Submitted to AAMAS 2003

Locating Moving Entities in Dynamic Indoor Environments with Teams of Mobile Robots

Matthew Rosencrantz, Geoffrey Gordon, and Sebastian Thrun
Carnegie Mellon University
Pittsburgh, PA

ABSTRACT

This article presents an implemented multi-robot system for playing the popular game of laser tag. The object of the game is to search for and tag opponents that can move freely about the environment. The main contribution of this paper is a new variable-dimension particle filter algorithm for tracking the location of opponents under prolonged periods of occlusion. This algorithm can cope efficiently with variable numbers of opponents, through mechanisms that dynamically increase and decrease the number of particle tracks. When searching for opponents, the individual agents greedily maximize their information gain, using a negotiation technique for coordinating their search efforts. Experimental results are provided, obtained with a physical robot system in large-scale indoor environments.

Figure 1: The laser tag robots are equipped with light-emitting devices and receivers.

1. INTRODUCTION

This paper describes a multi-robot system capable of locating and pursuing moving actors (people, other robots) in dynamic environments. Our research is motivated by the popular game "laser tag" [27]. The object of laser tag is to find and tag individuals from an opposing team using a light-emitting tagging device. The robotic form of laser tag replaces people with robots: it is a game involving a team of robots searching though a dynamic environment and tagging opposing robots using beams of light. In our implementation, it is played in regular buildings, using teams of Pioneer-size robots equipped with laser range sensors (see Figure 1).

The game of laser tag provides unique opportunities for multi-agent robotics research. Just like robotic soccer [12], laser tag is inherently real-time. The environment is dynamic, leaving only limited time for information fusion and decision making. A key difference to robotic soccer arises from the pervasive presence of occlusion: most of the time, individuals are *not* in a robot's sensor range. To play well, the robots must carefully keep track of where possible opponents might be, and which areas of the environment have already been cleared. This creates a challenging multi-robot data fusion problem. Furthermore, the robots have to coordinate their actions in the face of massive uncertainty. This is reminiscent of the well-known robot exploration problem, but in laser tag it involves an ever-changing situation due to the fact that the environment is dynamic. We conjecture that the resulting research issues are characteristic for a much broader range of multi-robot application domains, ranging from personal service robots (e.g., tour-guide robots [4, 22, 32]) to the coordination of unmanned air or underwater vehicles in reconnaissance missions.

Multi-robot information gathering has been addressed by many researchers. The classical setting involves a team of robots locating stationary objects in an unknown environment [35, 10, 16, 23]. Most existing work in this field involves behavior-based strategies, in which the search is carried out through randomized motion. Coordination is often achieved through behaviors that maximize the distance between adjacent robots. Research in this field has predominantly focused on static environments [1], although some notable success has been reported for environments with dynamic objects [29]. However, randomized search is limited in that it relies on chance to find objects. Techniques that maintain environment models during search have been studied extensively in the field of multi-robot mapping [5, 20, 31]. These approaches apply to static environments in which objects do not move. The tracking literature, on the other hand, has long addressed the issue of tracking moving objects [3, 26, 15]. This work has recently been extended to mobile robots [11, 13, 18, 30] and distributed sensor systems [20, 24]. While these approaches work well in cases where the object of interest is within sensor reach (with possible brief periods of occlusion), they do not address the type of long-term occlusions found in the laser tag game. As we will show below, most of the bookkeeping in laser tag involves reasoning about places where opponents may *not* be, and opportunities to track opponents are rare.

This paper describes an implemented multi-robot system for actively locating moving actors in dynamic indoor environments, in the context of multi-robot laser tag. The core of our research is a new algorithm that enables a robot team to calculate posterior distributions over possible places where actors from the opposing team might be. To account for the immense occlusion and the resulting multi-modal nature of posterior distributions, our approach utilizes particle filters [7, 14] for representing the robots' beliefs. Our approach specifically addresses situations in which the number of actors in the environment is unknown. We propose a new particle filter algorithm that can effectively calculate posteriors for arbitrary number of actors. This algorithm builds on the insight that, even in environments with very large numbers of actors, many posterior distributions are practically equivalent. Hence, we can represent large numbers of such posteriors efficiently. This insight leads to the development of a variable dimension particle filter which dynamically generates new particle tracks when individual actors are observed, and merges old ones when the corresponding posteriors are (approximately) equivalent. The equivalence classes correspond to *roles* of the opponents. Roles represent the known characteristics of a set of opponents, ignoring their individual identities. As a result, our approach enables teams of robots to maintain complex distributions over any number of actors, under extended periods of occlusion, and in a computationally efficient manner.

When playing laser tag, our system employs a greedy technique for searching the environment, similar to techniques developed for multi-robot exploration [5, 31, 34]. Our approach maximizes the information gathered by the individual robots. The coordination of multiple robots is achieved through a negotiation technique that enables nearby agents to partition the space that is being searched. The approach has been implemented in simulation and using a physical robot system. This paper reports systematic scaling results obtained via simulation, along with actual robot results obtained in several indoor environment of different shapes and sizes. Our experiments demonstrate that our approach leads to systematic environment exploration and reliable localization of moving targets, even under prolonged periods of occlusion. Comparative experiments are provided which illustrate that the use of our new particle filter significantly improves overall system performance when compared to a memoryless multi-robot system.

2. TRACKING MULTIPLE ENTITIES UNDER MASSIVE OCCLUSION

2.1 State Representation

The main contribution of this paper is a new variable-dimension particle filter algorithm for tracking the location of opponents under prolonged periods of occlusion. Let $y^t = y_1, y_2, \ldots, y_t$ denote the robots' measurements up to time t, and u^t their control inputs. In our implemented system, sensor measurements consist of range readings at a known set of bearings from each robot produced by SICK laser range finders, but with appropriate modifications to the sensor model the readings could come from other perception systems such as vision or sonar. We will write y_{it} for the ith individual reading.

