacles and entities on the map. This is shown on Figure 2 (right).

We apply RANSAC to all the models in the library and generate a set of hypotheses (transform plus model) whose normalized match quality exceeds a specified threshold. These hypotheses are all consistent with the observed data. If desired, the transforms for each of these hypotheses can be refined by applying standard least squares estimation techniques to the set of inliers for each hypothesis (Hartley & Zisserman, 2000). This is generally unnecessary for our application, except as discussed in Section 5.4.

5. Evaluation

We implemented our spatial matching technique as a Java application that supports authoring of model libraries and automated matching of spatial models to annotated 2D maps (see Figure 1). The models that we created for the MOUT building clearing scenario contained 3–15 entities and were matched to maps with approximately 100 annotations.

5.1. Physical Team Behaviors for Building Clearing

The maps employed in our evaluation are motivated by the building clearing scenarios described in (Phillips et al., 2001) and the physical behaviors given in (Rhodes, 1980). The physical team behaviors are: traverse street (stacked formation), bypass window, enter building, flank enemy position, examine hazards, clear room, cross intersection (L-shaped and T-shaped). We also model less structured physical behaviors such as enemy sniper placements and civilian crowds.

We examine the effectiveness of our spatial matching technique on the following three dimensions:

generality: how well do spatial models developed for one scenario generalize to scenarios with different layouts?

robustness: is our matching technique robust to noisy data, mislabeled entities, and occluded points?

threat prediction: how can we use our model to predict the location of occluded elements such as enemy snipers and missing team members?

Each of these points is discussed below in greater detail.

5.2. Generality

To examine how models developed for one spatial layout generalize to different layouts, we developed several maps that include instances of the flanking behavior. When executing the flanking behavior two members of the team fire at an enemy soldier to pin him down while the other two team members move to a location that offers a better line of fire. In the left panel of Figure 3, the original spatial model for the terminal position of the flanking behavior is shown. Our technique successfully identifies the correct instance of flanking in the center panel despite significant spatial differences and the existence of outlier entities. The model easily generalizes to instances with different scale and rotation because RANSAC can correctly identify (even in the presence of outliers) the similarity transform that accounts for these changes.

The third panel shows an example of flanking where the angle between the two friendly fire teams is significantly different from the model; in this scenario they have achieved a crossfire position on the enemy opponent by moving around a building. This instance of flanking fails to match since no single similarity transform applied to the entire spatial model can explain the observations on the map. This could be addressed in several ways. The simplest solution is to provide additional spatial models for this behavior to account for the diversity. Another approach is to employ a broader class of transforms, either global (affine or projective) or non-uniform (elastic graphs). The trade-off of employing a broader class of transforms is discussed in Section 6.

5.3. Robustness

Since each map contains multiple models in addition to extraneous entities that don't match any of the models, our matching technique must be able to ignore these outlier entities. In the maps we analyzed, about 95% of the entities were effectively "outliers", unrelated to the model under comparison. Fortunately RANSAC handles outliers very well by generating hypotheses only using minimal subsets; as the number of outliers increases, we can compensate by iterating more times as described by the formula given in Section 6.

Maps generated from the perceptual viewpoint of a synthetic character are incomplete, due to occlusion and limited sensor range. Our technique should be able to match models to partially populated maps as described in Section 5.4; to do that we must examine the role of the model's normalized quality score and how it relates to the match acceptance threshold. Currently, we set the normalized quality threshold at 0.75; effectively this means that approximately three-quarters of the points in the model must be successfully matched. In Section 5.4 we discuss strategies for setting this threshold on a individual model basis according to signal detection theory.

5.4. Threat Prediction

We can exploit our technique's robustness to missing data to predict the potential locations of unseen threats. A complete spatial model can match a partial set of entities on the map if the normalized sum of the existing votes is sufficiently high. For instance, given a spatial model describing commonly occurring enemy sniper vantage points, our technique can predict likely sniper positions even if the enemy forces themselves are not visible. This is useful in the case where the map is incomplete and reflects only the accumulated observations from one team's agents.

To predict the location of hidden entities, we simply project all the unmatched entities in the matching model to the map using the similarity transform; no additional computation is necessary since the transform was already computed during the hypothesis generation phase. However this initial transform was computed from a minimal set of two point correspondences. To produce a more refined estimate of the similarity transform, we can incorporate information from all the points matching the model to create an overconstrained system of equations that can be solved using standard least squares techniques. This additional effort is justified only if we need extremely accurate estimates of the positions of entities within the model.

