People Tracking with a Mobile Robot Using Sample-based Joint Probabilistic Data Association Filters

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

10th February 2003

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 track the positions of the humans in its surrounding. In this paper we introduce sample-based joint probabilistic data association filters as a new algorithm to track multiple moving objects. Our method applies Bayesian filtering to adapt the tracking process to the number of objects in the perceptual range of the robot. The approach has been implemented and tested on a real robot using laser-range data. We present experiments illustrating that our algorithm is able to robustly keep track of multiple persons. The experiments furthermore show that the approach outperforms other techniques developed so far.

1 Introduction

The problem of estimating the positions of moving objects has become an important problem in mobile robotics. Over the last years, several mobile robots

have been deployed in populated environments like office buildings [2, 3, 28, 53], supermarkets [16], hospitals [17], and museums [9, 55]. Knowledge about the position and the velocities of moving people can be utilized in various ways to greatly improve the behavior of such systems. For example, this information allows a robot to adapt its velocity to the speed of people in the environment. Being able to distinguish between static and moving objects can also help to improve the performance of map-building algorithms [27]. It enables a robot to improve its collision avoidance behavior in situations in which the trajectory of the robot crosses the path of a human [54]. And of course, being able to keep track of people is an important prerequisite for other man-machine interaction capabilities.

Unfortunately, the problem of estimating the position of multiple moving objects with a mobile robot is significantly harder than estimating the state of a single object. First, one also as to solve the problem of estimating the number of objects that are currently in the field of view. Furthermore, the update process is harder since observations may result in ambiguities, features may not be distinguishable, objects may be occluded, or there may be more features than objects. Thus, a system that is able to track multiple moving objects or persons must be able to estimate the number of objects and must be able to solve the problem of assigning the observed features to the objects being tracked.

In this paper we present a probabilistic 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. In particular, we introduce sample-based Joint Probabilistic Data Association Filters (SJPDAFs) which combine the advantages of sample-based density approximations with the efficiency of Joint Probabilistic Data Association Filters (JPDAFs).

JPDAFs [4, 11] are a 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 only optimal for unimodal Gaussian distributions over the state to be estimated. More recently, particle filters have been introduced to estimate non-Gaussian, non-linear dynamic processes [24, 51]. They have been applied with great success to different state estimation problems including visual tracking [7, 30], mobile robot local-

ization [14, 20, 21, 25, 39, 44, 45, 47, 56, 58] and dynamic probabilistic networks [33]. 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 [26]. 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. Unfortunately, the complexity of this approach grows exponentially in the number of objects to be tracked and is therefore limited to a small number of objects only.

In contrast to that, SJPDAFs combine the advantages of particle filters with the efficiency of existing approaches to multi-target tracking: SJPDAFs use particle filters to track the states of the objects and apply JPDAFs to assign the measurements to the individual objects. Instead of relying on Gaussian distributions extracted from the sample sets as proposed by Gordon [23], 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 discussing related work in the next section, we introduce the general concept of SJPDAFs in Section 3. Section 4 describes our approach to estimate the number of objects and to adopt the number of particle filters accordingly. In Section 5, we present our application to features extracted from proximity information provided by a robot's laser rangefinders. Section 6 describes several experiments carried out on a real robot and in simulations. The experiments illustrate the capabilities and the robustness of our approach.

2 Related Work

Several approaches for tracking human motions have been developed over the last years (see [1] for an overview). The most widely used approach is visual tracking. The existing techniques mainly differ in the features or models they use to extract people from images. For example, Darell et al. [13] describe an approach that relies on a fixed background to locate the person. The background image is subtracted to obtain a contour of the person. Other approaches [59] rely on color for tracking. Rasmussen and Hager [52] additionally track different parts of a person independently using an extension of the JPDAF, which takes body constraints into account.

Cox and Hingorani [12] use the Multiple Hypothesis Tracking algorithm (MHT) to track corner edge features with a moving camera. In contrast to the JPDAF, which maintains a single filter for each object over time, the MHT expands an exponentially growing tree of Kalman filters according to the possible correspondences of objects to features. Efficient implementations make use of polynomial time algorithm in order to prune this tree to the most likely branches [10]. However, the copying of the filters involved prohibits an efficient sample-based implementation.

