Tracking Multiple Moving Targets with a Mobile Robot using Particle Filters and Statistical Data Association

Dirk Schulz[1] Wolfram Burgard[2] Dieter Fox[3] Armin B. Cremers[1]

[1]University of Bonn, Computer Science Department, Germany

[2]University of Freiburg, Department of Computer Science, Germany

[3]University of Washington, Dept. of Computer Science & Engineering, Seattle, WA, USA

Abstract

One of the goals in the field of mobile robotics is the development of mobile platforms which operate in populated environments and offer various services to humans. For many tasks it is highly desirable that a robot can determine the positions of the humans in its surrounding. In this paper we present a method for tracking multiple moving objects with a mobile robot. We introduce a sample-based variant of joint probabilistic data association filters to track features originating from individual objects and to solve the correspondence problem between the detected features and the filters. In contrast to standard methods, occlusions are handled explicitly during data association. The technique has been implemented and tested on a real robot. Experiments carried out in a typical office environment show that the method is able to keep track of multiple persons even when the trajectories of two people cross each other.

1 Introduction

The problem of estimating the positions of moving objects is an important problem in mobile robotics. Knowledge about the position of moving objects can be used to improve the behavior of the system especially if the robot is deployed in populated environments. For example, this ability allows a robot to adapt its velocity to the speed of people in the environment and enables a robot to improve its collision avoidance behavior in situations in which the trajectory of the robot crosses the path of a human.

In this paper we present a method for tracking multiple moving objects with a mobile robot. This technique uses the robot's sensors and a motion model of the objects being tracked in order to estimate their position and velocities. We apply a variant of Joint Probabilistic Data Association Filters (JPDAFs) [1, 3], a very popular approach to tracking multiple moving objects. JPDAFs compute a Bayesian estimate of the correspondence between features detected in the sensor data and the different objects to be tracked. Virtually all existing approaches to tracking multiple targets apply Kalman filters to estimate the states of the individual objects. While Kalman filters have been shown to provide highly efficient state estimates, they are restricted to Gaussian distributions over the state to be estimated.

More recently, particle filters have been introduced to estimate non-Gaussian, non-linear dynamic processes [6, 14]. They have been applied with great success to different state estimation problems including visual tracking [2, 8], mobile robot localization [5] and dynamic probabilistic networks [9]. The key idea of particle filters is to represent the state by sets of samples (or particles). The major advantage of this technique is that it can represent multi-modal state densities, a property which has been shown to increase the robustness of the underlying state estimation process [7]. However, most existing applications deal with estimating the state of *single* objects only. One way to apply particle filters to the problem of tracking *multiple* objects is to estimate the combined state space, as proposed in [11]. Unfortunately, the complexity of this approach grows exponentially in the number of objects to be tracked.

Our approach combines the advantages of particle filters with the efficiency of existing approaches to multi-target tracking: It uses particle filters to track the states of the objects and applies JPDAFs to assign the measurements to the individual objects. This technique is similar to the one proposed in [11]. However, instead of relying on Gaussian distributions extracted from the sample sets, our approach applies the idea of JPDAFs directly to the sample sets of the individual particle filters. To increase the robustness of the overall approach, our method explicitely deals with occlusions. This way, our robot can reliably keep track of multiple persons even if they temporarily occlude each other.

This paper is organized as follows. After introducing a general framework for JPDAFs, we propose our sample-based version of JPDAFs in Section 2. In Section 3, we describe how to extract features required for the JPDAF from proximity information provided by a robot's laser range-finders. Furthermore, we show how to deal with occlusions. Section 4 describes several experiments carried out on a real robot and illustrates the capabilities and the robustness of our approach.

2 Sample-based Joint Probabilistic Data Association Filters

To keep track of multiple moving objects one generally has to estimate the joint probability distribution of the state of all objects. This, however, is intractable in practice already for a small number of objects since the size of the state space grows exponentially in the number of objects. To overcome this problem, a common approach is to track the different objects independently, using factorial representations for the individual states. A general problem in this context is to determine which measurement is caused by which object. In this paper we apply Probabilistic Data Association Filters (JPDAFs) [3] for this purpose. Furthermore, to avoid the Gaussian assumption of Kalman filter-based JPDAFs, we present a sample-based version of JPDAFs. In what follows we will first describe a general version of JPDAFs and then a sample-based implementation.

