An Integrated Particle Filter and Potential Field Method Applied To Cooperative Multi-Robot Target Tracking

Roozbeh Mottaghi, Richard Vaughan
Autonomy Laboratory, School of Computing Science
Simon Fraser University, Burnaby, BC, Canada

Corresponding author: Roozbeh Mottaghi (roozbehm@cc.gatech.edu)

March 6, 2007

Abstract

We describe a novel method whereby a particle filter is used to create a potential field for robot control without prior clustering. We show an application of this technique to control a team of mobile robots to cooperatively locate and track a moving target. The particle filter models a probability distribution over the estimated location of the target, providing robust tracking despite frequent target occlusion. This method extends previous work in particle-filter-based tracking in two important ways. First, the particle cloud is never clustered to find a single estimate of target location. Instead, robot motion is guided by a potential field generated directly from the particle cloud. Secondly, effective coordinated multi-robot searching and tracking can be achieved by simply assigning a subset of the particles to each robot.

Simulation trials and real robot experiments demonstrate the method successfully locating and tracking targets, and experiments show that multiple coordinated robots outperform a similar but uncoordinated team.

1 Introduction

1.1 Problem Statement

Estimating the position of target objects is an important sub-problem in many real-world robotics applications. There are two basic variations of this problem: *locating* an object whose position is initially unknown, and *tracking* the changing location of an object given an initial location. The challenge in both variations of the problem is coping with the limitations of target-detecting sensors. Even assuming a sensor that reliably detects targets within its field of view, the target may be out of range or occluded. In this case, the task becomes one of moving the sensor so that the target falls within its field of view. If the target is moving, the sensor may need to move to keep the target in view. This is a natural application for mobile robots, which can sweep large areas with their sensors over time in contrast to static sensors. Using multiple robots can increase the effective sensor field of view, and so should in theory enable superior performance. However, a multi-robot coordination strategy is required to maximize the effective sensor area and to avoid spatial interference between robots.

This paper proposes a method for locating and tracking a moving target using one or more mobile robots. The method is robust to pervasive occlusion of the target. When more than one robot is used, the method automatically coordinates their activity to effectively parallelize the search task and avoid spatial interference.

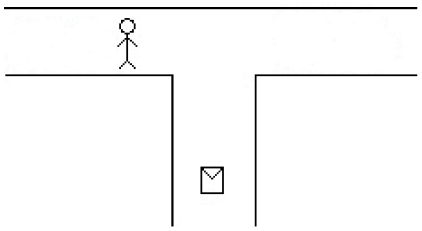


Figure 1: A robot (rectangle) tracking a person (stick figure) who has disappeared from view.

1.2 Background

Target tracking systems are vulnerable to failure in situations in which targets go out of the field of view of the sensors for a long period of time. Consider the task of a mobile robot tracking a person through a building. Figure 1 shows the situation where a robot has followed its target down a corridor to a T-junction, and the target has left the robot's sensor field of view. Assume that the robot did not detect whether the person moved to the right or the left. The naive tracker has now lost the target.

A general solution to the occlusion problem is to use a model-based approach, whereby the tracking system can make some predictions about the behaviour of the occluded target. Probabilistic models have become popular due to their robustness in the presence of uncertainty, which makes them able to cope with large amounts of sensor noise and occlusion. One elegant feature of using a probabilistic model is that the target locating problem can be reduced to the tracking problem. If the initial target location is represented as a prior distribution over possible positions, then a uniform distribution over all possible positions represents a 'null' or 'don't know' hypothesis (most practical implementations make use of the particle filter to approximate the true probability distribution). If the tracking method is powerful enough to locate the target from such a weak prior, then we have a general locating-and-tracking method.

In addition to extensive work on target tracking in the computer vision community, several authors have described tracking systems using autonomous mobile robots. As an example of a recent work (Schulz et al., 2003) have introduced sample-based joint probability data association filters to track multiple moving objects. (Montemerlo et al., 2002) present a probabilistic algorithm called the *conditional particle filter* to track a large distribution of people locations conditioned upon the robot poses. These approaches are concerned with the guidance of individual robots only.

Target tracking performance and reliability can be improved by deploying multiple robots, but in general some coordination strategy is needed to avoid interference and maximize parallelism.(Jung and Sukhatme, 2004) have proposed an example cooperative system. (Stroupe et al., 2004) have developed an optimized coordination strategy to track multiple targets with a team of robots. Their approach, in common with the other approaches mentioned above, works well when the target lies in the sensors' field of view or has a short-term occlusion but they do not address the long-term occlusions which cause large uncertainty for the tracker.

Pursuit-evasion games also model a searcher chasing an evader. This class of problems guarantees that even in the worst cases in which the evaders move arbitrarily quickly, any evader would be found by a group of pursuers. (LaValle et al., 1997) proposed a pursuit-evasion method for k-searchers where each searcher is equipped with k flashlights by which they can see the environment. (Gerkey et al., 2004) recently showed a generalization of this method by introducing a new class of searcher, the ϕ-searcher where each pursuer has a ϕ radian field of view instead of infinitely thin beams.

A practical deficiency of the known solutions to the pursuit-evasion problem is that they are highly computationally intensive and do not scale well in application to multiple-robot systems. For example, in Gerkey's approach, the joint information and action space grows exponentially

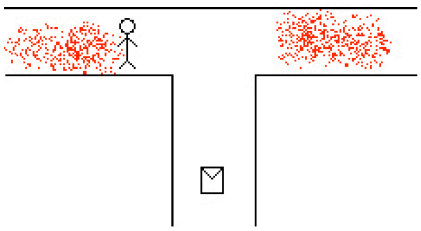


Figure 2: The point-cloud represents a set of hypotheses about the target's current position, generated by a probabilistic model of its movements.

in the number of searchers. In addition, this method cannot be implemented perfectly for real applications, since it requires searchers to exactly traverse wall and obstacle boundaries. In the real world this can not be perfectly achieved, so the algorithm would lose its correctness guarantee. A further limitation of pursuit-evasion methods is that these methods do not address the continuous tracking of intruders once they are found.

Building on this previous work, we propose a novel technique that tightly integrates robot motion control with the target position estimation system to locate and track targets. Further, the method can be used to coordinate multiple robots using a modest amount of robot-robot communication. As mentioned above, existing tracking methods (excluding the pursuit-evasion methods) focus on combining sensor measurements to track the object of interest. The problem arises when the object of interest goes out of the field of view for a long period. Our method simultaneously addresses this problem and the problem of coordinating multiple robot trackers.

1.3 Task and Approach

Consider the example shown in Figure 1 again. Assume that the target may have moved left or right, but its top speed is limited and it may not move through walls. A probabilistic model of the target's future movements would generate a bimodal probability distribution over the possible future target locations, which would be represented in a particle filter as a split particle cloud. Each dot in Figure 2 shows a possible location of the intruder. If each particle is weighted with the same probability, the density of particles in a region indicates the probability that the target will be found in that region. If a particle falls within the sensor's field of view, but no target is detected, the particle can be eliminated. Considering this model, we can state a simple rule that maximizes the probability of observing the target: the robot must visit all particles, where 'visit' is defined as positioning the robot so that the particle falls within the sensor field of view.

The task of our robot controller therefore is to maximize the number of visited particles. This approach was taken by (Rosencrantz et al., 2003) to locate the opponents in a laser tag game in which the opponents might be under pervasive occlusions. However, their work mostly addressed the improvement of the tracker for multiple opponent tracking, rather than coordination of multiple trackers.

In this paper which is an extension of (Mottaghi and Vaughan, 2006), a particle filtering method has been implemented to represent arbitrary multi-modal densities for the location of the intruder. Then we apply a potential field method on top of the particle filtering for coordinating multiple agents to move so as to reduce the uncertainty in the target location. Each agent decreases the uncertainty in its own target position estimate by observing as many particles as it can.

Section 2 presents an outline of the probabilistic tracking method, robot mapping and localization. Section 3 introduces a novel technique where a particle cloud and map are combined to create a potential field robot controller for a single robot. In section 4, we present cooperative action selection and optimization strategies for searching the environment by multiple robots. Then an experimental section shows the result of the implementation of the proposed method on

simulated and real robots in different environments. Additionally, the performance of the system has been studied with respect to different population sizes, with and without communication, and also different types of communication between teammate robots.