A combination of a particle filters with joint probabilistic techniques has also been proposed by MacCormick and Blake [43]. Similar to our approach, they employ joint probabilistic data association to deal with occlusion and clutter. They use a single particle filter to estimate the joint state space of a constant number of objects. To partly mitigate the problem of the exponential growth of the state space, they introduce partitioned sampling and obtain an robust estimate of the combined state space of two objects using a moderate sample set size. In contrast to this, our method tracks a varying number of objects independently.

Additionally, there is a variety of vision-based techniques for tracking persons with mobile robots [6, 32, 36, 57]. Existing approaches use color tracking, a mixture of features, or stereo vision to identify single persons and eventually their gestures. However, they do not apply any filtering techniques to keep track of the persons.

Furthermore, there are tracking algorithms for mobile robots that use range sensors instead of vision. For example, Kluge et al. [34] describe an approach to estimate moving obstacles with an autonomous wheelchair. However, they do not apply a motion-model to the objects so that they cannot reliably keep track of in-

dividual objects over time and they do not deal with occlusions or other detection failures. A recent work by Montemerlo et al. [45] addresses the problem of simultaneous localization and people tracking using range sensors. The authors use conditional particle filters to incorporate the robot's uncertainty about its position. In their method the authors apply a nearest neighbor approach to solve the data association problem. Our approach presented here provides a more robust method to solve the data association problem using sample-based JPDAFs instead of a nearest neighbor solution. Fod et al. [18] present an approach to track multiple moving people within a workspace using statically mounted laser range-finders. They use Kalman filters to keep track of objects during temporary occlusions. Finally, Lindström and Eklundh [41] describe an approach for tracking moving objects from a mobile platform that uses Gaussian hypotheses. Both approaches perform the data association on a nearest neighbor basis.

There also has been several work on determining the best actions of mobile platforms in order to keep track of moving targets. Lavalle et al. [38] describe a probabilistic algorithm that maximizes the probability of future observability of a moving target from a mobile robot. González-Baños et al. [22] especially focus on the problem of tracking a moving target in the presence of obstacles. Additionally, Parker [49], Jung and Sukhatme [31], Pirjanian and Matarić [50], and Murrieta-Cid et al. [48] present distributed approaches to control a team of mobile robots in order to maximize the visibility of moving targets. Whereas the focus of these approaches lies in the generation of appropriate actions of mobile robots to keep track of moving targets, the work presented in this paper is designed to provide a technique maintaining a belief about the positions of the targets being tracked.

Recently, several authors presented techniques for learning motion patterns or for predicting motions of persons [5, 8, 29, 60]. Furthermore, several authors have investigated the problem of improving collision avoidance given the information about moving objects. For example, Tadokoro et al. [54] use a given probabilistic model of typical motion behaviors in order to predict future poses of the persons and to adapt the actions of the robot accordingly. Kruse et al. [37] employ people tracking using a ceiling mounted camera. Their approach learns where people walk within a given environment. This information is used to adapt the robot's navigation plans according to the typical behavior of the people operating in the environment. The main contributions of all these techniques rather lies in a solution to the problem of how to process the information obtained from a tracking system and not on how to actually track the persons.

3 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 Joint Probabilistic Data Association Filters (JPDAFs) [11] for this purpose. In what follows we will first describe a general version of JPDAFs and then our sample-based variant.

3.1 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

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

Under the assumption that the estimation problem is Markovian and using the law of total probability we can compute the probability $P(\theta \mid \mathbf{Z}^k)$ of an individual

joint association event as

$$
\begin{aligned}
P(\theta \mid \mathbf{Z}^k) &= P(\theta \mid \mathbf{Z}(k), \mathbf{Z}^{k-1}) && (2)\\
&= \int P(\theta \mid \mathbf{Z}(k), \mathbf{Z}^{k-1}, \mathbf{X}^k) p(\mathbf{X}^k \mid \mathbf{Z}(k), \mathbf{Z}^{k-1})\, d\mathbf{X}^k && (3)\\
&= \int P(\theta \mid \mathbf{Z}(k), \mathbf{X}^k) p(\mathbf{X}^k \mid \mathbf{Z}(k), \mathbf{Z}^{k-1})\, d\mathbf{X}^k. && (4)
\end{aligned}
$$