2.1 Joint Probabilistic Data Association Filters

Consider the problem of tracking T objects. $\mathbf{X}^k = \{\mathbf{x}_1^k, \ldots, \mathbf{x}_T^k\}$ denotes the state of these objects at time k. Note that each $\mathbf{x}_i^k$ is a random variable ranging over the state space of a single object. Furthermore, let $\mathbf{Z}(k) = \{\mathbf{z}_1(k), \ldots, \mathbf{z}_{m_k}(k)\}$ denote a measurement at time k, where $\mathbf{z}_j(k)$ is one feature of such a measurement. $\mathbf{Z}^k$ is the sequence of all measurements up to time k. The key question is how to assign the individual features to the objects.

In the JPDAF framework, a joint association event θ is a set of pairs $(j,i) \in \{0,\ldots,m_k\} \times \{1,\ldots,T\}$. Each θ uniquely determines which feature is assigned to which object. Please note, that in the JPDAF framework, the feature $\mathbf{z}_0(k)$ is used to model situations in which an object has not been detected, i.e. no feature has been found for object i. Let $\Theta_{ji} \in \Theta$ denote the set of all valid joint association events which assign feature j to the object i. At time k, the JPDAF computes the posterior probability that feature j is caused by object i according to

$$\beta_{ji} = \sum_{\theta \in \Theta_{ji}} P(\theta \mid \mathbf{Z}^k). \tag{1}$$

The probability $P(\theta \mid \mathbf{Z}^k)$ of an individual joint association event can be computed according to Eq. (3), where α is a normalizer. To derive Eq. (3), we make use of Bayes' rule and the assumption that the estimation problem is Markov.

$$\begin{aligned} P(\theta \mid \mathbf{Z}^k) &= P(\theta \mid \mathbf{Z}(k), \mathbf{Z}^{k-1}) & (2)\\ &\overset{\text{Markov!}}{=} P(\theta \mid \mathbf{Z}(k), \mathbf{X}^k) \\ &\overset{\text{Bayes!}}{=} \alpha\, p(\mathbf{Z}(k) \mid \theta, \mathbf{X}^k)\ P(\theta \mid \mathbf{X}^k) & (3)\end{aligned}$$

The term $P(\theta \mid \mathbf{X}^k)$ corresponds to the probability of the assignment θ given the current states of the objects. Throughout this paper we make the strong assumption that all assignments have the same likelihood so that this term can be approximated by a constant. Throughout our experiments we did not observe evidence that this approximation is too crude. However, we would like to refer to [4] for a better approximation of this quantity.

The term $p(\mathbf{Z}(k) \mid \theta, \mathbf{X}^k)$ denotes the probability of making an observation given the state of the objects and a specific assignment between the observed features and the objects. In order to determine this probability, we have to consider the case that a feature is not caused by any of the objects. We will call these features false alarms. Let γ denote the probability that an observed feature is a false alarm. The number of false alarms contained in an association event θ is given by $(m_k - |\theta|)$. Then $\gamma^{(m_k-|\theta|)}$ is the probability assigned to all false alarms in $\mathbf{Z}(k)$ given θ. All other features are uniquely assigned to an object. Making the assumption that the features are detected independently of each other, we get

$$p(\mathbf{Z}(k) \mid \theta, \mathbf{X}^k) = \gamma^{(m_k-|\theta|)} \prod_{(j,i)\in\theta} p(\mathbf{z}_j(k) \mid \mathbf{x}_i^k). \tag{4}$$

Please note that in the case of Kalman filter based tracking, the terms $p(\mathbf{z}_j(k) \mid \mathbf{x}_i^k)$ are given by normal distributions. In the general case, Eq. (3) becomes

$$P(\theta \mid \mathbf{Z}^k) = \alpha\ \gamma^{(m_k-|\theta|)} \prod_{(j,i)\in\theta} p(\mathbf{z}_j(k) \mid \mathbf{x}_i^k). \tag{5}$$

To update the beliefs $p(\mathbf{x}_i^k)$ about the states of the individual objects, JPDAFs apply recursive Bayesian filtering. The new state of an object is *predicted* using the term

$$p(\mathbf{x}_i^k \mid \mathbf{Z}^{k-1}) = \int p(\mathbf{x}_i^k \mid \mathbf{x}_i^{k-1}, t)\ p(\mathbf{x}_i^{k-1} \mid \mathbf{Z}^{k-1})\ d\mathbf{x}_i^{k-1}. \tag{6}$$