2 Probabilistic Tracking

2.1 Prerequisites: Localization and Mapping

We assume a map of the robot's workspace is available. Such a map can be provided *a priori*, or acquired automatically online or offline using tools such as those described by (Thrun, 2002). For the experiments in this paper, we supply *a priori* maps. Also, the robots need an estimate of their location and orientation. Using the map, odometry and laser scan data, good pose estimates are obtained using Monte Carlo localization (Thrun et al., 2001), where the belief about the position of the robot at the current time step, $Bel(x_t)$, can be estimated by recursive update of the following equation:

$$Bel(x_t) = \eta p(z_t|x_t) \int p(x_t|x_{t-1}, a_{t-1}) Bel(x_{t-1})\, dx_{t-1}. \tag{1}$$

where, $p(z_t|x_t)$ is the sensor model and $p(x_t|x_{t-1}, a_{t-1})$ is the next state density or motion model and a_{t-1} is the action performed in the last time step. The separation of mapping stage from localization can somewhat help us to reduce the computational burden of the whole system.

2.2 Particle Filter Tracking Method

Since our target of interest is moving autonomously, and may be invisible to our sensors for extended periods, our probabilistic estimate of its position may have a multi-modal distribution. Thus we use particle filtering to track it. For instance, such a multi-modal distribution is caused when the particles arrive at a 3-way junction of corridors as shown in Figure 2. According to a pre-defined motion model of the target, combined with the map, the pose-estimate particles spread into the available space. The state that we want to estimate consists of the location and orientation of the object. So our state vector has the form $x_t = [x, y, \theta]$ where x and y are two-dimensional Cartesian coordinates of the object on the map and θ represents its orientation.

In the cases when we have an observation of the object, a probability distribution over the state space is found according to the measurement, $p(x_t|z_1, z_2, ..., z_t)$, that is the probability of the state at time t provided that the observations from time 1 up to time t are equal to z_1,z_2,...,z_t respectively. Using Bayes' rule, $p(x_t|z_1, z_2, ..., z_t)$ is computed as follows:

$$p(x_t|z_1, ..., z_{t-1}, z_t) = \frac{p(x_t|z_1, ..., z_{t-1}) p(z_t|x_t, z_1, ..., z_{t-1})}{p(z_t|z_1, ..., z_{t-1})} \tag{2}$$

Since the measurement at time t is independent of the previous measurements given the current state, according to the above rules $p(z_t|x_t, z_1, ..., z_{t-1}) = p(z_t|x_t)$. Also $p(z_t|z_1, ..., z_{t-1})$ is a constant. Therefore:

$$p(x_t|z_1, ..., z_t) = k p(z_t|x_t) p(x_t|z_1, ..., z_{t-1}) \tag{3}$$

We can compute $p(x_t|z_1, ..., z_{t-1})$ by applying the dynamic model of the object to $p(x_{t-1}|z_1, ..., z_{t-1})$ which is known from the previous time step. The dynamic model is a known motion model of the object and it is approximated before the start of tracking or during the tracking online and it relates the state vector at current time step to that of previous time step. The motion model can be defined as a combination of deterministic and stochastic parts:

$$x_t = f(x_{t-1}) + stochastic\ \ part \tag{4}$$

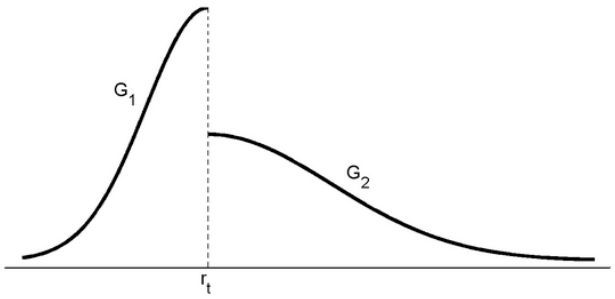


Figure 3: Motion of the target is modeled with two different Gaussian functions.

f can be any function which relates the current state of the samples to the previous state. Because the movement of the object is unpredictable from the point of view of the tracker, we add a stochastic part to the motion model to add unpredictability to the deterministic model. For example, the intruder can stop or move backward instead of moving forward which is forced by the deterministic model.

2.2.1 Motion Model

The motion model depends on the intrinsic properties of the object and the environment in which the object moves. For example, (Liao et al., 2003) and (Bruce and Gordon, 2004) have studied the motion model of people in an indoor environment where the former assume the motion is constrained to the voronoi graph of the environment and the latter use the intuition that people tend to move on efficient paths rather than random trajectories. In this paper, we use a conservative motion model since we do not have any assumption about the tracked object. The particular motion model used for our experiments is described below. If a robot perceived the previous movement of the target, we denote the magnitude of target displacement vector at time t by r_t. The general idea is to add some unpredictability to r_t and move each particle in the direction of its orientation by the amount computed from r_t.

Our motion model is a combination of two Gaussians which peak at r_t. This models our expectation that, at any timestep, the current motion of the target is the same as its previously observed motion. The reason for choosing two Gaussian functions is that the probability of increasing speed might be different from reducing speed. For instance, in most cases the target goes out of sensor range at corners and usually a target reduces speed when it turns. Therefore, it is a good assumption that the target will increase its speed after turning. Figure 3 shows the motion model distribution where it is composed of $G_1(\mu = r_t; \sigma_1)$ and $G_2(\mu = r_t; \sigma_2)$.

For every particle, we sample from this distribution and add the movement vector to the current position of the particle. Since the distribution is not well-defined, the sampling step is not straightforward. The sampling procedure is as follows: one of the distributions (G_1 or G_2) will be selected with probability of 0.5. After selecting one of the normal functions, we sample from that distribution. If a sample from distribution G_2 falls in the region on the left of r_t, we just consider a value that has the same probability but is larger than r_t. The same explanation is true for the samples from G_1. σ_1 and σ_2 are specified *a priori* such that they represent our assumption about the motion of the tracked target.

Therefore, the magnitude of the displacement is equal to the sample drawn from the above distribution but only a simple Gaussian noise is added to the orientation. It should be noted that when the target is observed σ_1 would be equal to σ_2.

If $\overline{s_i^t s_i^{t-1}} \cap C_{obs} \neq \emptyset$, the sample is moved to the nearest point in the free space in its direction of movement and noise is added to the previous orientation. $\overline{s_i^t s_i^{t-1}}$ is the line segment that connects

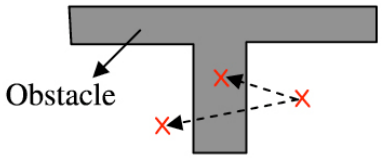


Figure 4: Motion of the target is constrained to free space.

the position of sample i at the current time step to its position in the previous time step and C_{obs} is the space of all of the obstacles present in the map. Intuitively, it means that the particles are not allowed to go inside an obstacle or through a wall. Figure 4 shows this case. This models the constraints on the target that it can only move through free space.

2.2.2 Reweighting

The next step would be the reweighting of the samples to find the probability distribution of the object over the state space. Two cases have been considered for the reweighting step:

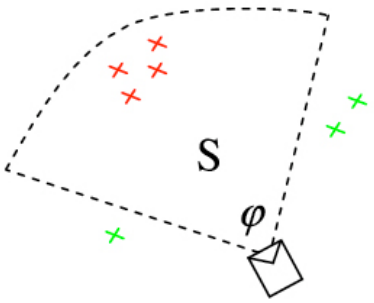
1. We define the area S which is a circular segment as the sensor visibility of the agents. This area is centered at the robot position and its central angle, φ, is in the range $\theta - \frac{F}{2}$ to $\theta + \frac{F}{2}$ where θ is the robot orientation and F is the field-of-view angle subtended by the sensor. It should be noted that not having perfect localization will not affect the method. Finally, the radius of the circular segment shows the sensor range. If the i^{th} sample $s_i \in S$ and the line that connects the sample to the robot does not intersect an obstacle, w_i the weight of that sample will be zero. This case is shown in Figure 5(a). Thus the robot deletes the particles that it now knows do not correspond to the real position of the target.

2. If have we an observation of the object, the Factored Sampling method (Isard and Blake, 1998) is used to find the new weight of the samples. If the number of samples goes to infinity the distribution of samples from $p(z_t|x_t)p(x_t|z_1, ..., z_{t-1})$ tends to be that of $p(x_t|z_1, ..., z_{t-1}, z_t)$. A reasonable assumption for accommodating noise in the sensor model is to be a Gaussian function where the mean of the Gaussian is located on the real measurement from the sensor and its deviation is determined according to the sensor and the map. Figure 5(b) shows the reweighing of each sample according to the Gaussian function. It should be noted that we have shown the observation density for a one-dimensional case. A higher degree function needed for higher dimensional state spaces.