As is common in the literature on tracking [3], our approach maintains a posterior probability distribution over the state x_t at time t: $P(x_t \mid y^t, u^t)$. For the laser tag problem, the most important state variables are the positions of all of the agents. These variables can be divided into two qualitatively different sets: the positions of robots on our own team and the positions of agents on the opposing team. Mobile robot localization for robots under our control has been studied extensively in the literature [9]. The problem of estimating the state of the opposing agents in situations with prolonged periods of occlusion, on the other hand, has received nearly no attention.

Maintaining an estimate of the opposing robot positions can be difficult and computationally expensive. This is because of complicated, high-dimensional, multi-modal dependencies between state variables. The main cause of these problems is that our robots' sensor views of other actors are constantly occluded. This occlusion makes it difficult to determine which opponent is which, and it can also put sharp edges and multiple modes into our posterior distributions due to our inability to see around corners and through walls. Such situations are clearly outside the realm of the existing tracking literature.

The presence of prolonged occlusion induces a difficult *data association problem*. Any individual range measurement y_{it} could in principle be caused by a return from any object in our world: the map, a teammate, or an opposing robot. Data association decisions of this type are particularly difficult to make in laser tag because of the massive occlusion periods, in which individual opponents' locations cannot be determined accurately. To cope with this problem, our approach introduces new (hidden) state variables z_t to represent the data association hypotheses we are considering. Intuitively, these variables will encode pieces of information such as "sensor reading y_{it} represents a return off of either opponent 1 or opponent 3." In this way, our approach can make rational data association decisions even if individual agents are occluded for long periods of time.

More formally, our approach divides the state x_t into three pieces (r_t, s_t, z_t). The first piece, r_t, contains the positions of the robots on our own team. We will write $r_{1t}, r_{2t}, \ldots$ for the individual robot positions. The second piece, s_t (divided similarly into $s_{1t}, \ldots$), contains the positions of the robots on the opposing team. Finally, z_t contains the data association history; we will never represent z_t explicitly, but we will need to reason about it in the derivation of the state-tracking equations below. We will separate the data association history into $z_{t,\mathrm{map}}$ (representing decisions about whether to associate sensor readings with static objects in our map) and $z_{t,\mathrm{other}}$ (representing all other data association decisions).

With the above notation, our tracking problem is to estimate the posterior

$$P(r_t, s_t, z_t \mid y^t, u^t)$$

In order to make tracking computationally feasible, we will work through a series of three simplifications which separate out various tractable pieces of the problem. These steps are *conditioning on teammates*, *role-based data association*, and *factoring observation and motion models*. We will describe each step in detail below. The process of performing these analytic transformations to make the Monte-Carlo estimation process easier is called *Rao-Blackwellization* [8].

2.2 Conditioning On Teammates

The problem of belief tracking is significantly simplified by factoring the posterior probability

$$\begin{aligned} P(r_t, s_t, z_t \mid y^t, u^t) \quad &= \quad P(r_t, z_{t,\mathrm{map}} \mid y^t, u^t) \\ &\quad\; P(s_t, z_{t,\mathrm{other}} \mid r_t, z_{t,\mathrm{map}}, y^t, u^t) \end{aligned} \tag{1}$$

This factorization is a simple application of Bayes' rule. It separates out the problem of localizing our robots from the problem of finding and tracking actors on the opposing team. This is an important simplification, because, as noted above, the self-localization problem has been studied extensively [9].

In our domain, a particle-filter-based localization technique [17, 33] nearly always produces unimodal, high-accuracy position estimates for robots on our own team, so long as we have a reasonable idea of their initial positions. This motivates us to approximate the position estimates of our robots by point estimators located at the most likely position r_{jt} for each j. Such an approximation is only appropriate when the uncertainty for the positions of our robots is small, as it precludes us from using events such as robots observing each other for the self-localization part of our tracking problem [9]. In practice the uncertainty of our own robots' positions is usually extremely small, so this approximation is valid.

2.3 Role-Based Data Association

We have now reduced belief tracking to the problem of estimating

$$P(s_t, z_{t,\mathrm{other}} \mid r_t, z_{t,\mathrm{map}}, y^t, u^t) .$$

We know the observations y^t and controls u^t, and the robot positions r_t and map associations $z_{t,\mathrm{map}}$ are given by our robots' self-localizers. For further simplification, we now move to a role-based representation of data association.

Data association uncertainty causes two unpleasant effects in our posterior distribution, as illustrated in Figure 2a. First, each plausible data association hypothesis can cause a separate mode in our belief for the location of opponent actors. Because the number of plausible data associations often grows exponentially with time, the result is a highly multi-modal belief. Second, these modes contain cross-robot correlations: if we are confused whether a particular object we are tracking is opponent 2 or opponent 3, then finding out (through some other evidence) that it is opponent 2 causes us to believe opponent 3 is elsewhere. This is a common characteristic of data association problems [3].

In many tracking problems, it suffices to keep track of the single most probable data association hypothesis. For example, Dissanayake and colleagues [6] describe a Kalman filter tracker which uses maximum-likelihood data association to keep track of hundreds of landmarks. In their problem, however, landmarks are fairly well separated compared to the amount of uncertainty in their positions, which means that the most likely data association usually accounts for most of the posterior probability mass.

In the laser tag problem, unfortunately, we are not so lucky. Because opponent actors may be occluded for sizable intervals, their uncertainty areas can grow quite large and will quickly begin to overlap significantly with each other. For example, our sensor log might show two opponents entering a room and, after a pause, one opponent exiting the room. In this case, we cannot determine the true identity of the actor exiting the room; all we know is that it may be one of the two which went in, and that at least one other must still be inside the room.