Whenever new sensory input arrives, the state is corrected according to

$$p(\mathbf{x}_i^k \mid \mathbf{Z}^k) = \alpha\ p(\mathbf{Z}(k) \mid \mathbf{x}_i^k)\ p(\mathbf{x}_i^k \mid \mathbf{Z}^{k-1}). \tag{7}$$

Since we don't know which of the features in $\mathbf{Z}(k)$ is caused by object i, we integrate the single features according to the assignment probabilities β_{ji}:

$$p(\mathbf{x}_i^k \mid \mathbf{Z}^k) = \alpha \sum_{j=0}^{m_k} \beta_{ji} p(\mathbf{z}_j(k) \mid \mathbf{x}_i^k)\ p(\mathbf{x}_i^k). \tag{8}$$

Again, α is a normalization factor. Thus, all we need to know are the models $p(\mathbf{x}_i^k \mid \mathbf{x}_i^{k-1}, t)$ and $p(\mathbf{z}_j(k) \mid \mathbf{x}_i^k)$. Both depend on the properties of the objects being tracked and the sensors used. One additional important aspect is how distributions over the state space of the individual objects are represented.

2.2 Implementing JPDAFs with Samples

In most applications to target tracking, Kalman filters and hence Gaussian distributions are used to track the individ-

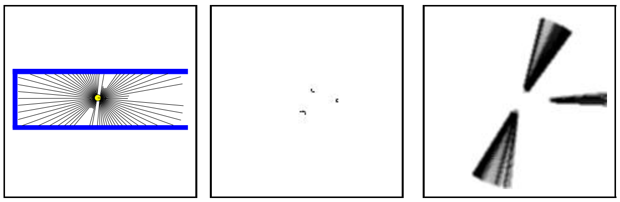

Figure 1: Typical laser-range finder scan. Two of the local minima are caused by people walking by the robot (left image). Feature grid extracted from the scan representing the corresponding probability densities $P(\text{legs}^k_{x,y})$ (center). Occlusion grid representing $P(\text{occluded}^k_{x,y})$ (right).

ual objects. In our approach, we use sample-based representations of the individual beliefs of the states of the objects which allows us to represent arbitrary densities.

The key idea underlying all particle filters is to represent the density $p(\mathbf{x}^k_i \mid \mathbf{Z}^k)$ by a set $\mathbf{S}^k_i$ of N weighted, random samples or *particles* $s^k_{i,n}(n = 1 \ldots N)$. A sample set constitutes a discrete approximation of a probability distribution. Each sample is a tuple $(x^k_{i,n}, w^k_{i,n})$ consisting of state $x^k_{i,n}$ and an importance factor $w^k_{i,n}$. The *prediction* step of Bayesian filtering is realized by drawing samples from the set computed in the previous iteration and by updating their state according to the prediction model $p(\mathbf{x}^k_i \mid \mathbf{x}^{k-1}_i, t)$. In the *correction* step, a measurement $\mathbf{Z}(k)$ is integrated into the samples obtained in the prediction step. In the context of the JPDAF the assignment probabilities β_{ji} have to be considered in the correction step. To compute the β_{ji} we use the following particle filter version of Eq. (5):

$$P(\theta \mid \mathbf{Z}^k) = \alpha \gamma^{(m_k - |\theta|)} \prod_{(j,i)\in\theta} \frac{1}{N} \sum_{n=1}^{N} p(\mathbf{z}_j(k) \mid x^k_{i,n}). \qquad (9)$$

Given the assignment probabilities we now can compute the weights of the samples

$$w^k_{i,n} = \alpha \sum_{j=0}^{m_k} \beta_{ji} p(\mathbf{z}_j(k) \mid x^k_{i,n}), \qquad (10)$$

where α is a normalization constant ensuring that the weights sum up to one over all samples. Finally, we obtain N new samples from the current samples by bootstrap resampling. For this purpose we select every sample $x^k_{i,n}$ with probability $w^k_{i,n}$.