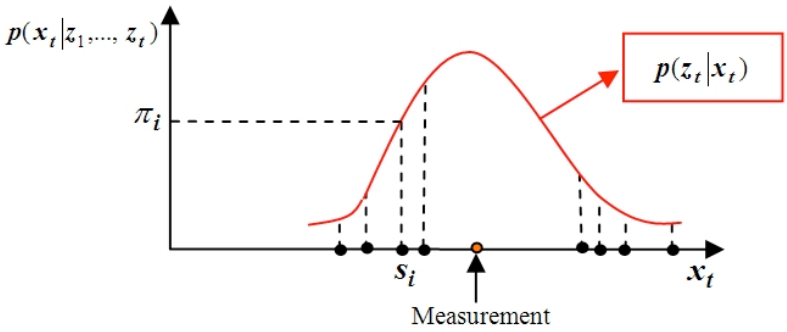
If neither of the above cases happened, the sample keeps its previous weight or we can assign an equal weight to all of the samples. We describe below how these particles and their weights are used to track the object. In the traditional particle filter tracking system, some clustering method is used to decide what the current 'actual' estimate is. Usually the weighted mean or median of a particle cluster is considered to be the target. We avoid this clustering step, and thus avoid the artifacts it can introduce. As we may not have an observation during the tracking, we try to maximize the number of visible particles while simultaneously optimizing the joint motion of the robots.

3 Tracking using potential fields

Our goal is to reduce the uncertainty in the target position by maximizing the number of visited particles. Also, the likelihood of finding the target is maximized if all the particles are swept off.



(a) The robot assigns low weights to the visited particles.



(b) Reweighing the particles according to the sensor model.

Figure 5: Weighting schemes.

A practical problem with most of the existing methods such as POMDPs (Partially Observable Markov Decision Processes) which are otherwise well-suited to solve this kind of problem, is that their computational complexity or memory needs grow exponentially with the number of robots (Roy et al., 2005). The problem also can be simplified to find a polygon in the environment to cover as many particles as possible and move the robot with a limited field of view to that area to cover a large number of particles. But there is no polynomial algorithm for performing these calculations.

The problem can be formalized as minimizing the entropy of the particles as a measure of predictability of the system. Therefore, the searcher robots should try to decrease the entropy of the probability distribution over target locations. In practice this means shrinking the area containing the particles into a small region which will be a good estimate of the location of the target. However, one of our major restrictions is that when the target is not visible, we do not have complete information about the probability distribution that is necessary to compute the entropy of the system. Therefore, we use a heuristic approach to decrease the uncertainty in the location of the target that may not always result in the optimum solution. However, the experiments described below suggest that our method performs well in practice.

We implement a potential field method for doing this task which is $O(N_p n)$ in the worst case where n is the number of cells if we represent the map by grid cells and N_p is the number of particles used in the tracking algorithm. For speed, the number of particles which are used for the calculation of the forces can be decreased by selecting at random a subset of the particles. Since the particle filtering results in producing more particles in the high probability areas, the chance of choosing particles in those areas would be higher and the distribution of the particles is

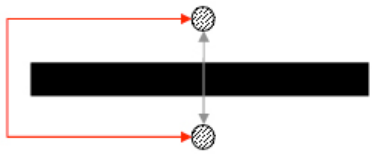


Figure 6: The short arrowed line shows the Euclidean distance between two points which is not useful for calculating the attraction force because of the presence of the obstacle. The longer arrowed line shows the traversible distance.

approximately the same. So we perform the update stage for the whole set of particles but calculate the forces based on the randomly chosen particle subset. Potential Field methods were used for robot control and motion planning in their early robotics applications (Khatib, 1986), where a robot was considered as a particle which moved in a free space. Obstacles exerted repulsive force on the robot while the goal exerted an attractive force on it and the resultant force caused the robot to move from a source to a goal without colliding with obstacles.

In the following subsection we explain the method for finding the distances on a traversibility map, instead of Euclidean distance, and subsequently, how a potential field can be generated to provide motion control for tracking.

3.1 Finding map distances

For maximizing the number of visited particles, the intuitive idea is that each particle exerts an attractive force on a robot. The greater number of particles in an area, the larger the attractive force in that direction experienced by the robot. This force is inversely proportional to the distance from the particle to the robot. This means that the robot tends to sweep the nearest particles first. But when the robot's mobility is limited by obstacles, as shown in Figure 6, the Euclidean distance from particle to robot does not indicate how quickly the robot can reach the particle. Instead we must calculate the shortest traversible path using the map.

In our implementation, shortest-traversible-path calculations are done using a simple occupancy-grid flood fill method, though any equivalent method could be substituted. The distance algorithm outputs a value which is assigned to each grid cell and shows the distance of that cell from the cell where the robot is located. The flood-fill works as follows: First, we assign a zero value to the cell in which the robot is located and an infinite number to the obstacles. Then, we pick one of the free cells around the robot cell and increment its value by one and put that cell in a queue and pick another neighbor cell until there is no cell around the current cell without an assigned value. The order of picking the neighbors is important and we pick only top, bottom, left and right neighbours. After that, we pop the first cell in the queue and perform the same procedure for its surrounding cells. This algorithm is continued until there is no cell in the queue. This method returns the minimum map distance of a point to the current position of the robot and its time complexity is $O(n)$ when implemented by a queue where n is the number of cells on the map. Figure 7 shows an output of this method for measuring the map distance of a cell of the map.

Thus we find the map distance of each particle from the robot as required for the calculation of the forces exerted by the particles. These calculations are explained in detail in the next subsection. For simplicity, from now on we represent the map distance of a cell, which is located at row i and column j, from the robot cell by $\Delta(i, j)$.

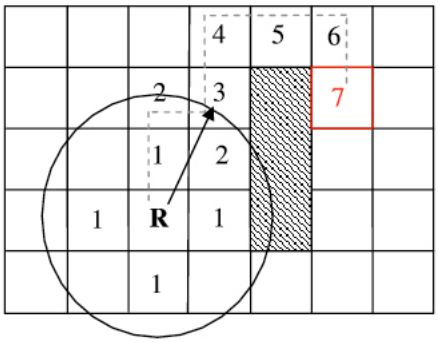


Figure 7: The map (traversible) distance of the cell numbered 7 the robot cell labeled **R** is calculated by using the flood fill method. The shaded area is an obstacle on the map. The circle shows the circle that encloses the cells used to compute the local motion vector, shown as an arrow.

3.2 Computation of potential forces

The navigation of our robots is based on the total force which is exerted on the robots by randomly selected particles. That means at each time step, we apply the normalized total force to the robot to find its next target position. An underlying position device based on the extended Vector Field Histogram (VFH) (Ulrich and Borenstein, 1998) performs the task of avoiding local obstacles while moving according to the potential field. We use the well-known VFH implementation that is provided with the Player robot server (Gerkey et al. 2001). To compute the total potential acting on a robot, we find the force vector for each particle. Then, we sum the vectors to find the magnitude and direction of the resultant total force.

To find the approximate direction of the resultant of the total particle forces, we start from the cell where the particle is located and we check its surrounding cells, the cell with the minimum value will be selected. We continue performing the same procedure for the minimum-value cell until we reach a certain distance from the robot cell (this approximate distance is indicated by the circle in Figure 7). The direction of the force is approximated by the direction of the vector from the robot to the cell that is reached through the above procedure. The reason that we do not use directly the vector from the robot to the particle, is that the vector may intersect the obstacles that has blocked the robot way. Figure 7 shows an example of finding the force direction. The dashed line shows one of the paths from the goal cell (marked 7) to the robot cell and the vector from the robot to the cell with value 3 can be considered as the force direction. A suitable value for this threshold parameter is determined empirically. We found that a few multiples of robot radius worked, in combination with VFH's local steering, to generate smooth paths around complex obstacles.

If m_i and n_i are the row and column index of particle i in the map grid, the magnitude of the force exerted by that particle, F_i, is calculated by the following Gaussian model:

$$F_i = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\frac{\Delta^2(m_i,n_i)}{\sigma^2}} \tag{5}$$

where σ is assumed to be a constant in the whole process of tracking or it can be determined according to particle data. σ is the parameter that determines the priority for sweeping the particles close to the robot. The reason for choosing a Gaussian function is to assign more priority to the closest particles while the deviation is tunable by σ. This equation means that the closer particles exert a larger force and the first priority of the robot is to sweep the nearest particles. Nevertheless, if the number of particles is large in an area the robot will be attracted to that area neglecting the nearest particles (unless they happen by chance to be swept by the sensor field of view as the robot moves). The magnitude and direction of the attractive force is determined by

the vector summation of the forces from all of the particles which were selected randomly from the whole set of particles. The robot will be driven around according to the direction of the resulting force.

The main criticism of potential field methods in general is that rapidly changing local optima can cause an oscillatory behaviour in the navigation of the robot. However, because of the random nature of the particle filtering method and clearing of the particles during the navigation, the symmetry breaks and we have not observed adverse oscillations in the robot movements. There are a lot of tricks to get rid of the oscillation caused by potential field methods. An ad hoc solution which improves the performance slightly is to consider the direction and magnitude of the past forces in the calculation of the current force that is exerted on the robot.

This method is very easily extended to rationally coordinate multiple tracking agents, as described in the next section.