According to Eq. (4) we need to know the state of the objects in order to determine the assignments θ. On the other hand, we need to know θ in order to determine the position of the objects. A common approach to overcome this "chicken and egg problem" is to apply an incremental approach. The key idea is to approximate $p(\mathbf{X}^k \mid \mathbf{Z}(k), \mathbf{Z}^{k-1})$ by the belief $p(\mathbf{X}^k \mid \mathbf{Z}^{k-1})$ about the predicted state of the objects, i.e. the prediction computed using all measurements perceived before time-step k. Accordingly, we obtain

$$
\begin{aligned}
P(\theta \mid \mathbf{Z}^k) &\approx \int P(\theta \mid \mathbf{Z}(k), \mathbf{X}^k) p(\mathbf{X}^k \mid \mathbf{Z}^{k-1})\, d\mathbf{X}^k. && (5)\\
&= \alpha \int P(\mathbf{Z}(k) \mid \theta, \mathbf{X}^k)\, P(\theta \mid \mathbf{X}^k)\, p(\mathbf{X}^k \mid \mathbf{Z}^{k-1})\, d\mathbf{X}^k. && (6)
\end{aligned}
$$

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 [12] 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

$$
P(\mathbf{Z}(k) \mid \theta, \mathbf{X}^k) = \gamma^{(m_k - |\theta|)} \prod_{(j,i) \in \theta} \int p(\mathbf{z}_j(k) \mid \mathbf{x}_i^k)\, p(\mathbf{x}_i^k \mid \mathbf{Z}^{k-1})\, d\mathbf{x}_i^k. \quad (7)
$$

By inserting Eq. (7) into Eq. (6), using the assumption that $P(\theta \mid \mathbf{X}^k)$ is constant, and after inserting the result into Eq. (1) we obtain

$$\beta_{ji} \;=\; \sum_{\theta \in \Theta_{ji}} \left[\alpha \; \gamma^{(m_k - |\theta|)} \prod_{(j,i) \in \theta} \int p(\mathbf{z}_j(k) \mid \mathbf{x}_i^k) \; p(\mathbf{x}_i^k \mid \mathbf{Z}^{k-1}) \; d\mathbf{x}_i^k \right]. \quad (8)$$

Note, that in principal we have to consider an exponential number of possible associations in this summation, which would make the approach intractable for a larger number of objects. To overcome this problem, we do not consider joint association events which contain assignments of negligible likelihood $p(\mathbf{z}_j(k) \mid \mathbf{x}_i^k))$. This technique is known as gating and also applied by the standard JPDAF [4].

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 can be accurately described by their first and second moment, and Kalman filtering is applied to update these densities. In the framework of Bayesian filtering, the update equation for the prediction of the new state of an object is

$$p(\mathbf{x}_i^k \mid \mathbf{Z}^{k-1}) = \int p(\mathbf{x}_i^k \mid \mathbf{x}_i^{k-1}, t) \; p(\mathbf{x}_i^{k-1} \mid \mathbf{Z}^{k-1}) \; d\mathbf{x}_i^{k-1}, \quad (9)$$

where t denotes the time expired since the previous update. Whenever new sensory input arrives, the state is corrected according to

$$p(\mathbf{x}_i^k \mid \mathbf{Z}^k) = \alpha \; p(\mathbf{Z}(k) \mid \mathbf{x}_i^k) \; p(\mathbf{x}_i^k \mid \mathbf{Z}^{k-1}), \quad (10)$$

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}

$$p(\mathbf{x}_i^k \mid \mathbf{Z}^k) \;=\; \alpha \; \sum_{j=0}^{m_k} \beta_{ji} p(\mathbf{z}_j(k) \mid \mathbf{x}_i^k) \; p(\mathbf{x}_i^k \mid \mathbf{Z}^{k-1}). \quad (11)$$

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.

An update cycle of JPDAFs can be summarized as follows. First, the individual states $\mathbf{x}_i^k$ are predicted based on the previous estimates and the motion model

of the objects. This prediction is computed using Eq. (9), which yields distributions $p(\mathbf{x}_i^k \mid \mathbf{Z}^{k-1})$. Then, Eq. (8) is used to determine the association probabilities β_{ji}. Finally, these association probabilities are used to integrate the observations into the individual state estimates (Eq. (11)).