This problem has important ramifications. If we were to attempt to keep track of the exact posterior distribution for all actors, we would quickly be overwhelmed with the number of plausible data association hypotheses. On the other hand, keeping track of only the most likely hypothesis would result in suboptimal behavior: for example, we might accidentally relabel two opponents as each other. If we have already tagged one of the two, that mistake will cause us to let the other pass freely. Examples such as this one demonstrate how difficult it is to track multiple agents under massive occlusion. To our knowledge, existing tracking and data association methods are unable to cope with such situations.

To avoid this problem, we will introduce a set of intermediate variables $\bar{s}_t$. These variables represent opponent positions just like s_t, but they are indexed differently. Instead of indexing by opponent label, our approach indexes $\bar{s}_t$ by *role*. *Roles* represent classes of opponents that share similar posterior distributions. For example, $\bar{s}_{1t}$ might represent the position of the opponent we just saw come out of a room, without reference to its exact identity. The definition of roles makes it possible to track multiple opponents under severe data association problems arising from the nature of the occlusions. In particular, we can then allocate or merge roles dynamically as we gain or lose information about the locations of enemy robots.

This sort of role-based (also called *deictic* [2]) representation allows us to collapse together many different symmetric modes of our posterior distribution, as indicated in Figure 2b. In particular, roles enable us to overcome the the data association problem: we can define the roles so that we are certain which role should be associated with each sensor reading, regardless of the identity of a specific actor. That means that that there will be no correlation between the positions of different roles, and so we can factor our belief and track each role separately.[1]

The new role-based representation requires us to maintain a belief about which roles might map to which individual actors (or more importantly, which properties such as "tagged" or "hostile" might be associated to each role). For this purpose, our algorithm maintains a list of possible identities and property-lists for every role which it creates. This list is called an *identity uncertainty set* or *id-set*. We will write d_{jt} for the id-set of the jth role at time t. For example, an id-set might say "This role represents 2 robots. Their identities are in the set $\{1, 3, 4, 6\}$, they definitely have the property 'hostile,' and they might or might not have the property 'tagged.' "

The role-based representation has an important impact on the posterior estimation problem. Mathematically, it allows us to simplify our belief as follows:

$$\begin{aligned} & P(s_t, z_{t,\mathrm{other}} \mid r_t, z_{t,\mathrm{map}}, y^t, u^t) \\ &= P(s_t \mid \bar{s}_t, d_t)\, P(\bar{s}_t, d_t, z_{t,\mathrm{other}} \mid r_t, z_{t,\mathrm{map}}, y^t, u^t) \\ &= P(s_t \mid \bar{s}_t, d_t)\, P(\bar{s}_t, d_t, \mid r_t, z_t, y^t, u^t) \qquad \text{for ML } z_{t,\mathrm{other}} \\ &= P(s_t \mid \bar{s}_t, d_t) \prod_{\mathrm{roles}\ j} P(\bar{s}_{jt}, d_{jt} \mid r_t, z_t, y^t, u^t) \qquad (2) \end{aligned}$$

The first equality holds because $(\bar{s}_t, d_t)$ is a sufficient statistic containing all of the information we have about the opposing team positions s_t; in other words, $(\bar{s}_t, d_t)$ d-separates [25] s_t from the remaining variables. The second equality holds when we set $z_{t,\mathrm{other}}$ to its maximum likelihood value because, by definition, we know the data association perfectly if we express it in terms of roles. The third equality holds because we have chosen our roles so that their positions are independent. The advantage of this factorization lies in the fact that it makes tracking highly efficient. In contract, the common formulation of the tracking problem requires us to maintain exponentially many hypotheses, which clearly is impractical in the presence of many opponent actors.

2.4 Factoring Sensor and Motion Models

Our final simplification is to factor our motion and observation models. With joint motion and observation models, the positions of different roles might become dependent whenever we moved or observed, destroying the factorization which we obtained in the previous steps. This is *not* the case for motion and observation models that model each role independently. As discussed in this section,

[1] In practice, we rarely need to worry about confusion of the sort illustrated in Figure 2. Much more frequent is the type of confusion illustrated in Figure 3 below, in which two enemy robots leave our sensor range in the same direction and quickly become indistinguishable. But the same principle applies in either case: if we try to track the robots individually we wind up with multi-modal posteriors and global correlations, while if we reason about roles as described in this section the posterior becomes much easier to track.

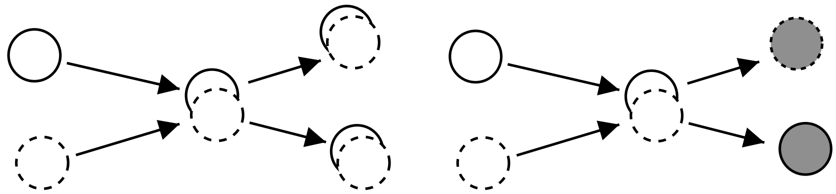

Figure 2: (a) We can confuse two tracked objects if they move close together and then apart. This confusion manifests in two ways: multi-modal distributions for each object, and negative correlation between the positions of the two objects. (b) If we allocate new roles to represent "the robot which went left" and "the robot which went right," we have no data association problem and only unimodal posterior beliefs.

both the motion model and the measurement model are indeed independent for different actors, validating the factorization proposed in the previous section.

The independence of the motion model for teammates is straightforward. Each of our robots moves according to local motion commands; the noise in motion is independent for each of the robots. Similarly, it is natural to assume that the opposing agents move independently of each other, with independent random variables characterizing their next state transition functions. Clearly, the latter independence assumption might not actually hold: for example, it would be violated if we knew that opponents tended to move in groups. In practice, though, assuming independence in opponent motion is safe in that it establishes a *worst case* motion model.