3 Application to Laser-based People Tracking with a Mobile Robot

In this section we describe the application of the sample-based JPDAF to the task of tracking people using the range data obtained with a mobile robot. Our mobile platform is equipped with two laser-range scanners mounted at a height of 40 cm. Each scan of these two sensors covers the whole surrounding of the robot at an angular resolution of 1 degree. To represent the state of a person we use a quadruple $\langle x, y, \phi, v\rangle$, where x and y represent the position relative to the robot, ϕ is the orientation and v is the walking speed of the person.

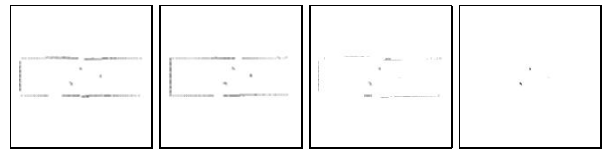

Figure 2: From left to right: the current occupancy map $P(\text{occ}_{x,y} \mid \mathbf{Z}(k))$, the previous occupancy map $P(\text{occ}_{x,y} \mid \mathbf{Z}(k-1))$, the resulting difference map $P(\text{new}^k_{x,y})$, and the fusion of the difference map with the feature maps for the scan depicted in Figure 1

To robustly identify and keep track of persons, our system uses two different patterns in range-scans which are typically caused by humans walking through a building. Consider as an example the situation shown in the left image of Figure 1. In this situation two people pass a robot in a corridor which cause two local minima in the distance profile of the range scan. The third minimum is caused by a trash bin placed in the corridor. Given these local minima we compute a set of two-dimensional position probability grids containing in each cell the probability $P(\text{legs}^k_{x,y})$ that a persons legs are at position $\langle x, y\rangle$ relative to the robot. We generate one such map for each feature in the range-scan. An overlay of all three maps representing the candidates found in our example is shown in the center of Figure 1.

Unfortunately, there are other objects in typical office environments which produce patterns similar to people. To deal with this problem our system additionally considers the changes in consecutive scans in order to distinguish between static and moving objects. This is achieved by computing local occupancy grid maps [13] for each pair of consecutive scans. Based on these occupancy grids we compute the probability $P(\text{new}^k_{x,y})$ that something moved to location $\langle x, y\rangle$:

$$P(\text{new}^k_{x,y}) = P(\text{occ}_{x,y} \mid \mathbf{Z}(k)) \cdot (1 - P(\text{occ}_{x,y} \mid \mathbf{Z}(k-1))).$$

Because the local maps $P(\text{occ}_{x,y} \mid \mathbf{Z}(k))$ are built while the robot is moving, we first align the maps using the scan-matching technique presented in [10]. Figure 2 shows two subsequent and aligned local grids and the resulting grid representing $P(\text{new}^k_{x,y})$.

To compute the importance factors of the samples in the correction steps of the particle filters, we combine the difference grid and the position probability grid of each feature $\mathbf{z}_j(k)$, $1 \le j \le m_k$ to compute $p(\mathbf{z}_j(k) \mid x_{i,n}(k))$. Under the assumption that the probabilities are independent, we have

$$P(\mathbf{z}_j(k) \mid x^k_{i,n}) = P(\text{legs}^k_{x_{i,n}}) \cdot P(\text{new}^k_{x_{i,n}}). \qquad (11)$$

Additionally we have to compute the likelihood $p(\mathbf{z}_0(k) \mid$

$x_{i,n}(k)$), that the person indicated by a sample $x_{i,n}(k)$ has not been detected in the current scan. This can be due to failure in the feature extraction or due to occlusion. The first case is modeled by a small constant value in all cells of the position and difference grids. To deal with possible occlusions we compute the so-called "occlusion map" containing for each position in the surrounding of the robot the probability $P(\text{occluded}^k_{x,y})$ that the corresponding position is not visible given the current features extracted in the first step. The resulting occlusion map for the scan shown on the left of Figure 1 is depicted in the right image of Figure 1. To determine $P(\mathbf{z}_0(k) \mid x_{i,n}(k))$ we use this occlusion map and a fixed feature detection failure probability $P(\neg\text{Detect})$:

$$P(\mathbf{z}_0(k) \mid x_{i,n}(k)) = P(\text{occluded}^k_{x_{i,n}} \vee \neg\text{Detect}). \quad (12)$$

To predict the samples estimating the motions of persons, we apply a probabilistic motion model. We assume that a person changes its walking direction and walking speed according to Gaussian distributions. Thereby, the translational velocity is assumed to lie between 0 and 150 cm/s. Additionally, we use static objects, which also are extracted out of the laser-range data, to avoid that samples end up inside of objects. As illustrated by the experiments, this leads to a physically more consistent model.