4 Coordination Strategies

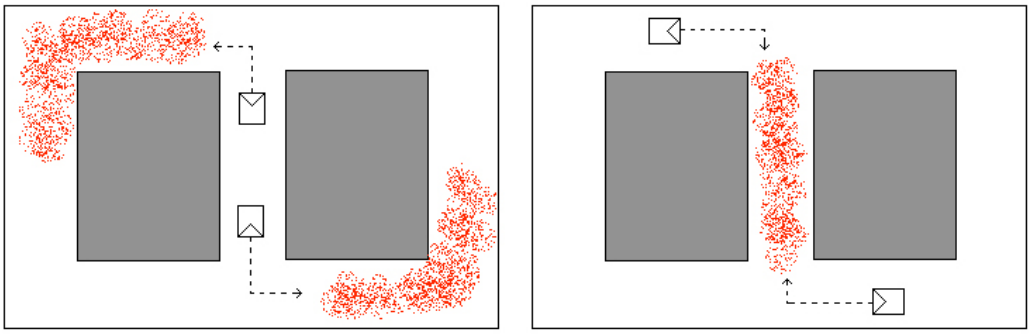
The tracking performance of the system can be improved by simply adding more robots, but to maximize performance, the robots' actions should be coordinated in some way. In this section we describe how multiple robots can cooperate to perform the assigned task according to the potential fields which have been formed by the particles. The basis of the technique is that each robot selects a subset of its particle cloud to visit, and ignores the rest. Various alternative methods of selecting particles can be considered, but in this paper we allocate to each robot those particles that are closer to it than to any other robot, i.e. in a Voronoi diagram of robot locations, we allocate each particle to the robot in its containing cell.

This requires that each robot has an estimate of the location of the other robots. Each robot can send its global position information to the teammates through communication or it can localize the other robots in its coordinate frame. Both of these constraints are feasible using current methods and since the exact position of the other team members are not required, any suitable approximation method can be used. In our simulations, we use communication among the robots. The communication can be direct communication between two robots or in the case of limited communication range, a robot can get the location information of one robot through communication hops via intermediate robots. Each robot has its own set of particles which are updated according to the robot observation and the information received from teammates and no particle information is exchanged among the robots.

As stated before, we want to reduce the uncertainty in target position estimate by maximizing the number of visited particles (observing the locations of more particles). So our goal is to cover an area that is occupied by larger number of particles and to prevent the particles from further spreading. Two simple cases are shown in Figure 8. Figure 8(a) shows the case where we have two high density regions that means the chance of finding the intruder is high in those two regions. The best action to minimize the uncertainty is that one robot goes toward one cloud of particles and the other robot goes toward the other cloud. The next figure (8(b)) shows the case where there is a single high density area. The best action to shrink the particles' area and prevent it from further growing is that the robots approach the covered area from different directions. Our coordination method achieves the desired behavior while minimizing the path that a robot navigates.

To coordinate the motion of the robots, we compute the cooperative forces which are exerted by the set of particles on each robot. These forces will determine the navigation direction of the robots. First, we assign a value to each particle according to density and distance of the robots. The more negative the value, more desirable for the agent to go toward that particle. This value which is represented by $V_{n,j}$ for particle n relative to agent j is determined by:

$$V_{n,j} = \sum_{i=1, i\neq j}^{N} \begin{cases} -w_i F_{nj} & \Delta_i > \Delta_j; \\ w_i F_{ni} & \Delta_i \leq \Delta_j. \end{cases} \tag{6}$$



(a) (b)

Figure 8: Results of the coordination strategy for two robots in two simple cases. Each robot moves towards those particles that are closest to it, leading to appropriate coordinated behaviour.

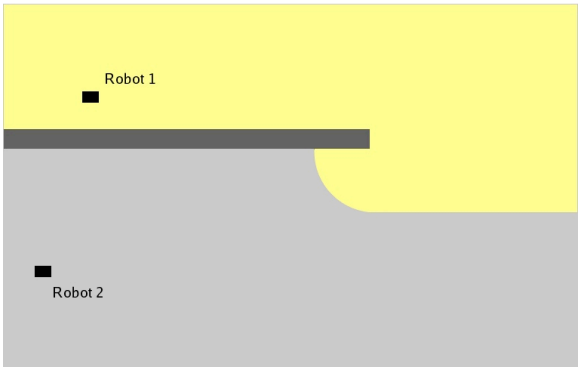


Figure 9: Two robots (small rectangles) are separated by a wall (dark rectangle) . The light and dark shaded areas show the set of points that are closer, in traversible map distance, to each robot.

where F_{nj} and F_{ni} are the forces that particle n imposes on agent j and agent i, respectively and are computed according to Equation 5. N is the number of agents used for tracking and w_i is a priority factor which is used to assign higher priorities to some agents. Also, Δ_i and Δ_j are the map distance of the n^{th} particle from agent i and agent j. Intuitively, this equation means that the parameter V will be more positive for a selected particle and a specific robot if the density of the other robots around the particle is high. That means each robot will attend to those particles that are closer to itself than to any other robot. The boundary between the shaded regions in Figure 9 is the boundary of positive and negative values of map distance differences in an example configuration of two robots. We normalize these values to get positive force magnitudes. The normalization is done by an exponential function again. So, the force magnitude that particle n exerts to agent j in presence of the other robots, $F_{n,j}$, is calculated as follows (note that $F_{n,j}$ is different from F_{nj} since F_{nj} is that force without the presence of the other robots):

$$F_{n,j} = e^{-(V_{n,j} - V_{min})^2} \tag{7}$$

where V_{min} is the most negative value. The direction of the force is also found by the procedure described in the last section. Now, we should find the vector sum of the forces which are exerted to one robot by the set of particles, So:

$$F_j^{tot} = \sum_{n=1}^{N_s} \vec{F_{n,j}} \tag{8}$$

where N_s is the number of randomly selected particles. As mentioned before, for the sake of efficiency, we use a small set of particles for force calculations and only the update step (in particle filtering) is done for the whole set of particles. The direction of navigation of robot j is determined by $\vec{F_j^{tot}}$.

5 Experiments

We present a series of experiments to demonstrate and evaluate the utility of our method. All of the requirements for the proposed method including localization and mapping, probabilistic tracking and path planning, have been discussed above. In this section, we show simulated and real examples of target tracking, highlighting the ability of the method to cope with periods when the target object is not observable.

The experiments are done in simulation using the well-known Player/Stage robot development and simulation system (Gerkey et al., 2001). We use the standard mobile robot base and scanning laser rangefinder models included with Stage, in their default configurations. These model the common ActivMedia Pioneer-3DX robot and SICK LMS-200 rangefinder. We also use a "fiducial-finder" to detect the objects of interest, generically modeling a feature detector on a camera or other sensor. A simple real-world validation is performed using a two physical Pioneer-3DX robots with SICK LMS-200s. In all cases, the robots communicate through TCP to exchange position and orientation information.

We present:

A1. A simulation demonstration of two robots cooperatively tracking a target in an artificial corridor environment.

A2. A real robot demonstration of two robots cooperatively tracking a target in an real office building corridor environment.

B1. A simulation experiment to compare the performance of *target tracking* with and without coordination.

B2. A simulation experiment to compare the performance of *target finding* with and without coordination with robot teams of different sizes.

It should be mentioned that in all of the experiments, $\sigma_2 = \mu$ and $\sigma_1 = \mu/3$ for our motion model.

5.1 A1: Simulated Robot Demonstration

In this simulation demonstration, closely resembling the experiment of (Jung and Sukhatme, 2004) two searcher robots, $S1$ and $S2$ start out at the bottom of a figure-8 shaped world the world as shown in Figure 10(a) (left), and try to locate the position of an intruder, T, which starts at some unknown location an can move. Each robot has an identical map of the world, and an initially null hypothesis about the location of the target: the particle filter particles are initially spread randomly in the traversible free space, indicating that the target could be anywhere on the map (Figure 10(a) right).

The intruder is controlled manually by the human experimenter who sees the whole world and tries to evade the searchers. In Figure 10 the left-hand pictures are screenshots from the Stage simulation and the right-hand pictures show the map and particle cloud maintained by one of the robot controllers. The robots start searching by moving in opposite directions along the corridors, which maximizes the number of particles visited and thus reduces the uncertainty in the target position estimate (Figure 10(b)). Then S2 sees the intruder but the intruder suddenly goes out of its field of view. Therefore the particles are gathered in a small region of the top right and robot

S1 takes the shortest path toward the particles (Figure 10(c)). But as the target becomes occluded again as it attempts to escape from S2, the particles start spreading along the top corridor. This causes S1 to reverse direction and to move left and up to catch the target as it moves left along the top corridor, while the other robot covers the area behind the intruder (Figure 10(d)).

Finally, the searchers trap the intruder and both observe it simultaneously. The particle cloud shrinks to a small region giving a very good estimate of the target position (Figure 10(e)). Note that the final target position distribution has much higher kurtosis than is apparent in the images, due to their limited resolution.

