Tracking Multiple Moving Objects with a Mobile Robot

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. For many tasks it is therefore highly desirable that a robot can determine the positions of the humans in its surrounding. In this paper we introduce sample-based joint probabilistic data association filters to track multiple moving objects with a mobile robot. Our technique uses the robot's sensors and a motion model of the objects being tracked. A Bayesian filtering technique is applied to adapt the tracking process to the number of objects in the sensor range of the robot. Our approach to tracking multiple moving objects has been implemented and tested on a real robot. We present experiments illustrating that our approach is able to robustly keep track of multiple persons even in situations in which people are temporarily occluded. The experiments furthermore show that the approach outperforms other techniques developed so far.

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 positions and velocities. We introduce sample-based Joint Probabilistic Data Association Filters (SJPDAFs). JPDAFs [1; 3] are a very popular approach to tracking multiple moving objects. They compute a Bayesian estimate of the correspondence between features detected in the sensor data and the different objects to be tracked. Like virtually all existing approaches to tracking multiple targets, they 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 [7; 15]. They have been applied with great success to different state estimation problems including visual tracking [2; 9], mobile robot localization [5] and dynamic probabilistic networks [10]. The key idea of particle filters is to represent the state by sets of samples (or particles). The major advantage of this technique lies in the ability to represent multi-modal state densities, a property which has been shown to increase the robustness of the underlying state estimation process [8]. 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 [12]. 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. Instead of relying on Gaussian distributions extracted from the sample sets as proposed in [6], our approach applies the idea of JPDAFs directly to the sample sets of the individual particle filters. To adapt the SJPDAFs to the different numbers of objects in the robot's sensor range, our approach maintains a probability distribution over the number of objects being tracked. The robustness of the overall approach is increased by applying a motion model of the objects being tracked. Furthermore, it uses different features extracted from consecutive sensor measurements to explicitely deal 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 introduce SJPDAFs in Section 2. Section 3 explains how we add and remove particle filters based on an estimate of the current number of objects. In Section 4, we describe how to extract features from proximity information provided by a robot's laser range-finders. Furthermore, we show how to deal with occlusions. Section 5 describes several experiments carried out on a real robot and in simulations. The experiments illustrate the capabilities and the robustness of our approach.

2 Sample-based Joint Probabilistic Data Association Filters (SJPDAFs)

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. 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 when tracking multiple objects is how to assign the observed features to the individual 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 Θ_{ji} 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

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

Using Bayes' rule and under the assumption that the estimation problem is Markovian, we can compute the probability $P(\theta \mid \mathbf{Z}^k)$ of an individual joint association event as

$$\begin{aligned} P(\theta \mid \mathbf{Z}^k) \quad &= \quad P(\theta \mid \mathbf{Z}(k), \mathbf{Z}^{k-1}) && (2)\\ &\overset{\text{Markov!}}{=} \quad P(\theta \mid \mathbf{Z}(k), \mathbf{X}^k) && (3)\\ &\overset{\text{Bayes!}}{=} \quad \alpha\ p(\mathbf{Z}(k) \mid \theta, \mathbf{X}^k)\ P(\theta \mid \mathbf{X}^k). && (4) \end{aligned}$$

Here α is a normalizer ensuring that $P(\theta \mid \mathbf{Z}^k)$ sums up to one over all θ. 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 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

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

By inserting Eq. (5) into Eq. (4), using the assumption that $P(\theta \mid \mathbf{X}^k)$ is constant, and after inserting the result into Eq. (1) we obtain

$$\beta_{ji} \quad = \quad \sum_{\theta \in \Theta_{ji}} \alpha\ \gamma^{(m_k - |\theta|)} \prod_{(j,i) \in \theta} p(\mathbf{z}_j(k) \mid \mathbf{x}_i^k). \tag{6}$$

It remains to describe, how the beliefs $p(\mathbf{x}_i^k)$ about the states of the individual objects are updated. In the standard JPDAF it is generally assumed that the underlying densities are Gaussians, and Kalman filtering is applied to update these densities. In the more general framework of Bayesian filtering, the update equation for the prediction of the new state of an object is

