

# Multiple Target Tracking for Surveillance: A Particle Filter Approach

#Punarjay Chakravarty<sup>1</sup>, Ray Jarvis<sup>2</sup>

<sup>1,2</sup>ARC Centre for Perceptive and Intelligent Machines in Complex Environments

Department of Electrical and Computer Systems Engineering

Monash University, Victoria 3800, Australia

{punarjay.chakravarty,ray.jarvis}@eng.monash.edu.au

## Abstract

*This paper describes a system that uses multiple particle filters to track an unknown number of targets from range data. The tracked targets do not move with constant velocity or acceleration, which fits the requirements of a system that tracks the movement of people. Results from simulations involving tracking of 10 manoeuvring targets in clutter are presented.*

## 1. INTRODUCTION

The work presented in this paper is part of a larger research effort undertaken at PIMCE (Perceptive and Intelligent Machines in Complex Environments), to track and identify people in an indoor environment using both fixed and mobile robot-based sensors.

Since the 1960s, a lot of research [1] for tracking multiple moving targets has focussed on military applications where the targets are projectiles or aircraft moving under the constraints of constant velocity or constant acceleration. The tracking algorithm of choice for these applications is the well-known Kalman Filter. When the targets to be tracked are people, this assumption breaks down, because people often change their direction of motion. This necessitates the need for a tracker with a motion model that can accommodate non-linearity in the motion of a tracked individual.

A tracker that uses a dynamically switching motion model to track people has been developed by [2]. The system tracks a single person in the presence of occlusion from Laser Range Finder data in a Multiple Hypothesis framework [1]. The Multiple Hypothesis Tracker (MHT) does not associate measurements with targets immediately, but maintains a set of measurement-to-target hypotheses as an exponentially growing tree of Kalman Filters. The most likely sequence of hypotheses is chosen after a number of frames have elapsed, in an N-scan back technique. This procedure makes the MHT more robust to occlusions and missing data. However, if it is to be used for multiple targets, separate trees need to be maintained and clustered (if the targets are close together). The complexity of the MHT grows exponentially with the number of targets to be tracked.

The Particle Filter [3] tracks a target by maintaining a set of weighted particles which approximate the target probability density function. Arbitrary densities (not restricted to single-mode or Gaussian as in the Kalman Filter) can be tracked, and the larger the number of particles, the better is this approximation of target density. In theory, because the particle filter can track a multi-modal density, it should be able to track multiple targets with a mode for each target. However, Schulz *et al.* [4] report a simulation (1 particle filter tracking 2 targets) that shows that as soon as one object is occluded, particles following that target get lower weights and subsequently, most of the particles are hijacked by the other target. This has been confirmed by simulations conducted for this paper.

This suggests the requirement of a particle filter for each target. The complexity of this approach is linear in the number of targets:  $O(Nn)$  where  $N$  is the number of targets and  $n$  is the number of particles per target. This is the approach that has been used in this paper.

Section 2 details the mechanism for the particle filter based multi target tracker. Simulation results involving tracking of 10 targets are given in Section 3.

## 2. TRACKING MECHANISM

$N$  particle filters are used to track  $N$  targets. Each filter consists of a set of  $n$  particles. Each particle at time  $t$  has a state  $\hat{x}_t$ , which evolves according to the following equation:

$$\hat{x}_t = A\hat{x}_{t-1} + v_t \quad \text{Equation 1}$$

where  $A$  is the state transition matrix and  $v_t$  is the system noise. Each particle also has a probabilistic weight associated with it.

For each filter, apart from the particle states and their associated weights, the following data is maintained:

1. The mean position of all the particles in the filter.
2. The standard deviation of the particles from the mean position.

3. The number of times in the past  $\tau$  frames that the filter's cumulative un-normalized probability has fallen below a threshold.

#### A. Data Association

An important issue in multiple target tracking is the method of association of measurements to the filters. The simplest way to do this is Nearest-neighbour data association, which associates a measurement to its nearest filter. The Joint Probabilistic Data Association (JPDA) method is a more complex method, which considers all allowable measurement-to-target associations in the update of the filter state and prevents the hijacking of 2 or more filters by a single target. However, this approach requires the number of targets to be known before-hand. Karlsson and Gustafsson [5] give a comparative study between the 2 above approaches of data association.

In this paper, the association of the measurements from the latest frame with the filters is done like [6], in a modified nearest neighbour fashion because it is simple to implement and computationally less demanding than the JPDA.