Figure 4 shows a fire team of soldiers (denoted as blue circles) moving in a stacked formation toward the building entry. To the left hidden behind a clump of trees (denoted by the green rectangles) is an enemy sniper covering the building. Although the sniper itself is not visible, its likely position is detected (marked by the empty red circle) by a successful match of the model shown in the top right hand corner.

Applying spatial reasoning for threat prediction typically entails a trade-off; as the number of unmatched entities in the model increases we must lower the matching threshold to enable these partial matches to be detected. Increasing the

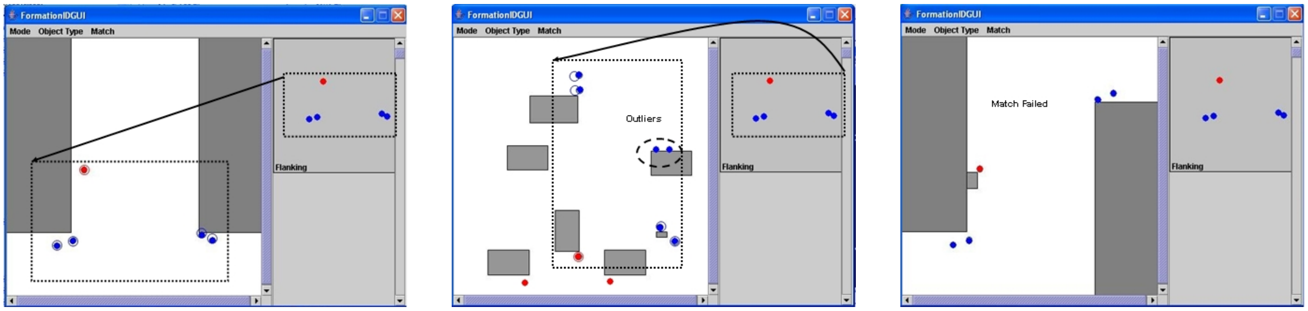

Figure 3: Three examples of team flanking behavior in different scenario layouts. The spatial flanking model (shown in the right hand side side of the GUI) was originally designed for the layout displayed in the left panel. The center panel shows the flanking behavior occurring in different spatial layout that includes more opponents and an additional fire team. Our technique successfully matches the flanking model to this new situation in spite of the differing spatial layout and the addition of extraneous entities (marked as outliers). The right panel shows an instance of the flanking behavior that fails to match the model because no single similarity transform applied to the entire spatial model can explain the observations on the map.

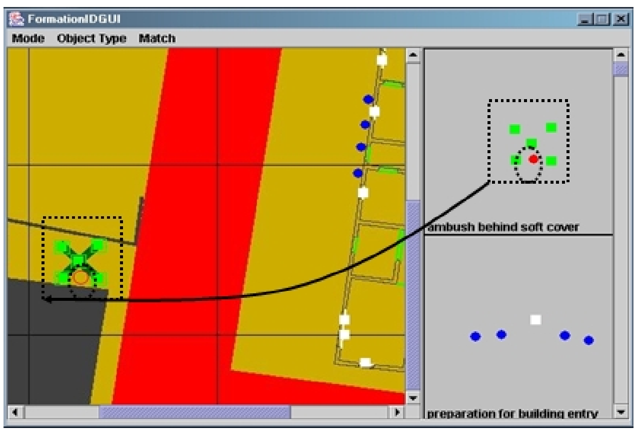

Figure 4: Using the best hypothesis match to predict unseen threat locations. A fire team of soldiers (denoted as blue circles) is shown moving in a stacked formation toward the building entry. To the left hidden behind a clump of trees (denoted by the green rectangles) is an enemy sniper covering the building. Although the sniper itself is not visible, its likely position is detected (marked by the empty red circle) by a successful match of the model shown in the top right hand corner. The transform found by RANSAC during the matching process is used to project all the entities of the spatial model to their hypothesized locations.

sensitivity in this manner also increases the number of false alarms (i.e., hallucinating threats where none exist). It may be useful to set matching thresholds on a model by model basis to correctly reflect the cost of false alarms vs. missed matches. For detecting enemy behaviors (snipers, enemies concealed in crowds) the matching threshold should be set lower to reflect the high cost of missing enemy threats.