4 Experimental Results

To evaluate the performance of our approach, we performed extensive experiments with the mobile robot Rhino in our office environment. Additionally we carried out a series of simulation experiments. All experiments demonstrate that our approach can accurately and reliably keep track of moving objects even in situations in which occlusions occur and in which the robot is moving at speeds of up to 40 cm/s.[1]

4.1 Implementation Details

To initialize the JPDAF estimate, the number of persons is determined by analyzing for each new feature whether it can be explained using the belief of the objects currently being tracked. If too few samples fall into the area of a feature, we initialize a new sample set. This set is sampled from a normal distribution with mean at the center of the feature. Track termination is achieved by counting for each sample set the number of times it corresponds to at least one feature. If we cannot find any features for a certain period of time (which is 3.5 seconds in our current system), the sample set is removed.

[1] See http://www.cs.uni-bonn.de/˜schulz/Tracker.html for animations showing for example the evolution of sample sets over time.

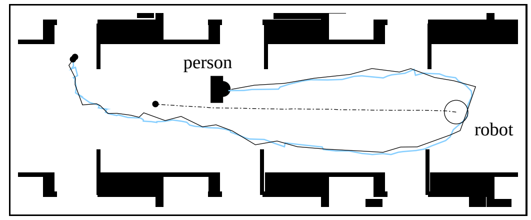

Figure 3: Rhino tracking a single person. Whereas the trajectory of the person is shown in light grey, the estimated path is depicted in black. The dotted line indicates the path of the robot.

Figure 4: Single person tracked by the robot.

During all of these experiments the resolution of the grids was set to 10 cm and the update frequency was two laser-range scans per second.

4.2 Accuracy of JPDAF-Tracking

In the first experiment the robot Rhino was moving with speeds of up to 40 cm/s in the corridor of the department building. Simultaneously, two persons were walking in the corridor, frequently entering the robot's perceptual range of 7m. The longest continuous track of a person was approximately 102 m and took 147 seconds. A typical example is given in Figure 3. Here the robot moved in the center of the corridor and a person walked around the robot and then entered a room. Whereas the trajectory of the robot is indicated by a dashed line, the trajectory of the person is shown by the black line. The trajectory of the person estimated by our system is indicated by the grey line. Figure 4 shows on the left side a camera image of the robot and the person passing it. The right image of this figure contains a 3D-visualization of the situation based on the system's estimate.

Additionally, we used the data recorded during this experiment to evaluate the accuracy of our approach. We manually determined the position of the person for each laser scan contained in the data file and determined the average displacement of the positions estimated by our JPDAF approach. It turned out that the average estimation error is 15 cm in this experiment.

In the next experiment, the task of the system was to simultaneously track multiple moving objects which temporarily occluded each other. Figure 5 shows a typical simulation experiment in which the robot moved at a speed of 25 cm/s

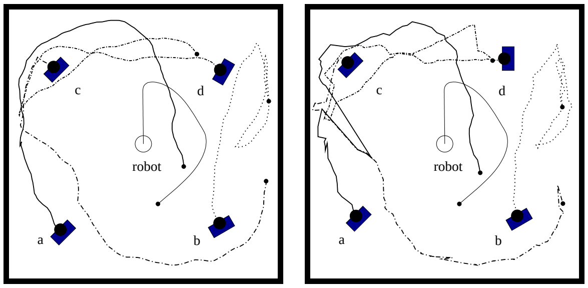

Figure 5: Tracking four objects with a moving robot. The real trajectories are shown on the left side. The right image contains the estimated paths.

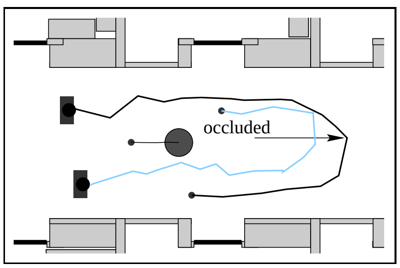

Figure 6: Two estimated trajectories of two persons in a situation in which one person temporarily occludes the other. The arrow indicates the point in time, when the occlusion occurred.

and simultaneously tracked 4 moving objects. The objects performed a random walk with a constant velocity of 50 cm/s. The change of the movement direction follows a Gaussian distribution with a standard deviation of 45 degree. In this experiment, the average estimation error was 29 cm.