A measurement is assigned to the filter with the closest mean position, only if it is within 1.5 standard deviations of that filter. Measurements already associated with a filter are not associated again with any other filter, in what is known as a "hard" assignment. If a "soft" assignment procedure is followed, it has been observed from the simulations, that 1 target tends to capture more than 1 particle filter when the filters are close together.

Once a measurement has been assigned to a filter, the weights of all the particles in that filter are updated based on their distances from the measurement. A Gaussian centred on this distance is used as a probabilistic look-up table to update the weight of the particle as shown in Figure 1.

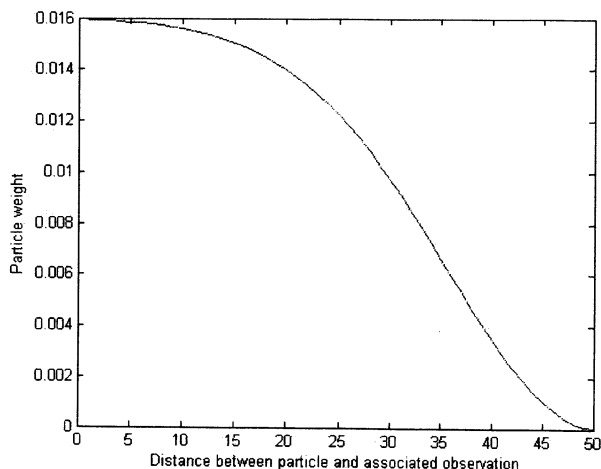


Figure 1: Gaussian centred on distance between particle and associated observation is used as a look-up table to calculate particle weight

#### B. Interaction Weights

The interaction weights (inspired from the MRF motion model described by Khan *et al.* [7]) are required to be calculated when 2 or more targets are close to each other. It has been noticed from the simulations that particles from filters following targets close to each other tend to switch targets. This is mitigated by calculating an interaction weight for each particle to be a function of its distance from the mean position of another filter. An inverted Gaussian (Figure 2) centred on the distance is used as a look-up to calculate this weight, which penalizes particles that are closer than a threshold to another filter.

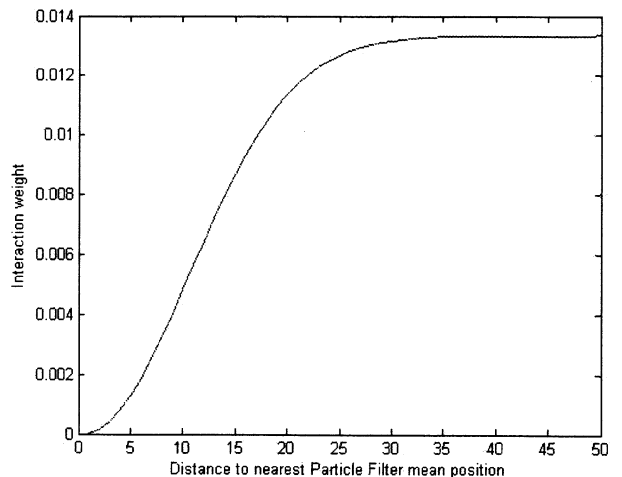


Figure 2: Inverted Gaussian used as look-up to calculate interaction weight

#### C. Clustering

The measurements from the past  $\tau$  frames are clustered to discern the true targets from the false alarms. This procedure (adapted from [8]) is outlined next:

Given a set of observations  $\{y_{t'}\}_{t'=1}^{\tau}$  over the past  $\tau$  frames...

Set  $t' \leftarrow 1, m \leftarrow 1$ .

1. Select a measurement  $y_a = y_{m,t'=1}$  with  $m = 1, 2, \dots, M_{t'}$  to search for  $k^{\text{th}}$  cluster  $s_t^k$ . Initialize the exemplar of this  $k^{\text{th}}$  cluster as  $y_a$ .
2. Group  $y_a$  with all  $y_b = y_{l,t'+1}$  for all  $l = 1, 2, \dots, M_{t'+1}$  such that  $y_b$  is within a threshold distance of  $y_a$ . Find  $y_b$  that is the farthest away from  $y_a$  and also within the threshold distance, and replace the exemplar for this cluster with  $y_b$ . If no points within the threshold distance are found, do not change the exemplar.
3. Goto step 2, with  $t' \leftarrow t'+1$ .

4. When all the  $\tau$  frames have been visited, a cluster  $s^k$  is found, only if the number of points in this cluster is greater than a pre-defined threshold.
5. Goto step 1 with  $m \leftarrow m + 1$ .