3.2 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 particle filters to estimate the states of the individual objects. An excellent overview of particle filters can be found in [15]. 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* $s_{i,n}^k (n = 1 \ldots N)$. 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. (11) we have to consider the assignment probabilities β_{ji} in this step. To determine β_{ji}, however, we have to integrate over the state $\mathbf{x}_i^k$, weighted by the probability $p(\mathbf{x}_i^k \mid \mathbf{Z}^{k-1})$. With the sample-based representation, this integration can be done by summing over all samples generated after the prediction step:

$$\beta_{ji} \quad = \quad \sum_{\theta \in \Theta_{ji}} \left[\alpha \, \gamma^{(m_k - |\theta|)} \prod_{(j,i) \in \theta} \frac{1}{N} \sum_{n=1}^{N} p(\mathbf{z}_j(k) \mid x_{i,n}^k) \right]. \tag{12}$$

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

$$w_{i,n}^k = \alpha \, \sum_{j=0}^{m_k} \beta_{ji} p(\mathbf{z}_j(k) \mid x_{i,n}^k), \tag{13}$$

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 $w_{i,n}^k$.

4 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 distribution $P(N^k \mid \mathbf{M}^k)$ over the number of objects N^k at time k, where $\mathbf{M}^k = m_0, \ldots, m_k$ is the sequence of the numbers of features observed so far. Using Bayes' rule, we have

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

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

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

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

$$\begin{aligned} P(N^k \mid \mathbf{M}^k) \;=\; & \alpha \cdot P(m_k \mid N^k) \\ & \cdot \sum_n \Big[P(N^k \mid N^{k-1} = n) \cdot P(N^{k-1} = n \mid \mathbf{M}^{k-1}) \Big]. \end{aligned} \quad (14)$$

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. In this figure, the values on the diagonal represent the probability that the system observes the same number of features as there are objects. Values to the left of the diagonal correspond to the probability of false alarms. Furthermore, values to the right of the diagonal are the probability of observing a specific number of missing features given the number of objects N^k. Please note that the probability of an occlusion and therefore of a missing feature increases with the number of

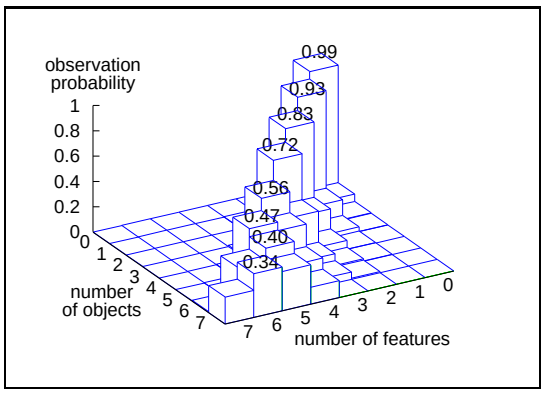

Figure 1: The sensor model $P(m_k \mid N^k)$ specifying the probability of observing m_k features, if N^k objects are in the perceptual range of the sensor

objects. Accordingly, the more objects are in the perceptual field of the robot, the more likely it becomes that a feature is missing. Thus, 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.

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. In our current system we initialize the new filters by drawing samples from a uniform distribution over the state space. We then rely on the SJPDAF to disambiguate this distribution during subsequent filter updates. This turned out to work reliably in practice. 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:

$$\hat{W}_i^k = (1-\delta)\hat{W}_i^{k-1} + \delta W_i^k. \tag{15}$$

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 indica-

Figure 2: The RWI B21 robot Rhino used for the experiments.

tor 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 $\hat{W}_i^k$.

5 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 with a mobile robot. The states of the persons are represented by quadruples $\langle x, y, \phi, v \rangle$, where x and y correspond to the position relative to the robot, ϕ is the orientation, and v is the walking speed of the person. The experiments were carried out using our mobile platform Rhino, an RWI B21 robot (see Figure 2). This robot is equipped with two laser range-finders 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 robustly identify and keep track of persons, our system uses a combination of different features. In the remainder of this section we will describe how we use the data obtained with the range scanners to generate the features that are the input to the SJPDAF.