Figure 6 shows a particularly challenging situation in which Rhino is able to successfully track two persons walking with velocities of 60cm/s and 80cm/s and temporarily occluding each other. To analyze the advantage of the explicit occlusion handling, we analyzed a data set acquired with Rhino and extracted four situations with occlusions. We then created ten different fractions out of the corresponding part of the data set and obtained 40 different sequences of data. For all these 40 sequences we evaluated the performance of our tracking algorithm with and without occlusion handling. Thereby we regarded it as a tracking failure, if one of the two sample sets is removed or if one sample set tracked the wrong person after the occlusion took place. Without occlusion handling, the system could keep track of the persons in only 33 cases. Using the occlusion maps the system tracked both persons reliably in 39 cases. Thus the explicit modeling of occlusion increases the performance of the system.

These experiments demonstrate, that our system is able to reliably track several persons even in the case of occlusion and that it provides accurate estimates.

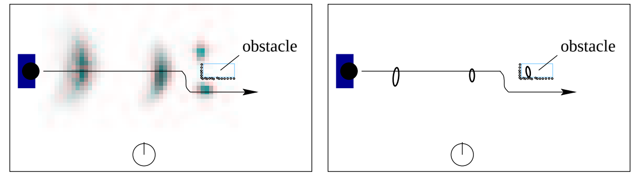

Figure 7: Tracking one object approaching a static obstacle; using a particle filter (left) and using a Kalman filter (right). The arrow indicates the trajectory taken by the object. The Kalman filter erroneously predicts that the object moves into the obstacle.

4.3 Comparison to Kalman Filter-based Approaches

As pointed out above, the main advantage of particle filters compared to Kalman filters is that particle filters in principle can represent arbitrary densities, whereas Kalman filters are restricted to Gaussian distributions.

Figure 7 shows a typical example in which this restriction leads to a wrong prediction. Here the robot tracks a person which walks straight towards an obstacle and then passes it on the right side (see solid line). The person is tracked using both, a Kalman filter and a particle filter. The left image of Figure 7 also contains the probability densities of the particle filters at different points in time. The grey-shaded contours indicate the corresponding distributions of the particles (the darker the higher the likelihood), which are obtained by computing a histogram over a discrete grid of poses. The Gaussians computed with the Kalman filter are shown in the right image of Figure 7. Here the ellipses indicated the Mahalanobis distance.

As can be seen from the figure, both filters correctly predict the position of the person in the first two steps. However, in the third step the Kalman filter predicts that the person will move into the obstacle. whereas the sample-based approach can exploit the fact that persons do not move inside of obstacles. Thus, the samples correctly predict that the person will pass the obstacle either to the left or to the right. This is indicated by the bimodal distribution shown in the left image Figure 7. Thus particle filter-based approaches produce more accurate densities than methods using Kalman filters.

4.4 Advantage of the JPDAF over Standard Particle Filters

In the past, single state particle filters have also been used for tracking multiple objects [12, 8]. This approach, which rests on the assumption that each mode in the density corresponds to an object, is only feasible if all objects can be sensed at every point in time and if the measurement errors are small. If, for example, one object is occluded, the samples tracking this object obtain significantly smaller impor-

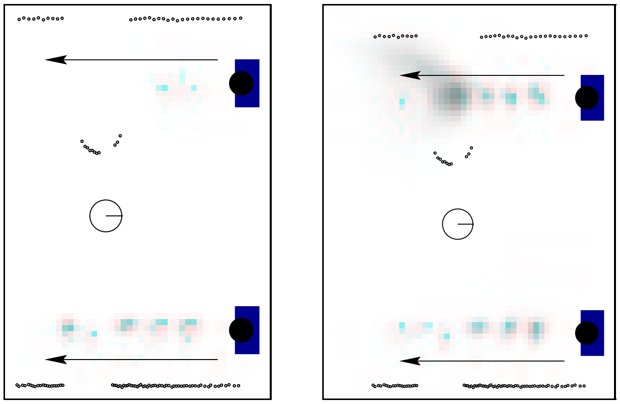

Figure 8: Tracking two persons with one sample set (left) and with two sample sets (right). The arrows indicate the movement direction

tance factors than the samples tracking the other objects. As a result, the occluded object gets quickly lost, since the samples tend to focus on the other objects.