Independence also holds for measurements, as discussed in length in [19]. In particular, there are two types of information we can get from our sensors, *positive* and *negative*. Positive information tells us where an opponent actor *is*; we receive positive information by associating a sensor reading i to a role j. Negative information tells us where an opponent *isn't*; we receive negative information when our sensor beams pass through a space without detecting an opponent. Negative information produces complicated posteriors because our visibility region is distorted by occlusion from static obstacles in the map. Most position-tracking systems can only deal with positive information [3]. If the tracked objects are in view most of the time, positive information is usually sufficient to achieve good tracking. In laser tag, however, the majority of the information we receive is negative, and so it is critical to take advantage of negative information in our tracking system.

Because opponents are often massively occluded, our approach needs to take advantage of negative information to keep a reasonable idea of where opponents might be. For example, if our system observes an opponent enter a room with only one exit, it is imperative to keep track of the fact that it remains in the room until our robots see it leave (or until our robots lose sight of the exit). Negative information tends to produce multi-modal beliefs because our sensor-coverage area can cut a region out of the middle of a belief distribution. In particular, negative information can increase the variance of our belief distribution (although it always reduces entropy).

It is well known that we can approximately factor the problem of tracking several objects using positive information into several independent tracking problems with one object each [26]. For our system, though, we make the novel observation that we can also factor the effects of negative information. The observations enter into the tracking problem through the observation likelihood $P(y_t \mid r_t, s_t, z_t)$, or equivalently $P(y_t \mid r_t, \bar{s}_t, z_t)$. Since each observation is conditionally independent once we know positions and data associations, we have

$$P(y_t \mid r_t, \bar{s}_t, z_t) = \prod_i P(y_{it} \mid r_t, \bar{s}_t, z_t)$$

Now, suppose observation i is associated to role k. In this case, the probability of observation i is a product of two terms: the probability of generating y_{it} given that the sensor beam reached role k, and the probability that the beam reached role k. The latter term in turn factors into the probability that the beam passed through role 1 without being intercepted, times the probability that the beam passed through role 2 without being intercepted, and so forth for all roles $j \neq k$. In other words, we can write

$$P(y_t \mid r_t, \bar{s}_t, z_t) = \prod_i \left[P(y_{it} \mid r_t, \bar{s}_{kt}, \text{hit}) \prod_{j \neq k} P(y_{it} \mid r_t, \bar{s}_{jt}, \text{no-hit}) \right] \tag{3}$$

which is factorized so that each term depends only on the belief for only a single role. Thus, sensor measurements maintain the conditional independence of our role estimates, and so our factorized representation remains valid as we incorporate sensor information.

2.5 Summary of Tracking Algorithm

Combining equations (1) and (2), our final factorization of the belief state is

$$P(r_t, s_t, z_t \mid y^t, u^t) = P(r_t, z_{t,\mathrm{map}} \mid y^t, u^t) P(s_t \mid \bar{s}_t, d_t) \prod_{\text{roles } j} P(\bar{s}_{jt}, d_{jt} \mid r_t, z_t, y^t, u^t) \tag{4}$$

Equation (3) shows that this factorization is preserved across evidence updates, and a similar argument shows that it is preserved across motion updates.

In contrast to the naive approaches of tracking each opponent separately or all opponents together—neither of which is computationally feasible for problems with massive occlusion—our factorization represents an efficient and accurate way to compute our belief state after any sequence of observations and actions. The reason for this efficiency is that we have separated the overall belief-tracking problem into a number of smaller tracking problems, one for each factor of equation (4).

2.6 Details of Tracking Algorithm

For completeness, we now review the details of our approximations to the terms of equation (4), as well as our strategy for splitting and merging roles. Our approach approximates the first term

$P(r_t, z_{t,\text{map}} \mid y^t, u^t)$ in equation (4) with a vector of maximum-likelihood estimates returned by individual self-localizers, one for each team member. The second term $P(s_t \mid \bar{s}_t, d_t)$ corresponds to the information contained in the id-sets of our roles. If decisions to split and merge roles were always correct, our representation of this term would be exact; but in practice we only keep track of an approximation since our decisions are based on statistical tests (described below). Finally, we represent each term inside the product with a separate particle filter tracker. This representation is inexact since particle filters are Monte Carlo algorithms, but it is asymptotically correct as we increase the number of particles. Since each individual particle filter only has to track a single role, we have observed that we do not need too many particles in practice to get good tracking performance.

In order to fully specify our tracking algorithm, the only remaining step is to describe how we allocate and merge roles in order to maintain independence. We begin with a single role which represents all of the robots on the opposing team. This role's track count is set to an upper bound on the number of opposing robots, and its position is initialized to a uniform distribution over free space. As we receive new observations, there are two types of decisions we must make: when to split a role into two, and when to merge two roles into one.

Role splitting is driven by sensor observations: if an observation's maximum likelihood association is to a role j that has a track count higher than 1, our approach splits off a copy of role j (call this copy k). Role k starts out identical to role j in every way except one: its position distribution is the same and its id-set is the same, but its track count is always exactly 1. (We of course must also decrement j's track count by 1 to maintain consistency.) Now we can incorporate the new observation into role k. When doing so, k's position uncertainty will become smaller than j's. In this way, future observations in similar locations will tend to associate with k instead of j, causing role k to track the newly-observed robot while j represents the position of the remaining unseen robots.

Role merging is a more expensive operation. After every T_{merge} time steps, our approach checks each role j against each other role k to see if their position distributions are very similar. If they are, we can no longer tell the two roles apart, and so we can merge them into one. This test is quadratic in the number of roles, but since it tends to keep the number of roles small the expense is usually worthwhile. (In any case it is much lower than the exponential costs associated with tracking multiple modes for each individual opponent.)