This gives a set  $S_\tau = \{s_\tau^k\}_{k=1}^K$  of clusters with corresponding exemplars. It has been observed in the simulations, that sometimes, a number of clusters end up with the same exemplar and so, duplicate clusters have to be deleted.

#### D. Track Initiation

The track initiation procedure uses measurements from the past  $\tau$  frames as input into the clustering algorithm described in the previous section. When the multi-target tracker is switched on initially, all the measurements in the first  $\tau$  frames are used. Subsequently, only the measurements that have been un-associated with any particle filters in the past  $\tau$  frames are taken. The particle filters are initialized at the exemplar positions for the set

$$S_\tau = \{s_\tau^k\}_{k=1}^K \text{ clusters.}$$

#### E. Track Deletion

The track deletion procedure uses the cumulative un-normalized probability, which is maintained for each filter as a measure of its confidence in tracking a target. If this cumulative weight falls below a pre-defined threshold for a majority of the last  $\tau$  frames, then the particular filter is deactivated.

#### F. Track Continuation

The track continuation procedure implements the following steps:

1. Prediction
2. Update
3. Re-sampling

In the prediction step, the particles for each filter are evolved according to Equation 1.

The update step consists of associating the latest data from the measurements with the filters and updating the probabilities of their particles based on these associations (outlined in the data association sub-section earlier). In addition, each particle probability is multiplied by an interaction factor (explained in sub-section B) if it is closer than a pre-defined threshold to the mean position of another filter. The particle probabilities are then normalized for each filter.

Finally, the particles for each filter are re-sampled, with replacement according to their weights. Particles with higher weights have a higher probability of getting propagated to the next iteration. Systematic re-sampling, described in [3] is used to re-sample the particles. This algorithm has  $O(n)$  complexity, where  $n$  is the number of particles per filter.

#### G. Multi-Target Tracker Algorithm

The Multi-Target Tracker, which starts tracking the targets at frame  $\tau$ , can be summarized as follows:

1. If frame number =  $\tau$ , goto step 2.  
Else if frame number  $> \tau$  and not a multiple of  $\tau$ , goto step 3.  
Else if frame number  $> \tau$  and is a multiple of  $\tau$ , goto step 4.
2. Start track initiation procedure with all the measurements from the past  $\tau$  frames. Goto step 1.
3. Apply track continuation procedure. Goto step 1.
4. Apply track deletion procedure, followed by the track initiation procedure, with only the un-associated measurements from the past  $\tau$  frames. Goto step 1.

### 3. SIMULATIONS

Each particle at time  $t$ , is modelled by the state vector

$\hat{x}_{it} = [x_{it}, y_{it}]$ , where  $i$  indicates the particle filter index and  $(x_{it}, y_{it})$  its position.

To simulate the random changes in the direction of movement of people, the targets are propagated anisotropically, in an area of size 640 by 480 pixels. Target movement between frames is drawn from a normal distribution centred around its position in the previous frame. In addition, some false alarms are introduced with a uniform distribution in every screen.

Readings as would be obtained by a Laser Range Finder (LRF) placed in the centre of the surveillance region are simulated. This is done by converting the measurement coordinates from cartesian  $(x, y)$  to polar  $(r, \theta)$ , and binning them in 1 degree angle increments, from 0 to 360 degrees. For each bin, only the measurement corresponding to the least  $r$  is kept, and all the rest discarded. This simulates the effect of targets nearer to the LRF occluding the ones further away. This can be seen in Figure 3, where a target painted orange in the lower right is occluded by another target. Thus, it is not always possible to resolve all the targets and this aims to be an accurate simulation of LRF readings. The target positions are filtered through the above procedure before they are given to the Multi-Target Tracker.

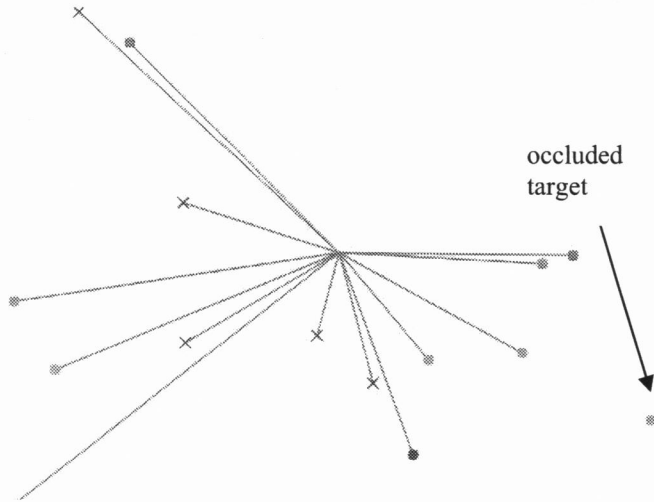


Figure 3: Simulated readings from a Laser Range Finder. Targets are indicated by filled-in circles and false alarms by crosses. The sensor is placed in the centre of the surveillance region.