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

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

$$p(\mathbf{x}_i^k \mid \mathbf{Z}^k) = \alpha\ p(\mathbf{Z}(k) \mid \mathbf{x}_i^k)\ p(\mathbf{x}_i^k \mid \mathbf{Z}^{k-1}), \tag{8}$$

where, again, α is a normalization factor. Since we do not 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}

$$p(\mathbf{x}_i^k \mid \mathbf{Z}^k) \quad = \quad \alpha \sum_{j=0}^{m_k} \beta_{ji} p(\mathbf{z}_j(k) \mid \mathbf{x}_i^k)\ p(\mathbf{x}_i^k). \tag{9}$$

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 the distributions $p(\mathbf{x}_i^k)$ over the state spaces of the individual objects are represented.

2.2 The Sample-based Approach

In most applications to target tracking, Kalman filters and hence Gaussian distributions are used to track the individual 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 $p(\mathbf{x}_i^k \mid \mathbf{Z}^k)$ by a set $\mathbf{S}_i^k$ of N weighted, random samples or *particles* $s_{i,n}^k(n = 1 \dots N)$. A sample set constitutes a discrete approximation of a probability distribution. Each sample is a tuple $(x_{i,n}^k, w_{i,n}^k)$ consisting of state $x_{i,n}^k$ and an importance factor $w_{i,n}^k$. 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 $p(\mathbf{x}_i^k \mid \mathbf{x}_i^{k-1}, t)$. In the *correction* step, a measurement $\mathbf{Z}(k)$ is integrated into the samples obtained in the prediction step. According to Eq. (9) we have to consider the assignment probabilities β_{ji} in this step. To determine β_{ji}, however, we have to compute $p(\mathbf{z}_j(k) \mid \mathbf{x}_i^k)$. In our particle filter-based version this quantity is obtained by integrating over all samples:

$$p(\mathbf{z}_j(k) \mid \mathbf{x}_i^k) \quad = \quad \frac{1}{N} \sum_{n=1}^{N} p(\mathbf{z}_j(k) \mid x_{i,n}^k). \qquad (10)$$

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

$$w_{i,n}^k = \alpha \sum_{j=0}^{m_k} \beta_{ji} p(\mathbf{z}_j(k) \mid x_{i,n}^k), \qquad (11)$$

where α is a normalizer 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 $x_{i,n}^k$ with probability $w_{i,n}^k$.

3 Estimating the Number of Objects

The Joint Probabilistic Data Association Filter assumes that the number of objects to be tracked is known. In practical applications, however, the number of objects often varies over time. For example, when a mobile robot is moving in a populated environment, the number of people in the perceptual field of the robot changes frequently. We deal with this problem by additionally maintaining a density $P(N^k \mid \mathbf{M}^k)$ over the number of objects N^k at time k, where $\mathbf{M}^k = m_0, \dots, m_k$ is the sequence of the numbers of features observed so far. Using Bayes' rule, we have

$$P(N^k \mid \mathbf{M}^k) \quad = \quad \alpha \cdot P(m_k \mid N^k, \mathbf{M}^{k-1}) \cdot P(N^k \mid \mathbf{M}^{k-1}).$$

Under the assumption that, given the current number of objects, the number of observed features is independent of the number of previously observed features, we obtain

$$P(N^k \mid \mathbf{M}^k) \quad = \quad \alpha \cdot P(m_k \mid N^k) \cdot P(N^k \mid \mathbf{M}^{k-1}).$$

Using the law of total probability, and given the assumption that N^k is independent of $\mathbf{M}^{k-1}$ given N^{k-1} we have

$$P(N^k \mid \mathbf{M}^k) = \alpha \cdot P(m_k \mid N^k) \cdot \sum_n P(N^k \mid N^{k-1} = n) \cdot P(N^{k-1} = n \mid \mathbf{M}^{k-1}) \quad (12)$$

As a result, we obtain a recursive update procedure for $P(N^k \mid \mathbf{M}^k)$, where all we need to know are the quantities $P(m_k \mid N^k)$ and $P(N^k \mid N^{k-1})$. The term $P(m_k \mid N^k)$ represents the probability of observing m_k features, if N^k objects are in the perceptual field of the sensor. In our current implementation, which uses laser-range sensors, we learned this quantity from a series of 5100 range scans, recorded during a simulation experiment, where up to 10 objects where placed at random locations within the robot's surrounding. The resulting density is depicted in Fig. 1. Please note that the probability of an occlusion increases with the number of objects. Accordingly, the more objects are in the perceptual field of the robot, the more likely it becomes that a feature is missing. Therefore, the resulting distribution shown in Fig. 1 does not correspond to a straight line lying on the diagonal. The term $P(N^k \mid N^{k-1})$ specifies how the number of objects changes over time. In our system we model arrivals of new objects and departures of known objects as Poisson processes.