6. Discussion

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:

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

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 which is computationally inexpensive, especially compared to exhaustive searching on a large map.

The number of iterations is relatively small due to the low value of s, the number of elements required to define the minimal set which completely specifies the spatial transform. If we expanded the class of allowable transforms, either to include 3D similarity transforms or a broader class of 2D transforms (e.g., affine), the number of elements required to specify the spatial transform would increase to $s = 3$ and the iterations required would increase to 36841.

7. Conclusion and Future Work

Our spatial matching technique is designed to be incorporated into a larger recognition system that recognizes high-level team behaviors from sequences of low-level movements. In addition to using spatial cues our system incorporates motion information from dynamic human entities by classifying frames of motion capture data (Sukthankar & Sycara, 2005). By developing composite spatial models for team behaviors, we avoid an exponential expansion of state space; instead of representing each team member's state separately we collapse the state of the team into a single spatial model representing the total state of the team. Currently our spatial model does not include a representation of how each behavior evolves over time; without this it is difficult to identify repetitive behaviors such as bounding overwatch that are defined by a combination of temporal and spatial relationships. In the future, we hope to extend our model to track team formations as they evolve over time.

Our current spatial model is robust to outliers, invariant to many types of spatial transforms, and can express a wide variety of static relationships and constraints between physical entities. For a given model and quality function, we can guarantee that 99% of the time RANSAC finds the best potential transform within 2000 iterations which can be executed in a fraction of a second.

Acknowledgements

The authors would like to thank Rahul Sukthankar for thoughtful proof-reading and commentary. This work was supported by ONR grant N0014-02-1-0438.

References

Ballard, D., & Brown, C. (1982). *Computer vision*. Prentice-Hall.

Best, B., & Lebiere, C. (2003). Spatial plans, communication, and teamwork in synthetic MOUT agents. In *Proceedings of Behavior Representation in Modeling and Simulation Conference (BRIMS)*.

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

Hartley, R., & Zisserman, A. (2000). *Multiple view geometry in computer vision*. Cambridge University Press.

Livak, T. (2004). *Collaborative warrior tutoring*. Unpublished master's thesis, Worcester Polytechnic Institute.

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

Phillips, J., McCloskey, M., McDermott, P., Wiggins, S., & Battaglia, D. (2001). *Decision-centered MOUT training for small unit leaders* (Tech. Rep. No. 1776). U.S. Army Research Institute for Behavioral and Social Sciences.

Pynadath, D. (1999). *Probabilistic grammars for plan recognition*. Unpublished doctoral dissertation, University of Michigan.

Rhodes, J. (1980). *Military Operations on Urbanized Terrain (MOUT)* (No. MCWP 3-35.3).

Sukthankar, G., Mandel, M., Sycara, K., & Hodgins, J. K. (2004). Modeling physical variability for synthetic MOUT agents. In *Proceedings of Behavior Representation in Modeling and Simulation Conference (BRIMS)*.

Sukthankar, G., & Sycara, K. (2005). A cost minimization approach to human behavior recognition. In *Proceedings of Autonomous Agents and Multi-Agent Systems (AAMAS)*.

Author Biographies

GITA SUKTHANKAR is a doctoral candidate in the Robotics Ph.D. program at Carnegie Mellon University. She holds an A.B. in psychology from Princeton University and an M.S. in Robotics from CMU. She resumed her Ph.D. studies in 2003 after a 2 year stint of working as a researcher with the Compaq Cambridge Research Lab in the mobile computing group. Her research interests include teamwork, planning, and simulated game environments.

DR. KATIA SYCARA is a Research Professor in the School of Computer Science at Carnegie Mellon University and Director of the Advanced Technology Lab. She has been the PI for many multimillion-dollar grants from DARPA, AFOSR, ARDA, ONR, NASA, NSF and industry. Dr. Sycara was General Chair of Autonomous Agents 1998, founding editor-in-chief of the Journal of Autonomous Agents and Multiagent Systems, and is on the editorial board of four other journals. She has authored over 200 technical papers and book chapters on multi-agent systems. Dr. Sycara is a AAAI Fellow and recipient of the ACM Agents Research Award.