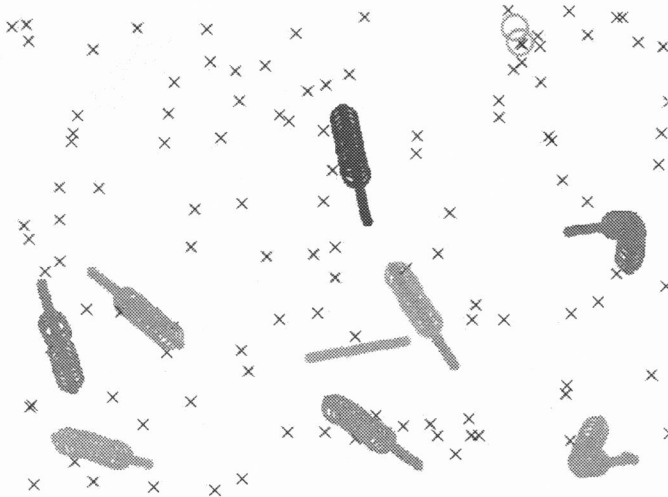


Figure 4: Track initialization: Coloured lines indicate the track of the targets over 25 frames and circles indicate mean positions of the particle filters initialized at the end of the 10<sup>th</sup> and 20<sup>th</sup> frames ( $\tau = 10$ ). Crosses denote deliberately introduced false alarms. Note: Targets moved linearly for illustration purposes

Next, the choice of parameters in target movement and the Multi-Target Tracker are indicated.

Simulation parameters:

1. Number of targets = 10
2. Number of false alarms per frame = 5
3. Angular resolution of LRF = 1 deg
4. Target motion parameters:  $\sigma_x^2 = 10, \sigma_y^2 = 10$

Multi-target tracker parameters:

1. Maximum number of allowed particle filters = 10
2. Number of particles per filter = 700
3. Particle predict step: particles are evolved according to  $\hat{x}_t = A\hat{x}_{t-1} + \nu_t$

where

$$A = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix},$$

$$\nu_t = \begin{bmatrix} \nu_x \\ \nu_y \end{bmatrix}, \nu_x \sim N(0,15), \nu_y \sim N(0,15)$$

4. Particle update step:  
The variance of the Gaussians used in data association and the interaction factor for updating the particle weights are set at 15 and 10  
The interaction distance threshold is set at 25
5. Clustering step:  
Number of past frames used for clustering,  $\tau = 10$   
Minimum distance threshold for points to be clustered = 25  
Minimum number of points in a cluster for a filter to be initialized around its exemplar = 3
6. Track deletion:  
Minimum threshold for cumulative un-normalized probability is 3. This threshold will depend on the number of particles/target.  
Number of times in the past previous  $\tau$  frames this threshold has to be crossed for the filter to be de-activated. This parameter depends on  $\tau$  and is set at 8

Most of the aforementioned parameters are determined empirically. Because the system will ultimately be used to track people, the target motion parameters are based on the average walking speed of a person. Assuming this to be 1.6 km in 20 min (1.34 metres/sec), and assuming a minimum LRF update rate of 10 Hertz, the movement of a person between successive scans should not exceed 0.134 m. Given that the range of the average LRF is about 50 m, this ratio of movement between frames and maximum range translates to about 1 pixel for a maximum range of 400 pixels (distance between centre and corner of the 640 x 480 surveillance region). It is found that the system can track targets that moved up to 10 pixels between frames. This is well above the required 1 pixel movement between frames.

It is observed that the variance of the Gaussian used to move the particles in the predict step has to be higher than the variance of the Gaussians used in target motion for successful tracking.

In reality, people will not occupy the same spot on the surveillance region, but in this simulation, the targets are moved independently and sometimes chance to be in the same spot. When this happens, trackers sometimes switch targets. However, when the targets pass within 5 pixels

(translates to 60 cm, using the previously discussed ratio of movement between frames and maximum range of the LRF), of each other, without actually occupying the same space, the trackers do not switch and continue tracking the targets successfully after they cross.