Figure 3: 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).

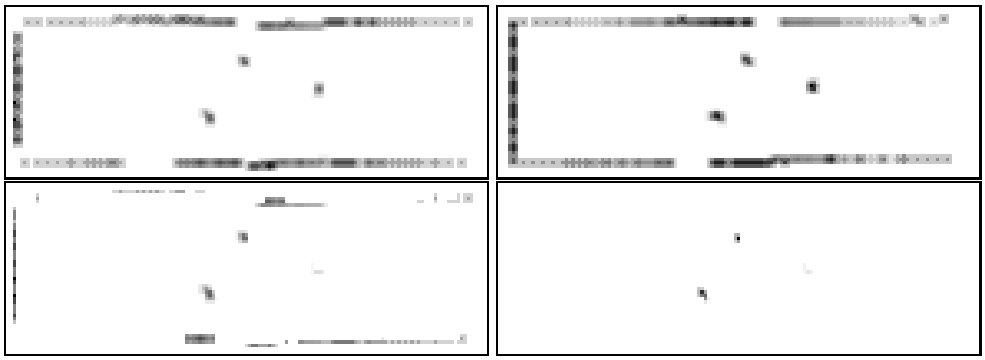

Figure 4: From left to right, top-down: the current occupancy map $P(\text{occ}_{x,y} \mid \mathbf{Z}(k))$, the previous occupancy map $P(\text{occ}_{x,y} \mid \mathbf{Z}(k-1))$, the resulting difference map $P(\text{new}^k_{x,y})$, and the fusion of the difference map with the feature maps for the scan depicted in Figure 3

Typically, people walking in the sensor range of the laser scanner result in local minima in the distance histograms. Therefore we compute a set of two-dimensional position probability grids, one for each local minimum, containing in each cell the probability $P(\text{legs}^k_{x,y})$ that a person's legs are at position $\langle x, y \rangle$ relative to the robot. The left image of Figure 3 shows a typical range scan with three local minima. An overlay of the histograms computed based on these local minima is shown in the center image of Figure 3.

Unfortunately, there are other objects in typical office environments which produce patterns similar to people. Consider, as an example, the situation shown in the left image of Figure 3. In this situation two people passed the robot and caused two local minima in the distance histogram of the range scan. The third

minimum, however, is caused by a trash bin placed in the center of the corridor. To deal with this problem our system additionally considers the changes in consecutive scans. We compute local occupancy grid maps [46] 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 a standard scan-matching technique [42]. Figure 4 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 step of the particle filter, 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

$$P(\mathbf{z}_j(k) \mid x^k_{i,n}) = P(\text{legs}^k_{x_{i,n}}) \cdot P(\text{new}^k_{x_{i,n}}). \tag{16}$$

Additionally we have to compute the likelihood $p(\mathbf{z}_0(k) \mid x_{i,n}(k))$, that the person represented 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 3 is depicted in the right image of Figure 3. To determine $P(\mathbf{z}_0(k) \mid x_{i,n}(k))$ we use this occlusion map and a fixed feature detection failure probability $P(\neg\text{Detect})$:

$$P(\mathbf{z}_0(k) \mid x_{i,n}(k)) = P(\text{occluded}^k_{x_{i,n}} \vee \neg\text{Detect}). \tag{17}$$

Additionally, we employ information about static objects visible in the current laser scan to avoid that samples end up inside of objects or move through objects. For this purpose, we compute an occupancy probability grid from the static part of the current scan, i.e. using all beams that do not correspond to a moving feature. Based on this grid we reject motions of samples that do not entirely lead through free space.

6 Experimental Results

We have implemented the SJPDAF and applied it to tracking people with our mobile robot Rhino (see Figure 2). The current implementation is able to integrate laser range-scans at a rate of 3Hz. We carried out a series of real-robot and simulation experiments. During the experiments, the robot was moving with speeds of up to 40 cm/s. Because the ground truth information about the positions of persons is hard to obtain, we additionally carried out simulation experiments using a B21 simulator. In these experiments 1000 samples were used for each particle filter and the false alarm probability γ was set to 0.1. To model the motions of persons we assume that a person changes its walking direction and walking speed according to Gaussian distributions. The translational velocity is assumed to lie between 0 and 150 cm/s. The goal of the experiments described in this section is to demonstrate that SJPDAFs can robustly and accurately keep track of a varying number of moving objects. They also illustrate that using our features, the SJPDAF can track multiple persons even from a moving robot and even if persons temporarily occlude each other. Additionally, we compare SJPDAFs to standard JPDAFs and demonstrate that SJPDAFs are more robust than JPDAFs with respect to the number of correct associations as well as with respect to the tracking error.