5.2 A2: Real Robot Demonstration

We seek to verify that the proposed method can work in practice using real robots. The main challenge is dealing with noise in robot sensors and actuators that is not present in simulation. We collected map data offline from our lab environment and used two Pioneer-3DX robots operating in our office building to run the real-world demonstration.

The robots' task is similar to that of the simulation demonstration, except that in this case, we allow for the possiblity that no target is present. This is defined as the case if the robots can continously and simultaneously observe all possible locations of the target and thus determine that is is not present. The corridor map provided to the robots is shown in Figure 11(a). The particles are distributed uniformly on the entire map at the beginning which means the intruder can be anywhere on the map. Intuitively, if the robots clear all of the particles, they can be certain that there is no intruder inside that part of the building.

Photographs of the real robot demonstration are shown in Figure 11, showing the robots at five different points on the traversed path (Figure 11(a)). Figure 11(c) shows the two robots together near the beginning of the trial. The robots exhibit desirable behaviour, heading off in opposite directions to avoid clearing the same particles, thus maximizing parallelism, and searching the area faster. Figure 11(g) shows the two searchers a few minutes later converging on the only remaining unexplored area. Having observed that no target is present here, the robots correctly conclude that there is no target on the map. Figure 11(b) shows the approximate trajectories of the two robots.

5.3 B1: Experiment 1 - Evaluation of Tracking

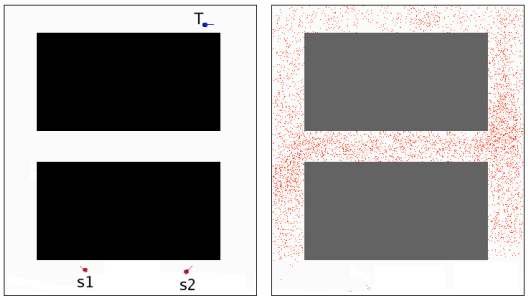
In this section, we evaluate the performance of the method for target finding and tracking for a team of robots. Our performance metrics are the amount of time spent for finding the intruder, and the amount of time following first observation that the target is occluded. Less is better in both cases. Our goal is to show that a system of coordinated robots will outperform similar but uncoordinated robots.

We perform three experiments in this section:

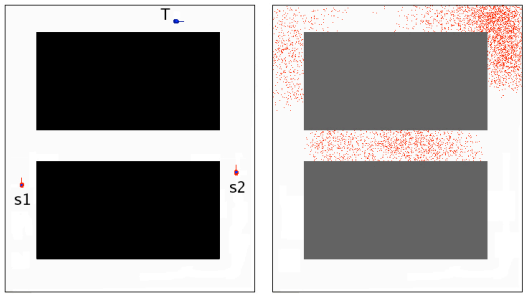
1. Searcher robots try to find the intruder without any coordination strategy. They do not have any information about the teammates and act individually.

2. The robots communicate by broadcasting their pose and any target observations they make.

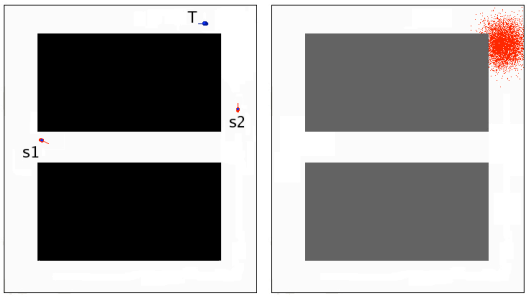
3. The teammate robots share the set of randomly selected particles used in force calculations as well as the pose and observation which means the robots have approximately identical models of the probability distribution of the target pose. This method requires more communication bandwidth but more information is provided for coordination of the robots. Because of the random nature of particle filters, in some situations, the number of particles may be small in an area from one robot's view while the other robot has large number of particles in that area. Sharing the particles will usually provide more information to the robots in these situations.



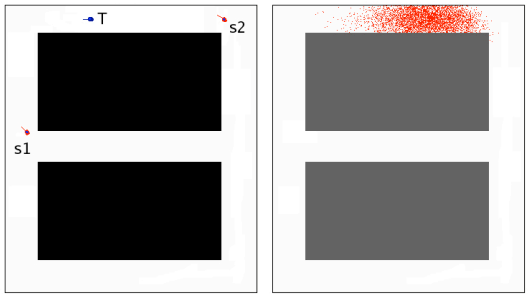
(a) t_0 Initial positions.



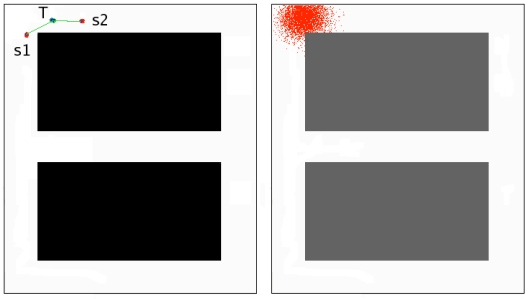
(b) t_1 S1 and S2 split up to explore opposite coridoors.



(c) t_2 S2 observes T. S1 takes shortest path to T.

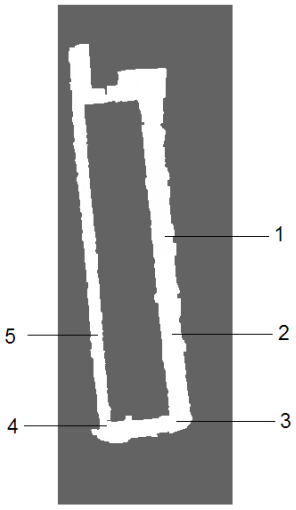


(d) t_3 S2 loses obervation of T, so position estimate becomes more uncertain. S1 moves to explore to the left side again.

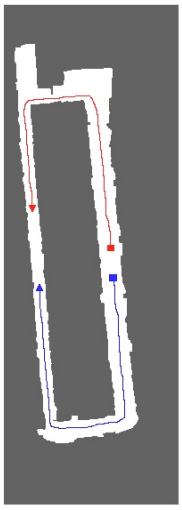


(e) t_4 S1 and S2 both observe T.

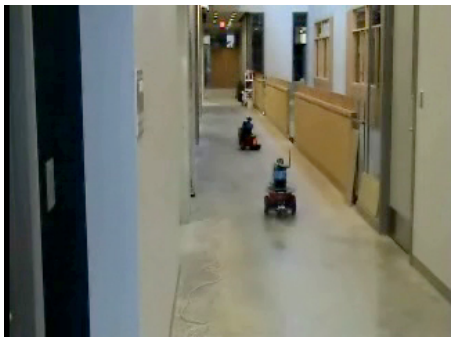
Figure 10: Simulation results of two searchers tracking an intruder over time, from t_0 to t_4. S1 and S2 are two searchers that try to find and catch target T. In each case, the left-hand image is a screenshot from the Stage simulation, and the right-hand image shows the map and particle cloud maintained by S1.



(a) Map of building showing locations of photographs



(b) Map of building showing approximate trajectories of robots



(c) Snapshot taken at point 1

(d) Snapshot taken at point 2

(e) Snapshot taken at point 3

(f) Snapshot taken at point 4

(g) Snapshot taken at point 5

Figure 11: Photographs of the real robot demonstration with two robots in the SFU Computing Science building. The corresponding points on the map are shown in Figure 11(a).

Table 1: Time that the searchers had no observation before the first observation of the target.

Trial Type	Mean (sec)	σ	T-test p-value
No coordination	175.7	99.5	—
Shared Observation and Pose	106.5	85.3	0.0076
Shared Observation, Pose and Particles	85.7	54.5	0.0009

The experiments were performed in a simulated indoor office environment whose map is represented in Figure 12. Two searcher robots must initially locate and then track a moving intruder. From a random start position, the intruder chooses a random location on the map as a goal point and heads toward that location while avoiding obstacles using VFH. After a fixed interval of time, a new goal location is selected.

The start position of the searchers is the same in all of the experiments. The range of the robot sensors is set to be 8 meters with an angle of view of 120 degrees while the environment dimension is $24m \times 20.5m$. While the sensor field of view is a large fraction of the total map, the shape of the environment means that the intruder is often occluded. The start position of the robots is shown by rectangles in Figure 12. The shorter trajectory on the left side of the picture is the path that the target has traversed.

The results shown in Figure 5.3 were gathered from two robots in 10 trials of five minutes of tracking (20 samples for each experiment in total). The light gray area shows the mean percentage of time a searcher spends before first locating the intruder. The dark gray area is the average percentage of time that the searchers had no observation after they see the intruder for the first time. Smaller is better is both cases.