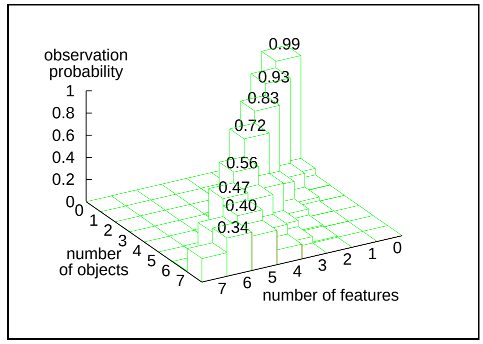

Figure 1: The sensor model $P(m_k \mid N^k)$ specifying the probability of observing m_k features, if N^k objects are in the perceptual range of the sensor

To adapt the number of objects in the SJPDAF, our system uses the maximum likelihood estimate of $P(N^k \mid \mathbf{M}^k)$. If the number of particle filters in the SJPDAF is smaller than the new estimate for N^k, new filters need to be initialized. Since we do not know which of the features originate from new objects, we initialize the new filters using a uniform distribution. We then rely on the SJPDAF to disambiguate this distribution during subsequent filter updates.

In the alternative case that the new estimate for N^k is smaller then the current number of particle filters, some of the filters need to be removed. This, however, requires that we know which filter does not track an object any longer. To estimate the tracking performance of a sample set, we accumulate a discounted average $\hat{W}_i^k$ of the sum of sample weights W_i^k before the normalization step:

$$\hat{W}_i^k = (1-\delta)\hat{W}_i^{k-1} + \delta W_i^k. \qquad (13)$$

Since the sum of sample weights decreases significantly, whenever a filter is not tracking any feature contained in the measurement, we use this value as an indicator that the corresponding object has left the perceptual field of the robot. Whenever we have to remove a filter, we choose the one with the smallest discounted average $\hat{W}_i^k$.

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

In this section we describe the application of the SJPDAF to the task of tracking people using the range data obtained

Figure 2: 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).

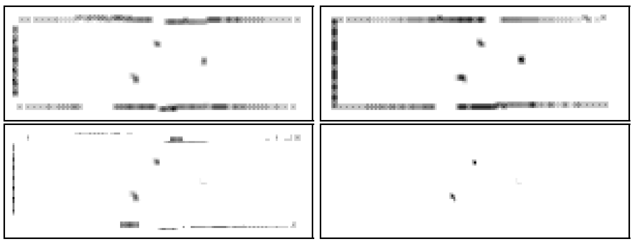

Figure 3: From left to right, top-down: the current occupancy map $P(\text{occ}_{x,y} \mid \mathbf{Z}(k))$, the previous occupancy map $P(\text{occ}_{x,y} \mid \mathbf{Z}(k-1))$, the resulting difference map $P(\text{new}^k_{x,y})$, and the fusion of the difference map with the feature maps for the scan depicted in Figure 2

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.

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 2. In this situation two people pass a robot in a corridor which cause two local minima in the distance histogram 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 2.

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 [14] 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 [11]. Figure 3 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 $\mathbf{z}_j(k)$, $1 \leq j \leq m_k$ to compute $p(\mathbf{z}_j(k) \mid x_{i,n}(k))$. Under the assumption that the probabilities are independent, we have

$$P(\mathbf{z}_j(k) \mid x^k_{i,n}) = P(\text{legs}^k_{x_{i,n}}) \cdot P(\text{new}^k_{x_{i,n}}). \tag{14}$$

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 occlusions. 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 2 is depicted in the right image of Figure 2. To determine $P(\mathbf{z}_0(k) \mid x_{i,n}(k))$ we use this occlusion map and a fixed feature detection failure probability $P(\neg\text{Detect})$:

$$P(\mathbf{z}_0(k) \mid x_{i,n}(k)) = P(\text{occluded}^k_{x_{i,n}} \vee \neg\text{Detect}). \tag{15}$$

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 the static objects also extracted from consecutive scans to avoid that samples end up inside of objects or move through objects.