The performance of the system is tested by evaluating the root mean square error (RMSE) and the number of track failures while varying the number of particles per filter for 10 targets, over the course of 300 frames. All the other parameters are fixed as described earlier in this section.

A track failure is registered whenever the track deletion procedure de-activates a particle filter. The RMSE is calculated as:

$$RMSE = \sqrt{\frac{1}{NT} \sum_{t=1}^T \sum_{i=1}^N \|x_{i,t} - \hat{x}_{i,t}\|^2}$$

where  $x_{i,t}$  is the position of the target  $i$  at time  $t$  and  $\hat{x}_{i,t}$  is the mean position of the particle filter tracking it,  $N$  is the number of targets and  $T$  is the total number of frames in the sequence.

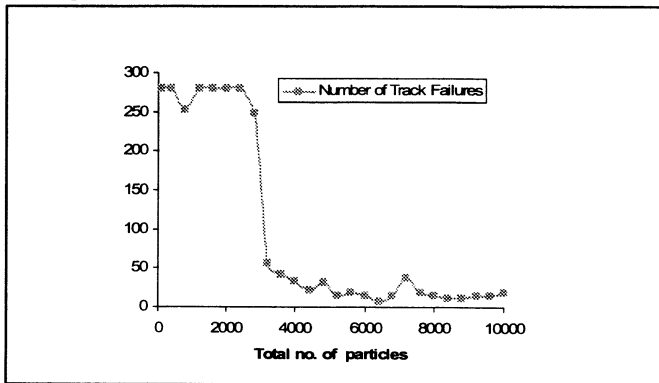


Figure 5: Total number of particles versus number of track failures over 300 frames

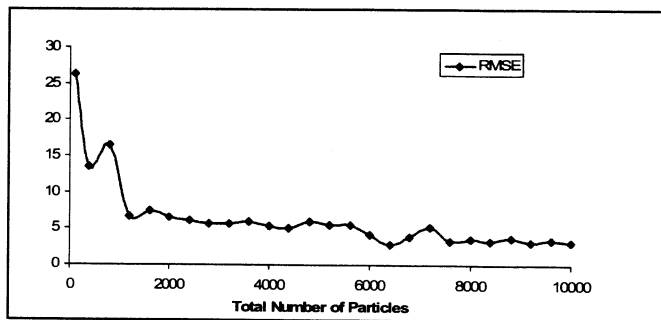


Figure 6: Total number of particles versus RMSE over 300 frames

It can be observed from Figure 5 that the number of track failures decreases drastically at about 3600 particles (360 particles per target), while Figure 6 suggests that the RMSE decreases to about 5 and remains relatively constant from 1200 particles upwards. There is a slight undulation in the graph illustrating track failures for

particle numbers greater than 4000, even though the overall trend is towards lesser track failures for larger particle numbers. This is because track failures mostly happen with targets occupying the same spot in the simulation. Since the movement of targets is different in every simulation, this number will depend on the number of times different targets chanced upon the same spot in the particular simulation run. The simulations suggest that around 400 particles/target is sufficient to track the targets.

A C implementation of the multi-target tracker (including generation of the target motion and the on-screen visualization of targets and particle filter mean positions) runs at about 30 frames/sec with 3600 particles (360 particles/target) on a 1.6 GHz Pentium M laptop.

The simulations are available from the following website: <http://users.monash.edu/~pcha25/ISSNIP05.htm> A screenshot from the same is shown in Figure 7.

#### 4. CONCLUSIONS

In this paper, a real-time system for tracking an unknown number of targets in clutter has been presented. The system incorporates a clustering algorithm to discern legitimate targets from clutter. It includes track deletion and initiation procedures for tracking new targets and removing under-performing tracks. Simulations show that around 400 particles per target is enough for successful tracking.

Future work will involve testing using actual range data in indoor environments and fusion of range data with images from cameras to maintain the identities of the tracked individuals.