5.3.1 Results

Figure 5.3 shows the total time that the robots have no observation (sum of the values of light and dark gray areas) in the shared-particle and non-shared particle case is less than that of no-coordination case, indicating that the performance is improved on average by cooperation. Tables 1 and 2 show the results of the T-test comparing the times spent before the first observation of the target and the total time with no target observation, respectively. Since the p-values of the T-test are less than 0.05, we can conclude that the data do not belong to distributions with equal means and there is a significant difference between the performance of non-coordinated and coordinated cases where coordinated cases have performed better. The large standard deviations are an expected issue in this experiment because the starting position of the intruder was different in the trials and the amount of time required for finding the target can vary greatly. Nevertheless, these data show an improvement in average search time between coordinated and uncoordiated methods.

5.4 B2: Experiment 2 - Evaluation of Communication Effect

In this experiment, we evaluate the target-locating performance of various robot population sizes in coordinated and non-coordinated cases. Our hypotheses are that (a) increasing the number of the robots will improve performance; and (b) coordinated robot teams will preform better than non-coordinated teams. For this experiment, coordinated robots communicate only their current pose and any target observations they make. Uncoordinated robots do not communicate.

We performed 50 trials of this experiment and tested the effect of communication for two, three and four searcher robots. The one-robot case was also performed for comparison purposes.

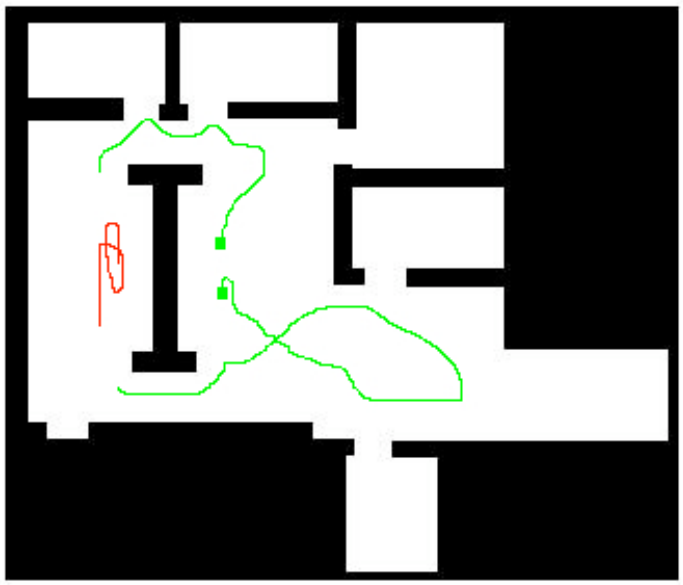


Figure 12: The map of an office environment used in Experiment 1. Example robot trajectories are shown, with rectangles indicating the robot start positions. The two longer lines are the searcher robots, the shorter line is the evader robot. These trajectories are from the third experimental trial. The searcher robots approach and trap the evader from opposite sides.

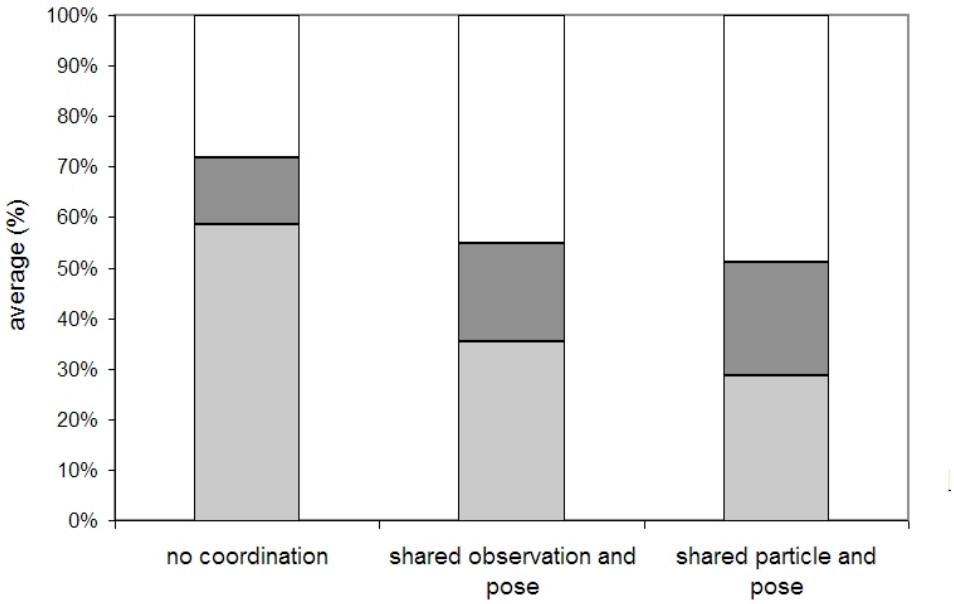
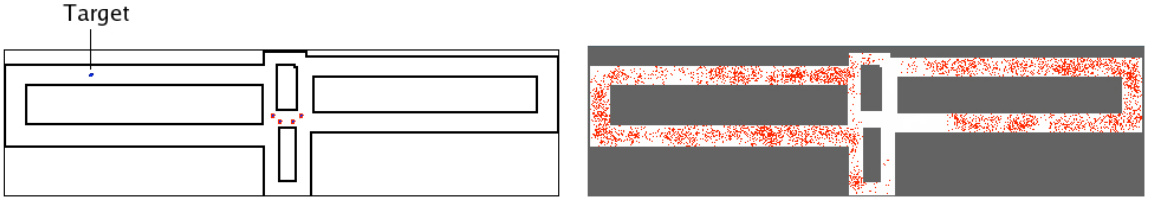


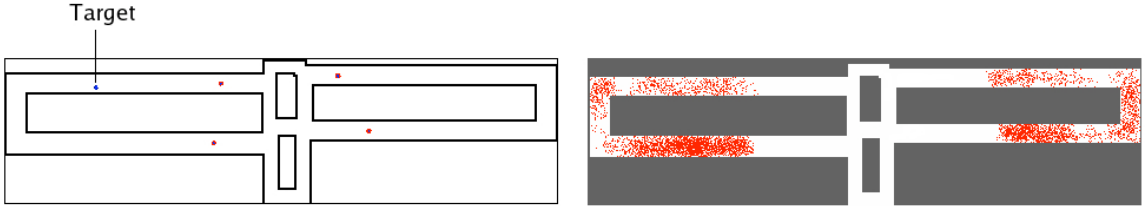
Figure 13: No coordination, Shared observation and pose and Shared particles (from left to right) are three cases shown in this diagram. The light gray area shows the average percentage of time an agent has spent before visiting the object for the first time. The dark gray area is the average percentage of time that the robots had no observation after they see the intruder for the first time.

Table 2: The total time that the searchers had no observation of the target.

Trial Type	Mean (sec)	σ	T-test p-value
No coordination	215.8	70.3	—
Shared Observation and Pose	164.6	63.7	0.0208
Shared Observation, Pose and Particles	153.4	53.3	0.0046



(a) $t = 0$



(b) $t = 50\,sec$

Figure 14: (a) Start configuration of robots in experiment B2 in which four robots try to locate an intruder. Left pictures are Stage snapshots and right pictures are plots of the map and particle cloud of one robot. (b) Robots configuration after 50 seconds. Robots have dispersed by taking one corridor each, rapidly reducing the uncertainty in the target position estimate, though no robot has yet seen the target.

The initial positions of searchers and the target are random. The target moves autonomously using the simple controller described above. Its stochastic nature makes the makes the target's movements unpredictable in the long term. The environment for this experiment is a model of our office building (SFU TASC) shown in Figure 14. Its topology is complex, making the search for a moving target difficult.

The map dimension is $70m \times 19m$ and the size of each cell in the occupancy grid is $15cm \times 15cm$. Each robot carries a laser range scanner with field of view of up to 8 meters at 180 degrees in its front. Fiducial-finders are used to detect the target. The searcher robots move at $0.2ms^{-1}$ and the target moves at $0.3ms^{-1}$. The pseudo-random number generator seed is the same for corresponding trials. Therefore, the initial positions and traversed path of the target were the same in the first trial of all of the experiments.

The experiments start with randomly distributed particles on the map and the searchers continue until the intruder is found or a time limit of 400 seconds is reached. Initially randomly distributed particles is the worst case for the tracker which means the searchers have had no idea where the intruder is located until the first observation is made.

Table 3: Wilcoxon test results for checking the difference between coordinated and non-coordinated cases for different population sizes.

Number of Robots	z ratio	p value	Sum of Neg. Ranks	Sum of Pos. Ranks
Four	-4.518	0.000	59.00	682.00
Three	-2.540	0.011	234.50	626.50
Two	-0.942	0.346	475.00	653.00

Table 4: Pairwise Wilcoxon test results to check if there is any advantage in increasing the population size.