5 Experimental Results

Our approach has been implemented and tested extensively with our mobile robot as well as in simulation runs. Throughout the experiments the robot was moving with speeds of up to 40 cm/s. The current implementation is able to integrate laser-range scans at a rate of 3Hz.

5.1 Accuracy of SJPDAF-Tracking

In the first experiment our robot was placed in the corridor of the department building. Simultaneously, up to four persons were walking in the corridor, frequently entering the robot's perceptual range of 8m. The complete experiment took 235 seconds. Throughout this experiment the persons entered the robot's sensor range 16 times. In all cases our SJPDAF was able to correctly keep track of the persons.

A typical example is given in Figure 4. Here three people are walking along the corridor. The upper image shows the ground truth extracted from the data recorded during the experiment. The lower image depicts the trajectories estimated by our algorithm. Please note that there is a certain delay in the initialization of the sample set for the person marked "a". This delay is due to the occlusion by person "b". To obtain quantitative results about the capabilities of the SJPDAF, we measured the distance between the manually determined

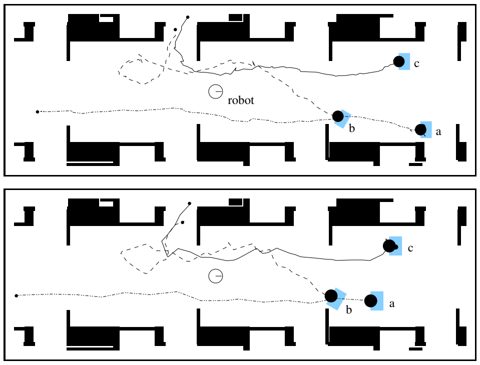

Figure 4: Typical situation in which the robot had to track three persons. Whereas the ground truth is shown in the upper image, the estimated trajectories are depicted in the lower image.

positions of the persons and the robot's estimate for each individual scan. It turned out that the average displacement between the ground truth position and the estimated position was 19 cm and the maximum displacement was 37 cm.

In the next experiment, carried out in simulation, the task of the system was to simultaneously track multiple moving objects while the robot was moving. Figure 5 shows a typical fraction of this experiment during which the robot moved at speeds of up to 25 cm/s. In this case the robot had to track 4 moving objects. The objects performed a random walk with a constant velocity of 50 cm/s. In this case, the changes of the movement directions of the objects followed a Gaussian distribution with a standard deviation of 45 degrees. Here the average estimation error was 29 cm.

To analyze the advantage of the explicit occlusion handling, we systematically analyzed a data set recorded with our robot while it was moving at a speed of 40 cm/sec. In the experiment shown in Figure 6, two persons were walking in the corridor of our department with speeds of 60cm/s and 80cm/s, respectively. As can be seen in the figure, one person temporarily occludes the other person. From this data set we created 40 different sequences which were used for the quantitative analysis. For all these 40 data sets 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 failed in seven cases

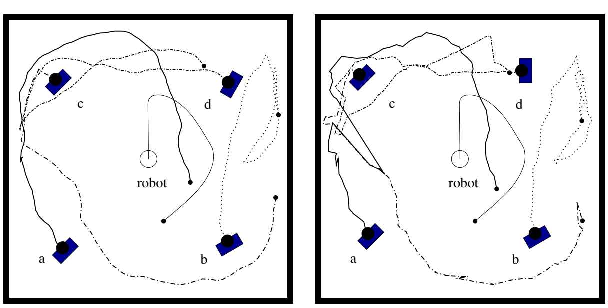

Figure 5: Tracking four objects with a moving robot. The real trajectories are shown on the left side. The right image contains the estimated paths.