Figure 8 shows an example situation in which the robot is tracking two persons. Whereas one person is visible all the time, the second one is temporarily occluded by a static obstacle. The left part of the figure shows the evolution of the probability distributions using such a single-state particle filter. This filter was initialized with a bimodal distribution using 5000 particles for each object. As soon as the upper object is occluded, the density becomes unimodal and the filter loses track of the occluded object. The right part of the figure shows the resulting densities using our JPDAF approach. Although the uncertainty about the position of the occluded object increases, the filter is able to reliably keep track of both objects.

5 Summary and Conclusions

In this paper we presented a technique for keeping track of multiple moving objects with a mobile robot. In order to avoid the exponential complexity of joint state spaces, each object is tracked using a particle filter and Joint Probabilistic Data Association Filters are applied to solve the problem of assigning measurements to the individual objects. By integrating particle filters with JPDAFs, our approach inherits the advantages of both: it can represent arbitrary densities over the state space of the individual objects while still being able to efficiently solve the data association problem. Our approach uses special techniques to extract the necessary features from proximity data and to deal with occlusions.

The technique has been implemented and evaluated on a real robot as well as in simulation runs. The experiments carried out in a typical office environment demonstrate that our approach is able to reliably keep track of multiple persons. They furthermore illustrate that our approach outperforms other techniques developed so far.

References

[1] Y. Bar-Shalom and T.E. Fortmann. *Tracking and Data Association*. Mathematics in Science and Engineering. Academic Press, 1988.

[2] M.J. Black and A.D. Jepson. A probabilistic framework for matching temporal trajectories: Condensation-based recognition of gestures and expressions. In *ECCV*, 1998.

[3] I.J. Cox. A review of statistical data association techniques for motion correspondence. *International Journal of Computer Vision*, 10(1):53–66, 1993.

[4] I.J. Cox and S.L. Hingorani. An efficient implementation of reids multiple hypothesis tracking algorithm and its evaluation for the purpose of visual tracking. In *IEEE Transactions on PAMI*, volume 18, pages 138–150. February 1996.

[5] D. Fox, W. Burgard, F. Dellaert, and S. Thrun. Monte Carlo localization: Efficient position estimation for mobile robots. In *Proc. of the National Conference on Artificial Intelligence (AAAI)*, 1999.

[6] N.J. Gordon, D.J. Salmond, and A.F.M. Smith. A novel approach to nonlinear/non-Gaussian Bayesian state estimation. *IEE Proceedings F*, 140(2):107–113, 1993.

[7] J.-S. Gutmann, W. Burgard, D. Fox, and K. Konolige. An experimental comparison of localization methods. In *Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 1998.

[8] M. Isard and A. Blake. Contour tracking by stochastic propagation of conditional density. In *Proc. of the European Conference of Computer Vision*, 1996.

[9] K. Kanazawa, D. Koller, and S.J. Russell. Stochastic simulation algorithms for dynamic probabilistic networks. In *Proc. of the 11th Annual Conference on Uncertainty in AI (UAI) Montreal, Canada*, 1995.

[10] F. Lu and E.E. Milios. Robot pose estimation in unknown environments by matching 2d range scans. In *IEEE Computer Vision and Pattern Recognition Conference (CVPR)*, 1994.

[11] J. MacCormick and A. Blake. A probabilistic exclusion principle for tracking multiple objects. In *Proc. of 7th International Conference on Computer Vision (ICCV)*, pages 572–587, 1999.

[12] E.B. Meier and F. Ade. Using the condensation algorithm to implement tracking for mobile robots. In *Proc. of the Third European Workshop on Advanced Mobile Robots, Eurobot99*, pages 73–80. IEEE, 1999.

[13] Hans P. Moravec and A.E. Elfes. High resolution maps from wide angle sonar. In *Proc. of the IEEE International Conference on Robotics & Automation (ICRA)*, pages 116–121, 1985.

[14] M.K. Pitt and N. Shephard. Filtering via simulation: auxiliary particle filters. *Journal of the American Statistical Association*, 94(446), 1999.