N of Robots	z ratio	p value	Sum of Neg. Ranks	Sum of Pos. Ranks
4 vs. 3	-4.283	0.000	170.50	1005.50
4 vs. 2	-4.949	0.000	105.50	1070.50
4 vs. 1	-6.144	0.000	1.00	1274.00
3 vs. 2	-2.241	0.025	369.50	806.50
3 vs. 1	-4.614	0.000	159.50	1115.50
2 vs. 1	-3.171	0.002	309.00	966.00

5.4.1 Results

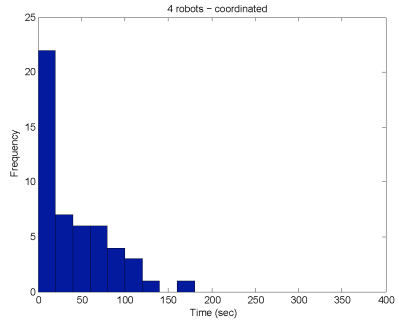
The results show that in the coordinated robots cases, the robots effectively distribute the search space amongst themselves. In practice this means they spread out, reducing the chances of spatial interference and increasing the effective sensor coverage area. While this heuristic method offers no optimality guarantees, we can informally state that in practice the robots are observed to perform the optimal action (Figure 14) in the scenarios we examined.

Histograms of the time taken to locate the target using each method are shown in Figure 15. Statistical tests were performed to check if there is any significant difference between coordinated and non-coordinated cases and also to examine if increasing number of robots has any effect on the performance.

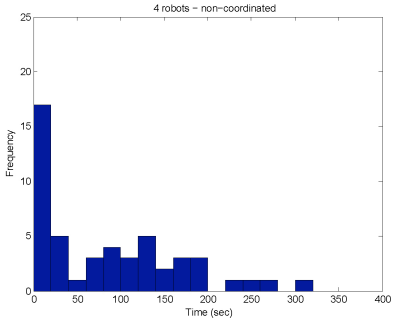
We chose the Wilcoxon Signed-Rank test (Daniel, 1978) to analyze the data. This test is equivalent to the T-test for non-parametric data. In summary, this test ranks the absolute differences between paired data, then it examines if the median of the differences is greater or less than zero according to the signed sum of the ranks. If the p-value is less than 0.05, we can conclude that there is a significant difference between two distributions.

As Table 3 shows, our data show a statistically significant difference between non-coordinated and coordinated cases for four and three robots but the coordination strategy does not improve the performance of tracking for two robots in this environment. We believe that the reason for no improvement seen in this two-robot case is because the task is very difficult for two robots due to the size and complexity of the environment. We argue that this would be true for any method, given the opportunities for the target to slip past two robots into previously-cleared areas.

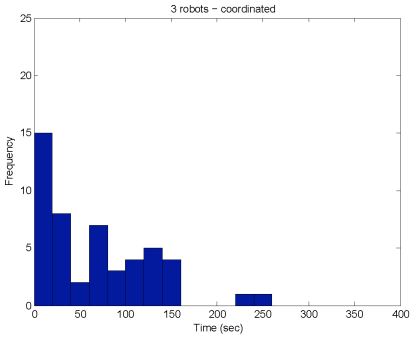
We compared the results of the experiments for different numbers of robots for coordination case. For each pair of population sizes, we aim to establish whether the performance data is drawn from distributions with significant difference. Table 4 shows there is a significant statistical difference among the results of the experiments for different number of robots which means increasing the number of robots has improved the performance of cooperative tracking.



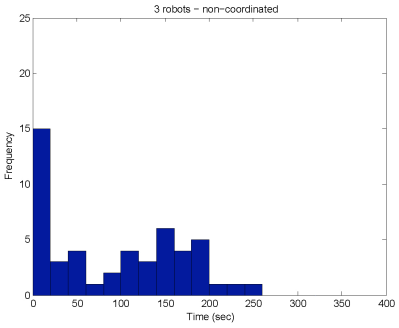
(a) 4-robot coordinated



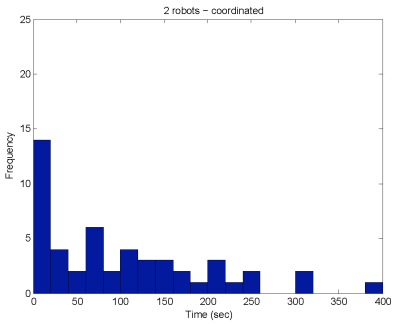
(b) 4-robot non-coordinated



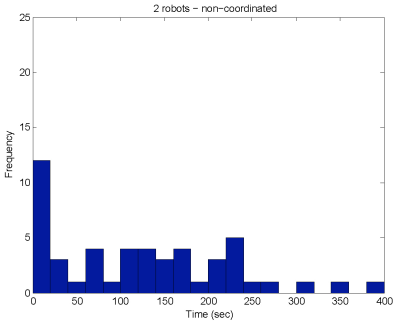
(c) 3-robot coordinated



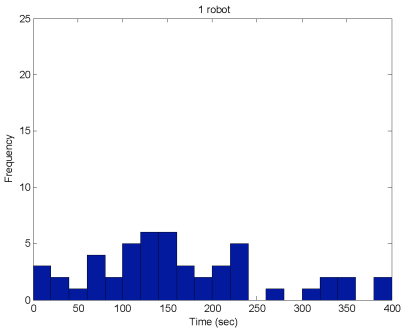
(d) 3-robot non-coordinated



(e) 2-robot coordinated



(f) 2-robot non-coordinated



(g) 1-robot

Figure 15: Histograms for the time spent for finding the target in the experiments in which the robots were supposed to locate an intruder in SFU TASC map with and without a coordination strategy.

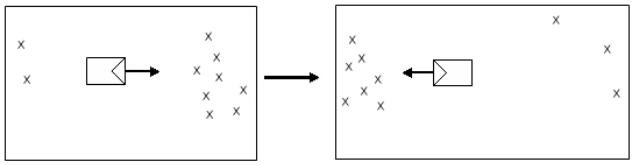


Figure 16: Illustration of the cyclical potential field problem. Observing the largest particle cloud causes the other cloud to grow, creating an oscillation in robot behavior.

6 Discussion

This section discusses some of the essential limitations and tricky issues encountered when implementing the method, and suggests some modifications and extensions.

6.1 No optimality claim

The main limitation of the method is that it is essentially heuristic and is not guaranteed to find optimal, or indeed any, solutions. In practice we observe that the method generates appropriate robot behavior in all the cases we have tried, including the experiments described above. Further, this approach is one of the (or perhaps the) most efficient approaches to the multi-robot tracking task so far described. As mentioned in the related work section, there are some optimization frameworks such as POMDPs that can deliver the optimal solution for this problem, but they are difficult to apply in practice since they are computationally intractable.

6.2 Dynamic loops in the potential field

A second and related criticism of this method is that it is theoretically subject to cyclical local optima. An example case is shown in Figure 16 where the problem is that the robot is positioned half-way between a large and a small cloud of particles. The robot is initially attracted toward the larger cloud, but as it approaches and clears the closest particles, the particle update rules cause the smaller cloud to grow. This contunues until the relative sizes of the two clouds are inverted, at which point the robot changes direction and heads for the now-larger cloud. Under certain conditions, this cycle can repeat forever.

This effect can cause an oscillatory behaviour in the navigation of the robot. We applied a simple adhoc method to make this behaviour very unlikely: we add a strongly weighted vector in the direction of the robot's current heading to the vector sum that steers the robot. This creates a behavioural "intertia" that tends to damp any undesirable oscillations.

This problem is common to many potential field approaches. While potential functions that have only a single instantaneous optimum are well known, for example harmonic functions (Connelly et al. 1993), we know of no approach that can provably eliminate such cycles over time in a dynamically generated potential field.

In practice it is likely that *ad-hoc* methods could be devised to recognize some periodic robot behaviours as they occur. This problem is not unique to our method, and good solutions would be widely useful.

6.3 Full communication assumption

The method relies on fully-connected communication between robots. This may be impossible in some situations. Although the robots can navigate based purely on their local sensor data, without communication the efficiency of coordination and therefore the performance of tracking will degrade.

One possible extension to this work is that each robot localizes teammate robots in its own coordinate frame using sensor data, possibly enhanced by intermittent communication. Recent literature on localization suggests that this is feasible. For example, (Fox et al., 2006) assume a probability distribution for teammate robots that are not in communication range to improve the task of exploring an unknown environment. (Howard et. al, 2002) describe a method for robots to achieve a maximum likelihood estimate of their mutual relative localization, given occasional mutual observations or commmunications.

To achieve coordination, each robot would self-assign the particles that are *likely* to be closer to itself than to any other teammate. In this case, the performance of the system will not change significantly if one robot loses communication with other robots.