For our measure of similarity of distributions, we use a grid-based Jensen-Shannon divergence. This measure places a coarse grid over the map and estimates the probabilities p_{ij} and p_{ik} that roles j and k assign to each grid cell i (using Laplace smoothing to ensure that no grid cell is assigned zero probability). Based on the frequency counts in this grid, our approach calculates the Jensen-Shannon divergence (or symmetric KL-divergence)

$$D_{JS}(p_j, p_k) = \sum_i \left[p_{ij} \ln \frac{p_{ij}}{p_{ik}} + p_{ik} \ln \frac{p_{ik}}{p_{ij}} \right]$$

between two different role posteriors. If this divergence is smaller than a cutoff, the roles j and k are merged into a single role. In the laser tag domain, the grid is two-dimensional, just like the maps of the environment.

When we merge two roles, the merged position distribution becomes the average of the two original position distributions before the merging operation (which is a good approximation since these two distributions were just determined to be nearly the same). The track count becomes the sum of the two original track counts. And, the merged id-set is the union of the two original id-sets, representing the fact that we no longer know the difference between the two roles.

3. PLANNING

The final, critical ingredient in the laser tag domain is the actual planning and coordination of the robots. Motion planning is implemented through standard path planning algorithms [21]. For multi-robot coordination, our system uses a heuristic approach that attempts to maximize information gain and minimize search time, similar to the one proposed by Burgard et al. [5] in the context of multi-robot mapping of static environments.

The coordination strategy works as follows. Every T_{replan} seconds each of the robots chooses a point to move toward. These destination points are chosen from a coarse three-dimensional grid, (x, y, θ), of points laid over the map. Here θ is the robot orientation, which is important because each robot's laser sensor covers only the area in front of it. In order to choose its destination, each robot computes a score for every grid point, with higher scores corresponding to more desirable locations.

The score for a destination point is based on the number of hypothesized enemy robot positions which are within sensor range of it. So, the robots will prefer to go to places where they are likely to eliminate many possible hypotheses from their belief state and therefore reduce their belief entropy. The score for eliminating a hypothesis is downweighted multiplicatively by its distance and angular offset from the destination point. The scores are further multiplicatively downweighted by an estimate of the cost to reach their associated destination point. The relative influences of all these factors—cost, distance, and angular distance—are controlled by a set of parameters that were chosen experimentally.

To resolve conflicts between robots, we assign the robots a fixed precedence ordering. After the first robot chooses a destination point, the scores of all nearby points within line of sight of the chosen point are reduced. The second robot then chooses a point, and due to the down-weighting of scores it is unlikely to choose a point near the one chosen by the first robot. This process continues for each robot. Coordination strategies like this one have been found to be highly effective in the context of coordinated multi-robot exploration of static environments [5, 31, 34]. Our approach extends these strategies to dynamic environments, where the target entities (opponents) may move.

4. EXPERIMENTAL RESULTS

In order to better understand the strengths of our system we performed a number of experiments, both in simulation and on real robots. First, we evaluated the utility of dynamically merging roles and the impact of that feature on scaling performance. Second, we systematically explored the impact of our tracking algorithm on our ability to find and tag opponents in the laser tag setting. Finally, we ascertained that our system performs well in large scale real world environments on real robots. In each of these experiments we found that our proposed system exhibits excellent performance. The following sections will treat each experiment in detail.

4.1 Split-Merge

In order to demonstrate the effectiveness of the split-merge feature we placed, in simulation, an observer robot in a doorway looking into a hall. We then moved a sequence of target robots through the hallway past the door at random intervals. The setting of the experiment is shown in Figure 3: the light-filled bold circle represents the observer robot, with a tic-mark showing orientation, and