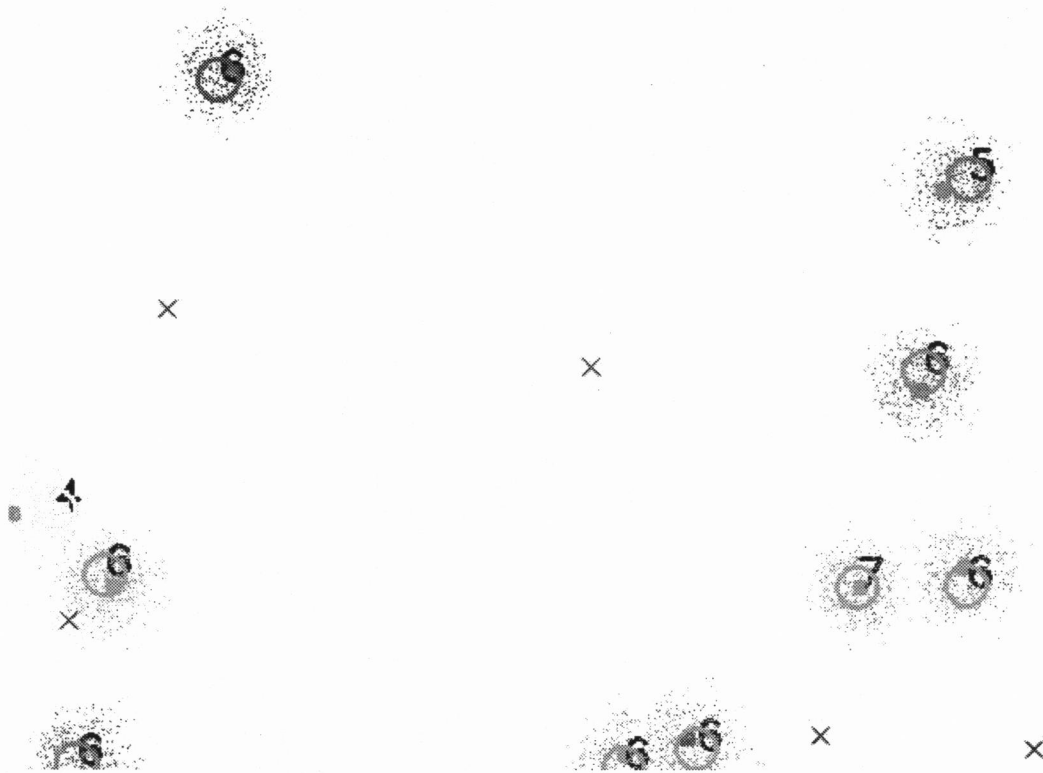


Figure 7: Frame from the simulation sequence showing 10 targets (filled-in coloured circles) being tracked in the presence of clutter. Particle filter mean positions are shown as empty circles of the same colour as the particles. False alarms are indicated by crosses. The number next to each filter is the cumulative sum of the un-normalized weights of the filter and indicates the confidence of the filter in tracking the target. Different colours are used to draw the circles to differentiate the filters and targets.

## REFERENCES

- [1] Y. Bar-Shalom and T. E. Fortmann, *Tracking and Data Association*. San Diego: Academic Press, 1988.
- [2] G. Taylor and L. Kleeman, "A Multiple Hypothesis Walking Person Tracker with Switched Dynamic Model," presented at Australasian Conference on Robotics and Automation (ACRA), Canberra, 2004.
- [3] S. Arulampalam, S. Maskell, N. J. Gordon, and T. Clapp, "A Tutorial on Particle Filters for On-line Non-linear/Non-Gaussian Bayesian Tracking," *IEEE Transactions of Signal Processing*, vol. 50, pp. 174-188, 2002.
- [4] D. Schulz, W. Burgard, D. Fox, and A. Cremers, "Tracking multiple moving targets with a mobile robot using particle filters and statistical data association," presented at IEEE International Conference on Robotics & Automation (ICRA), 2001.
- [5] R. Karlsson and F. Gustafsson, "Monte Carlo data association for multiple target tracking," presented at IEE Target tracking: Algorithms and applications, The Netherlands, 2001.
- [6] S. Thrun, M. Montemerlo, and W. Whittaker, "Conditional Particle Filters for Simultaneous Mobile Robot Localization and People-Tracking," presented at IEEE International Conference on Robotics and Automation (ICRA), Washington, DC, 2002.
- [7] Z. Khan, T. Balch, and F. Dellaert, "An MCMC-based Particle Filter For Tracking Multiple Interacting Targets," presented at European Conference on Computer Vision (ECCV04), 2004.
- [8] W. Ng, J. Li, S. Godsill, and J. Vermaak, "Multiple Target Tracking using Sequential Monte Carlo Methods and Efficient Data Association and Initialization," *IEEE Conference on Radar, Sonar & Navigation*, submitted 2004.