6.1 Tracking Multiple Persons

The first experiment in this section is designed to illustrate the capabilities of our approach to track multiple persons. In this experiment Rhino was placed in the corridor of the computer science department of the University of Bonn. The robot's task was to continuously estimate the position of up to six persons which were walking along the corridor. Please note that computing the joint association probabilities would require to evaluate more than 40,000 joint association events in some situations during this experiment. To keep the computation tractable, we did only consider feature to object assignments which have a marginal association probability of at least 0.01.

Figure 5 shows two sequences of this experiment.[1] Each sequence consists of pairs of images, an image recorded with a ceiling-mounted camera (left image) as

[1] Videos and animations of this sequence are available at http://www.informatik.uni-bonn.de/~schulz/research.html

Figure 5: Two short tracking sequences showing a real photo and a 3D visualization of the state estimate at the same point in time. The time delay between consecutive images is 1.25 seconds.

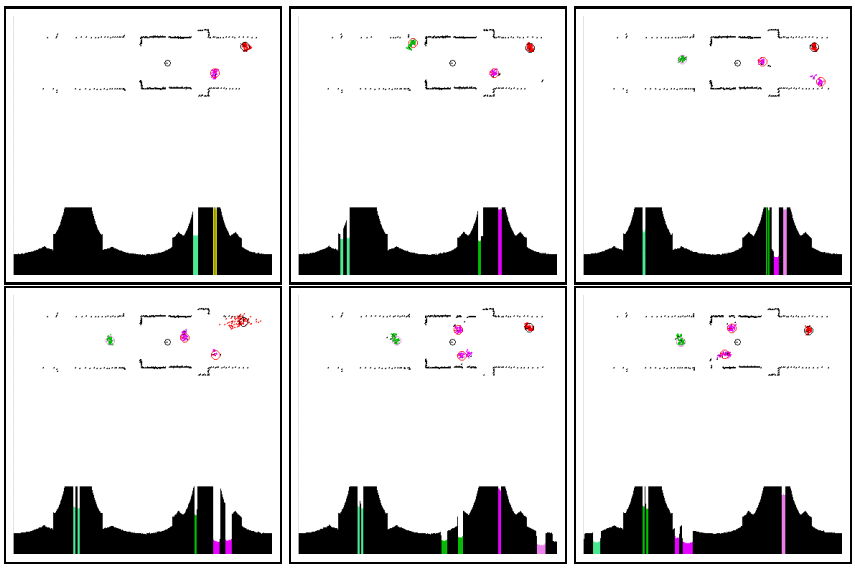

Figure 6: Tracking people using laser range-finder data.

well as a 3D visualization according to the current estimate of our system at the same point in time (right image). The time-delay between consecutive images is 1.25 seconds. Whereas the robot is standing still in the first sequence, it moves with speeds of up to 40 cm/sec in the second column. As can be seen from the figure, the robot is able to accurately estimate the positions of the persons even while it is moving and even if the persons are occluding each other.

Figure 6 depicts a typical situation in which Rhino is tracking four persons in its vicinity. It shows the sample sets as well as the corresponding range scans. As can be seen from the figure, our approach is robust against occlusions and can quickly adapt to changing situations in which additional persons enter the scene. For example, in the lower left image the upper right person is not visible in the range scan, since it is occluded by the person that is close to the robot. The knowledge that the samples lie in an occluded area prevents the robot from deleting the corresponding sample set. Instead, the samples only spread out, which represents the growing uncertainty of the robot about the position of the person.