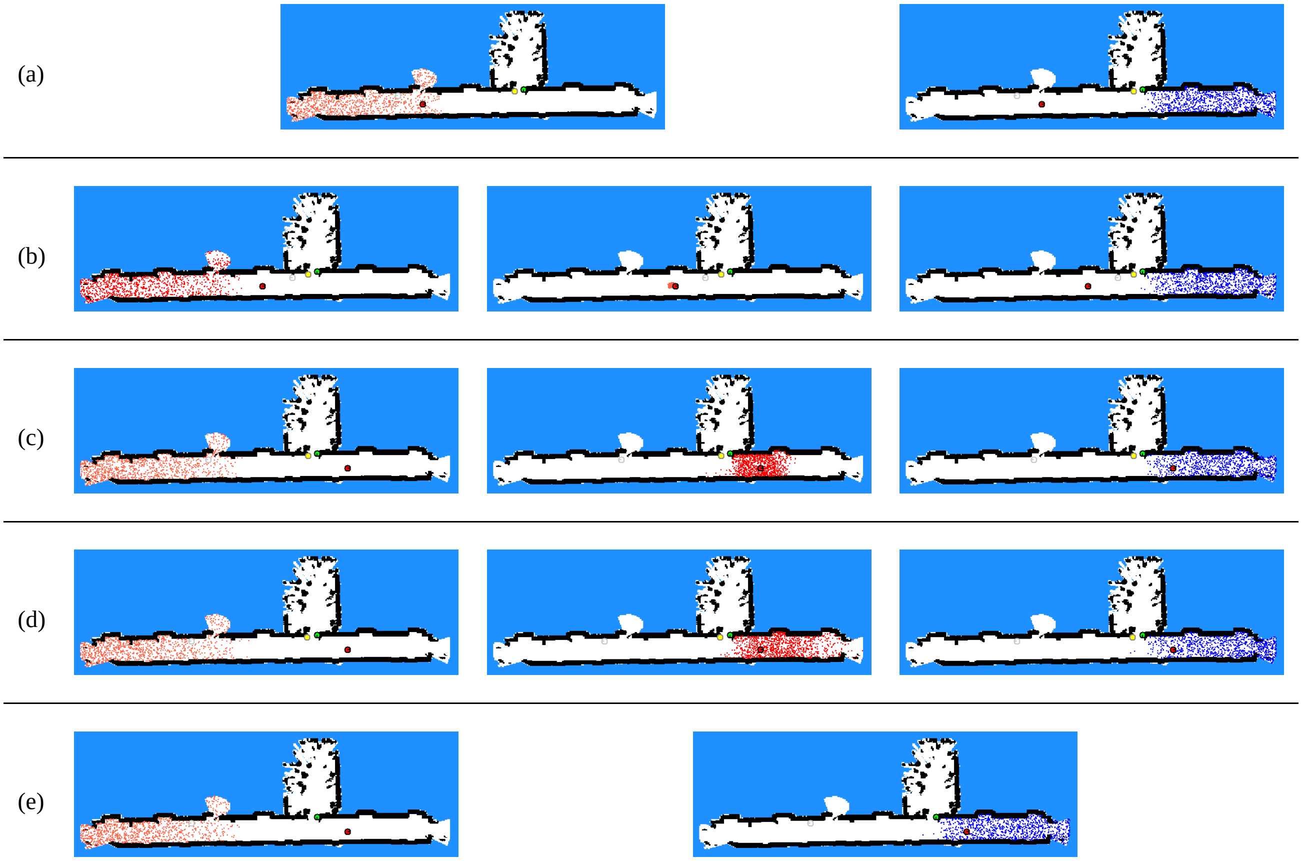

Figure 3: Each row represents the state of all the filters in the system at a given time. (a) Before splitting there are only two filters. (b) Just after splitting there are three filters. (c) The robot has moved down the hall, out of sight of the observer and the distribution that was assigned to it has begun to spread. (d) Distribution 2 is becoming similar to distribution 3. (e) Distributions 2 and 3 were deemed similar enough to merge.

the dark-filled bold circle is the moving robot. Each row of the figure represents the state of all active filters in the system at different times during a single pass of the moving robot. In this example the merge feature is being used, so the number of active filters can decrease over time. If the merge feature is disabled, the number of active filters is strictly increasing.

To understand the usefulness of roles in tracking, we ran the system with and without the merge feature. In each case we ran the entire experiment 25 times, with a single run consisting of approximately 20 target robots moving in sequence past the observer robot. Figure 4 shows the average number of active particle filters in the system in both cases. Without merging, the system keeps track of all the robots independently, despite the fact that their distributions are virtually identical. Thus, the number of filters grows linearly with time. With merging, however, the number of filters remains essentially constant as similar distributions are merged together.

4.2 Tracking

We furthermore explored the utility of our tracking system in the laser tag domain by comparing it to a baseline system that does not take advantage of state estimation information. This baseline system is similar to our original system in that it coordinates the pursuers by directing them to choose destination points that put them out of sight of each other. However, unlike our system, the destination points are chosen randomly except for this coordination criterion. We chose this strategy to represent the class of behavior-based strategies which do not attempt to track the true posterior belief about where the enemies might be. We believe that, within the class of memoryless strategies, this baseline strategy is a good performer; and, since it is similar in architecture to our laser tag system, it provides a good basis for comparison.

We compared these two strategies over 100 runs in simulation. A single run consists of all robots (two pursuers and one evader) starting at random positions, and the pursuing robots searching until the evading robot is tagged. If our solution, with tracking, allows us to tag the evader more quickly than the baseline system can, we can conclude that our tracking algorithm helps the pursuers find the evader. If the tracking approach does not yield superior performance then we will conclude that state estimation is not useful in this problem, or that the environment we are searching is too small and simple, or that the planner we are using is not capable of taking advantage of the tracking information. Figure 5 plots the probability of capture versus time for the two approaches. Our system consistently outperforms the baseline, demonstrating the value of our state tracking algorithm.

4.3 Complete System

Figure 6 shows a run of the complete system on physical robots. There are two pursuing robots, a Pioneer I and a Pioneer II, and one evading robot, built on a modified Scout base. Each robot carries a Pentium-class computer which is used to run components of the Carmen software suite [17]. These components provide each robot

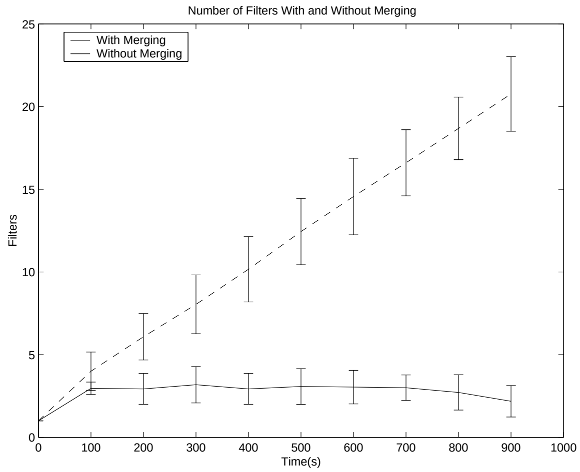

Figure 4: In this experiment an observer watches another robot move through a hall. Merging allows us to maintain a near constant number of filters, since there are only a constant number of beliefs in this system. Without merging the number of filters grows linearly.

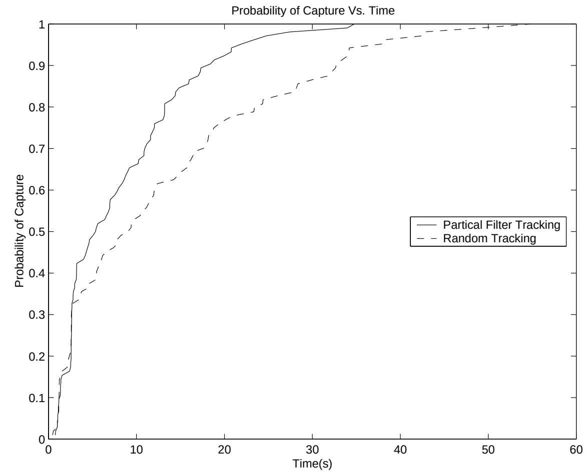

Figure 5: The state tracking in our system allows it to find the enemy robot consistently faster than a baseline system without the benefit of tracking.

with localization and point-to-point navigation in prebuilt maps.