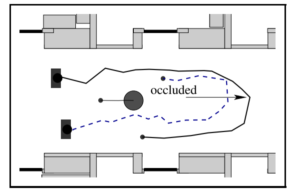

Figure 6: Estimated trajectories of two persons in a situation in which one person temporarily occludes the other. The arrow indicates the point in time, when the occlusion occurred.

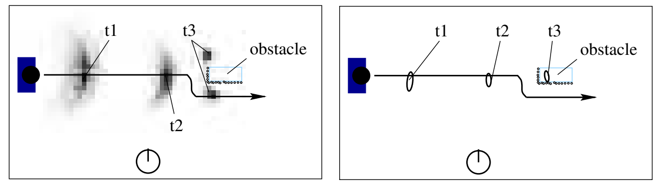

Figure 7: Tracking one object approaching a static obstacle; using a particle filter (left) and using a Kalman filter (right). The arrow indicates the trajectory taken by the object. The Kalman filter erroneously predicts that the object moves into the obstacle.

(17.5%). With occlusion handling, the robot was able to reliably keep track of both persons and failed in only one of the 40 cases (2.5%).

These experiments demonstrate, that our system is able to reliably and accurately keep track of several moving objects even in cases in which occlusions occur.

5.2 Comparison to Standard JPDAFs

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. Accordingly our SJPDAF-approach produces more accurate densities than the standard JPDAF using Kalman filters.

Figure 7 shows a typical example in which the restricted representation of Kalman filters 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 left image of Figure 7 also contains the probability densities of the particle filters computed by the SJPDAF at three different points in time. The grey-shaded contours indicate the corresponding distributions of the particles (the darker the higher the likelihood), which were obtained by computing a histogram over a discrete grid of poses. The Mahalanobis distance of the Gaussians computed at the same points in time by the standard JPDAF using a Kalman filter are shown in the right image of Figure 7.

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. Our SJPDAF, in contrast, correctly predicts 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 of Figure 7.

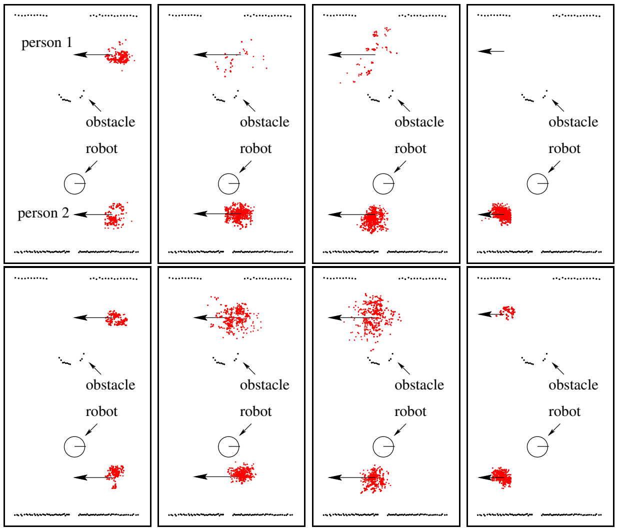

Figure 8: Tracking two persons with one sample set (top row) and with two sample sets (bottom row). The arrows indicate the position of the persons and their movement direction.

5.3 Advantage of the SJPDAF over Standard Particle Filters

In the past, single state particle filters have also been used for tracking multiple objects [13; 9]. 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 importance factors than the samples tracking the other objects. As a result, all samples quickly focus on the unoccluded object.

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 upper row of the figure shows the evolution of the samples using such a single-state particle filter. This filter was initialized with a bimodal distribution using 1000 particles for each object. As soon as the upper object is occluded, all samples are moved to the unoccluded person and the filter loses track of the occluded object. The lower row of the figure shows the samples sets resulting from our SJPDAF. Although the uncertainty about the position of the occluded object increases, the filter is able to reliably keep track of both objects.

6 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 SJPDAF 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 a probabilistic approach to deal with varying numbers of objects. Furthermore, it extracts appropriate features from proximity data to deal with possible 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. A hybrid bootstrap filter for target tracking in clutter. In *IEEE Transactions on Aerospace and Electronic Systems*, volume 33, pages 353–358. January 1997.

[7] 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.

[8] 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.

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

[10] 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.

[11] 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.

[12] 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.

[13] 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.

[14] 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.

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