6.4 Cost of the Flood-Fill method

As described in section 3, the magnitude and direction of particle forces are computed using a flood-fill method. The time complexity of this method for calculation of map distances is proportional to the size of the map and occupancy grid cells. Large, high-resolution maps could lead to unacceptable computation time. If speed is critical, we can trade memory space for compute time and use a pre-computed lookup table for every possible position of a robot on a map. Other alternatives exist between these extremes, and heuristic search methods such as A* could find map distances without necessarily filling the entire grid.

6.5 Potential to optimize searcher start positions

The time taken for a team of searchers to locate a target of initially unknown location is highly dependent on the searcher start positions. For any given population of searchers and map, there will be one or more sets of searcher start locations that gives the lowest expected time before observing the target. A potentially interesting extension of this work would be to automatically find good start positions *a priori*.

6.6 Adaptive target motion model

In our implementation, the target motion model was very simple, based on the last-observed target velocity. This is a reasonable model of a mobile robot target, but may be a poor model of a human target, such as a lost museum visitor or a shifty burglar. An extension of this work that we plan to pursue is to estimate the parameters of the motion model online based on observations of the target's behaviour. For instance, vision-based activity recognition methods can be used to estimate the current activity of humans in broad categories such as running, walking and standing (Efros et al., 2003). We hope that adjusting the motion model parameters accordingly (or switching between several motion models) could be shown to increase tracking performance.

6.7 Tracking multiple targets

The method can be trivialy extended to track multiple targets by maintaining a seprate particle filter for each target. If the combined set of all particles is allocated among all searchers, then the task is divided purely spatially. If instead we wish one robot to be assigned to each target, then we can assign to each target the searcher that has the minimum total map distance to the target's particle cloud. It would be interesting to see what other allocation strategies could be devised.

6.8 Video game non-player character motion planning

One potentially useful application of the method is in motion planning for non-player characters (NPCs) in video games. Using this technique, an NCP opponent could move to intercept a player based on what she was last observed (by the NPC) to do. Convincing anticipatory behaviour in

NPCs is hard to achieve, but looks particularly 'intelligent' to the player, making the game more engaging. Similarly, intelligent-looking coordination is also attractive but rare in current games. As the amount of CPU power available for NCP AI grows, we expect methods like this to find commercial application in the short term, and we are working in this area.

6.9 Generality of integrated particle filter and potential field method

We have described in detail the application of the integrated particle filter and potential field to multi-robot target tracking. But this method is novel in itself and can be applied to other problems. It is a natural approach when the distribution modelled by the filter is directly navigable by a robot or end effector. More abstractly, the technique is applicable when we wish to control a system to reach the same point in state space as an observed system.

For example, consider an application in human-computer interaction, where the computer models the affective state of the human as a distribution in the space with axes [Arousal, Valence, Stance], instead of the single point in this space used in the social robot Kismet (Breazeal 2003). A potential field created from this particle cloud can drive the computer's current affect to track the human's, robustly achieving the socially important affect-matching.

7 Conclusion

We have described a method for a team of mobile robots to cooperatively track a moving target. This approach addresses the main limitation of previous approaches in that it actively reduces the uncertainty caused when the target is occluded for long periods.

Our method includes a novel integration of the particle filter and potential field, which can be applied in a variety of tracking applications. In this application, the particle filter represents a probability distribution over the possible pose in (x, y, θ) of the target. A potential field is generated using the particle cloud directly as input - no clustering of particles is performed, so that no information is lost. Each particle exerts an attractive potential on the searcher robot, so that the resulting potential guides the searcher to visit particle locations with its sensors. Multiple searchers can be coordinated by allocating a subset of the particle cloud to each robot. We used a simple nearest-robot filter to achieve this.

Under certain (as yet undefined, but probably practical) assumptions about the number of searchers, the nature of the map, and the relative speeds of the searcher and target, the entropy in the particle cloud can be reduced and the target located. The qualitative behavior of the seacher robots is attractive, with the robots splitting up to search for targets from opposite directions.

This heuristic method is simple to implement and produces effective searching behaviour in single and multiple robots, and could also be useful in other applications.

References

[1] Breazeal, C. 2003. "Towards sociable robots," T. Fong (ed),*Robotics and Autonomous Systems*, 42(3-4), pp. 167-175.

[2] Bruce, A. and Gordon, G. 2004. Better Motion Prediction for People-Tracking. In *IEEE International Conference on Robotics and Automation.*

[3] Connolly, C. and Grupen,R. "On The Application of Harmonic Functions to Robotics," Journal of Robotic Systems, 10(7):931-946.

[4] Daniel, W. W. *Applied Nonparametric Statistics* (Houghton Mifflin Company, 1978) page 135.

[5] Efros, A. A., Berg, A. C., Mori, G., and Malik, J. 2003. Recognizing Action at A Distance. In *Proceedings of the 9th International Conference on Computer Vision.* vol 2, pp. 726–733.

[6] Fox, D., Ko, J., Konolige K., Limketkai B., Schulz, D., and Stewart, B. 2006. Distributed Multi-Robot Exploration and Mapping. In *Proceedings of the IEEE*.

[7] Gerkey, B. P., Vaughan, R. T., Stoy, K., Howard, A., Sukhatme, G. S., and Mataric, M. J. 2001. Most Valuable Player: A Robot Device Server for Distributed Control. In *IEEE/RSJ International Conference on Intelligent Robots and Systems.* pp. 1226–1231.

[8] Gerkey, B. P., Thrun, S., and Gordon, G. 2004. Visibility-based Pursuit-Evasion with Limited Field of View. In *Proceedings of the National Conference on Artificial Intelligence (AAAI 2004).* pp. 20–27.

[9] Howard, A., Matarić, M. J., and Sukhatme, G. S. 2002. Localization for mobile robot teams using maximum likelihood estimation. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp 434–459.

[10] Isard, M. and Blake, A. 1998. Condensation - Conditional Density Propagation for Visual Tracking. *International Journal on Computer Vision*, 29(1):5–28.

[11] Jung, B. and Sukhatme, G. S. 2004. A Generalized Region-based Approach for Multi-target Tracking in Outdoor Environments. In *IEEE International Conference on Robotics and Automation*, pp. 2189–2195.

[12] Khatib, O. 1986. Real-time Obstacle Avoidance for Manipulators and Mobile Robots. *International Journal of Robotics Research*, 5(1):90–98.

[13] LaValle, S. M., Lin, D., Guibas, L. J., Latombe, J.-C., and Motwani, R. 1997. Finding an Unpredictable Target in a Workspace with Obstacles. In *IEEE International Conference on Robotics and Automation*. pp. 737–742.

[14] Lioa, L., Fox, D., Hightower, J., Kautz, H. and Schulz, D. 2003. Voronoi Tracking: Location Estimation Using Sparse and Noisy Sensor Data. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*. pp. 723–728.

[15] Montemerlo, M., Whittaker, W., and Thrun, S. 2002. Conditional Particle Filters for Simultaneous Mobile Robot Localization and People-tracking. In *IEEE International Conference on Robotics and Automation*. pp. 695–701.

[16] Mottaghi, R. and Vaughan, R. T. 2006. An Integrated Particle Filter & Potential Field Method for Cooperative Robot Target Tracking. In *IEEE International Conference on Robotics and Automation.* pp. 1342–1347.

[17] Rosencrantz, M., Gordon, G., and Thrun, S. 2003. Locating Moving Entities in Dynamic Indoor Environments with Teams of Mobile Robots. In *Proceedings of Autonomous Agents and Multi-Agent Systems*. pp. 233–240.

[18] Roy, N., Gordon, G., and Thrun, S. 2005. Finding Approximate POMDP Solutions Through Belief Compression. *Journal of Artificial Intelligence Research*. 23:1–40.

[19] Schulz, D., Burgard, W., Fox, D., and Cremers, A. B. 2003. People Tracking with a Mobile Robot Using Sample-based Joint Probabilistic Data Association Filters. *International Journal of Robotics Research (IJRR)*. 22(2):99–116.

[20] Stroupe, A., Ravichandran R., and Balch, T. 2004. Value-Based Action Selection for Exploration and Dynamic Target Observation with Robot Teams. In *IEEE International Conference on Robotics and Automation*. pp. 4190–4197.

[21] Thrun, S., Fox, D., Burgard, W., and Dellaert, F. 2001. Robust Monte Carlo Localization for Mobile Robots. *Artificial Intelligence*, 128(1-2):99–141.

[22] Thrun, S. 2002. Robotic Mapping: A Survey. In G. Lakemeyer and B. Nebel, editors, *Exploring Artificial Intelligence in the New Millennium.* Morgan Kaufmann.

[23] Ulrich, I. and Borenstein, J. 1998. VFH+: Reliable Obstacle Avoidance for Fast Mobile Robots.In *Proceeding of the IEEE International Conference on Robotics and Automation*, pp. 1572–1577.