In figure 6, panel (a) shows the initial conditions. The two pursuing robots begin near the center of the hallway. They are represented by filled circles with a dark outline and a radial line indicating orientation. The position of the evading robot is shown at the far left of the hall (also a filled circle, but with a slightly different shading). The position of this robot is not used by the pursuers for planning, but only for distinguishing the enemy robots from ancillary objects, such as people, that might walk by during an experiment. In (b), the robots begin moving toward the target locations selected by the coordination technique (represented as unfilled circles with a light outline and a radial line to indicate orientation). Note that they have chosen to move in separate directions and have also cleared out most of the hypotheses in the room above the corridor. Moving to (c), the right end of the hall has been nearly cleared of hypotheses, meaning it is very unlikely the enemy is hiding there, so both robots decide to move toward the left of the corridor. In (d) the opponent robot is finally located and tagged by one of the pursuers. At this point the robots label the opponent as having been tagged and ignore it in their planning, though they still track it to avoid confusing it with another robot. In (f) and (g) the pursuers continue to search the map since they have been told that there are potentially two targets in the world and they have only tagged one.

We have run our system in several environments besides the one detailed in the case above. Figure 7 shows belief maps from two other environments, as well as a photo of the robots in the environment described above. In all these environments the system performed well, directing the robots to search through the map efficiently until they found the target. During these runs the system displayed intelligent behaviors such as splitting the robots up to explore branching corridors and keeping the robots together to cover wide areas more efficiently without letting the opponent past.

5. DISCUSSION

We have introduced a system for playing multi-robot laser tag. Laser tag presents an interesting opportunity for research because of its real-time, dynamic nature and because it leads to complex non-Gaussian beliefs over the state of the world. Our system allows multiple robots to search in a coordinated way for arbitrary numbers of opponents. This is accomplished by using a variable dimension particle filter which is the primary contribution of this paper. The filter we have introduced is able to track large numbers of opponents by mapping them into a small set of equivalence classes or roles.

Our research opens up many opportunities for future work. An avenue of particular interest is more sophisticated planning. One possible approach to this problem is to use belief compression [28] to compactly represent the belief state and use MDP planning in the compressed space. Another direction to explore is creating a more sophisticated opponent model to increase performance against smarter adversaries. A third interesting extension is to augment the tracker to handle some types of correlations among adversaries; this would allow us to, for example, build play-books of enemy tactics and use knowledge of these plays to infer something about occluded enemies given the position of a visible enemy.

We believe that the laser tag problem is highly interesting for multi-agent mobile robotics research. Just like robotic soccer, it involves fast-moving entities; however, the nature of the sensor data (specifically the massive occlusion problems) poses new challenges in robot perception. As this paper suggests, these problems can be addressed through a novel tracking approach that tracks opponents as roles, rather than individuals.

6. REFERENCES

[1] T. Balch, G. Boone, T. Collins, H. Forbes, D. MacKenzie, and J. C. Santamaria. Io, ganymede and callisto – a multiagent robot janitorial team. *AAAI Magazine*, 16(1), 1995.

[2] Dana H. Ballard, Mary M. Hayhoe, Polly K. Pook, and Rajesh P. N. Rao. Deictic codes for the embodiment of cognition. TR NRL95.1, 1996.

[3] Y. Bar-Shalom and T. E. Fortmann. *Tracking and Data Association*. Academic Press, 1988.

[4] W. Burgard, A.B. Cremers, D. Fox, D. Hähnel, G. Lakemeyer, D. Schulz, W. Steiner, and S. Thrun. Experiences with an interactive museum tour-guide robot. *Artificial Intelligence*, 114(1-2), 1999.

[5] W. Burgard, D. Fox, M. Moors, R. Simmons, and S. Thrun. Collaborative multi-robot exploration. *Proc. ICRA-00*.

[6] G. Dissanayake, P. Newman, S. Clark, H.F. Durrant-Whyte, and M. Csorba. A solution to the simultaneous localisation and map building (SLAM) problem. *IEEE Transactions of Robotics and Automation*, 2001.

[7] A. Doucet, J.F.G. de Freitas, and N.J. Gordon, editors. *Sequential Monte Carlo Methods In Practice*. Springer, 2001.

(a)
(b)
(c)
(d)
(f)
(g)

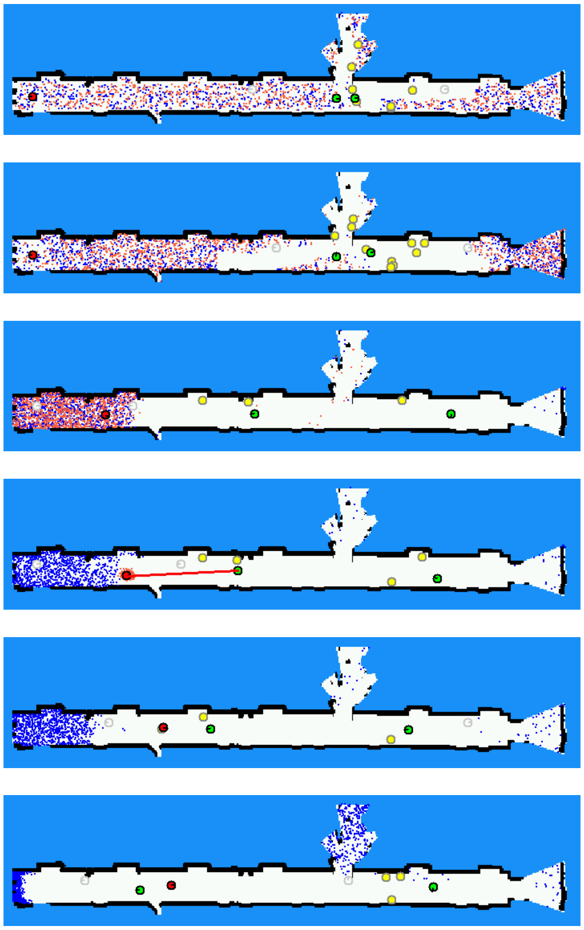

Figure 6: A complete run of the full system on real robots.

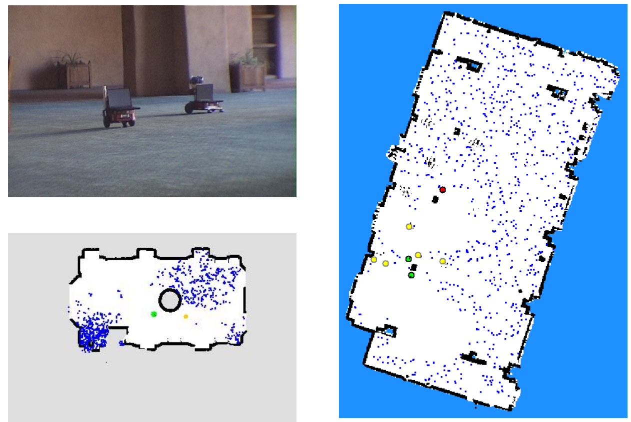

Figure 7: Environments in which our robots have played laser tag.

[8] Arnaud Doucet, Nando de Freitas, Kevin Murphy, and Stuart Russell. Rao-Blackwellised particle filtering for dynamic Bayesian networks. *Proc. UAI-00*

[9] D. Fox, W. Burgard, H. Kruppa, and S. Thrun. A probabilistic approach to collaborative multi-robot localization. *Autonomous Robots*, 8(3), 2000.

[10] D. Goldberg and M.J. Matarić. Robust behavior-based control for distributed multi-robot collection tasks. TR IRIS-00-387, USC, 2000.

[11] H. H. Gonzalez-Banos and J. C. Latombe. Navigation strategies for exploring indoor environments. *International Journal of Robotics Research*, 2002.

[12] H. Kitano, editor. *Proc. RoboCup-97*, Springer.

[13] S.M. LaValle, H. Gonzalez-Banos, C. Becker, and J.C. Latombe. Motion strategies for maintaining visibility of a moving target. *Proc. ICRA-97*.

[14] J. Liu and R. Chen. Sequential monte carlo methods for dynamic systems. *Journal of the American Statistical Association*, 93, 1998.

[15] J. MacCormick and A. Blake. A probabilistic exclusion principle for tracking multiple objects. *Proc. ICCV-99*.

[16] M.J. Matarić. Issues and approaches in the design of collective autonomous agents. *Robotics and Autonomous Systems*, 16, 1995.

[17] M. Montemerlo, N. Roy, and S. Thrun. Carnegie mellon robot navigation toolkit. Software package, download at www.cs.cmu.edu/∼carmen, 2002.

[18] M. Montemerlo, W. Whittaker, and S. Thrun. Conditional particle filters for simultaneous mobile robot localization and people-tracking. *Proc. ICRA-02*.

[19] K. Murphy. Bayesian map learning in dynamic environments. *Proc. NIPS-99*.

[20] E. Nettleton, H. Durrant-Whyte, P. Gibbens, and A. Goktoğan. Multiple platform localisation and map building. In *Sensor Fusion and Decentralised Control in Robotic Systems III*, 4196, 2000.

[21] N.J. Nilsson. *Principles of Artificial Intelligence*. Springer, 1982.

[22] I. Nourbakhsh, J. Bobenage, S. Grange, R. Lutz, R. Meyer, and A. Soto. An affective mobile robot with a full-time job. *Artificial Intelligence*, 114(1–2), 1999.

[23] L. E. Parker. On the design of behavior-based multi-robot teams. *Journal of Advanced Robotics*, 10(6), 1996.

[24] H. Pasula, S. Russell, M. Ostland, and Y. Ritov. Tracking many objects with many sensors. *Proc. IJCAI-99*.

[25] J. Pearl. *Probabilistic reasoning in intelligent systems: networks of plausible inference*. M. Kaufmann, 1988.

[26] D.B. Reid. An algorithm for tracking multiple targets. *IEEE Transactions on Aerospace and Electronic Systems*, AC-24, 1979.

[27] http://robotag.com/

[28] Nicholas Roy and Geoffrey Gordon. Exponential family PCA for belief compression in POMDPs. *Proc. NIPS-02*.

[29] R. Sargent, B. Bailey, C. Witty, and A. Wright. Dynamic object capture using fast vision tracking. *AI Magazine*, 18(1), 1997.

[30] D. Schulz, W. Burgard, D. Fox, and A. Cremers. Tracking multiple moving targets with a mobile robot using particles filters and statistical data association. *Proc. ICRA-01*.

[31] R. Simmons, D. Apfelbaum, W. Burgard, M. Fox, D. an Moors, S. Thrun, and H. Younes. Coordination for multi-robot exploration and mapping. *Proc. AAAI-00*.

[32] S. Thrun, M. Beetz, M. Bennewitz, W. Burgard, A.B. Cremers, F. Dellaert, D. Fox, D. Hähnel, C. Rosenberg, N. Roy, J. Schulte, and D. Schulz. Probabilistic algorithms and the interactive museum tour-guide robot Minerva. *International Journal of Robotics Research*, 19(11), 2000.

[33] S. Thrun, D. Fox, W. Burgard, and F. Dellaert. Robust monte carlo localization for mobile robots. *Artificial Intelligence*, 128(1-2), 2000.

[34] B. Yamauchi, P. Langley, A.C. Schultz, J. Grefenstette, and W. Adams. Magellan: An integrated adaptive architecture for mobile robots. TR 98-2, ISLE, 1998.

[35] Cao Y.U., Fukunaga A.S., and Kahng A.B. Cooperative mobile robotics: Antecedents and directions. *Autonomous Robots*, 4, 1997.