Figure 7 plots the number of features detected during this experiment for the first 950 scans. As can be seen, the number of persons varies constantly. Please note that the maximum number of features is seven. This is due to the fact that

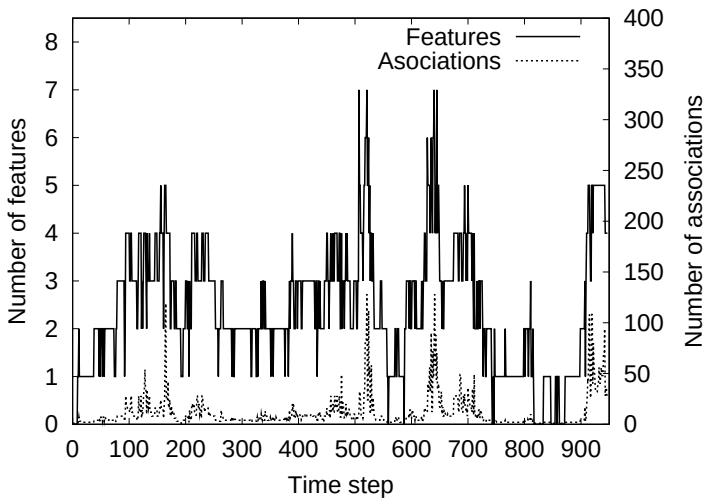

Figure 7: Number of features detected in each scan of the experiment depicted in Figure 5 and the number of joint association events evaluated for each scan.

doors also cause local minima which sometimes are identified as features due to slight scan alignment errors. The figure also shows the number of associations that are considered by our SJPDAF. As the figure shows, the number of associations that have to be evaluated stays below 150 during the whole sequence.

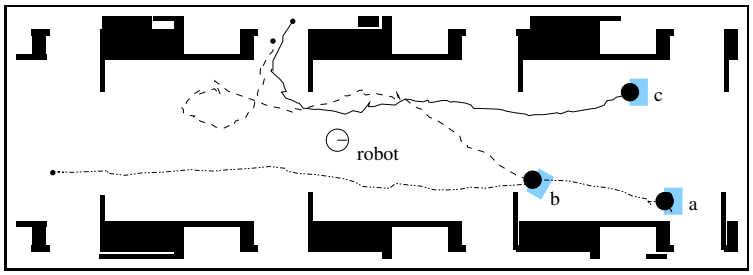

Figure 8: Trajectories of three persons walking along the corridor while the robot is tracking them.

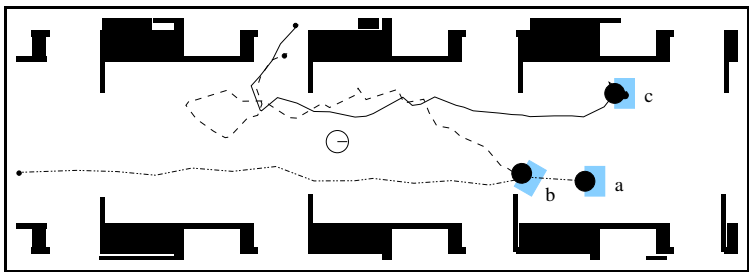

Figure 9: Trajectories estimated by the SJPDAF.

6.2 Accuracy of SJPDAF-Tracking

The next experiments were designed to identify the accuracy of our approach. Again we placed our robot 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 there were 16 different tracks, i.e. we observed 16 situations in which a person entered the robot's sensor range.

A short fraction of the whole experiment is shown in Figure 8. Here three people are walking along the corridor. The image shows the ground truth manually extracted from the data recorded during the experiment. Figure 9 depicts the trajectories estimated by our algorithm. As can be seen, the system is able to quite accurately keep track of the persons. 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 caused by person b.

To quantitatively evaluate the accuracy of the tracking process and of the estimator for the number of objects we manually determined the number of people and their positions in each scan. We then compared these values to the results obtained with our algorithm. First, our SJPDAF was able to correctly identify all 16 tracks. Thereby, the average displacement between the ground truth and the estimated position was 16 ± 5 cm. Second, our algorithm was able to robustly estimate the number of tracks. We found that the average detection delay of a new or disappearing track was 1 sec. At this point we would like to mention that the estimation process is a filter, which smoothes the effect of random feature detection failures and false alarms but also introduces a delay in the detection of the correct number of objects. Neglecting this delay effect, our estimator correctly

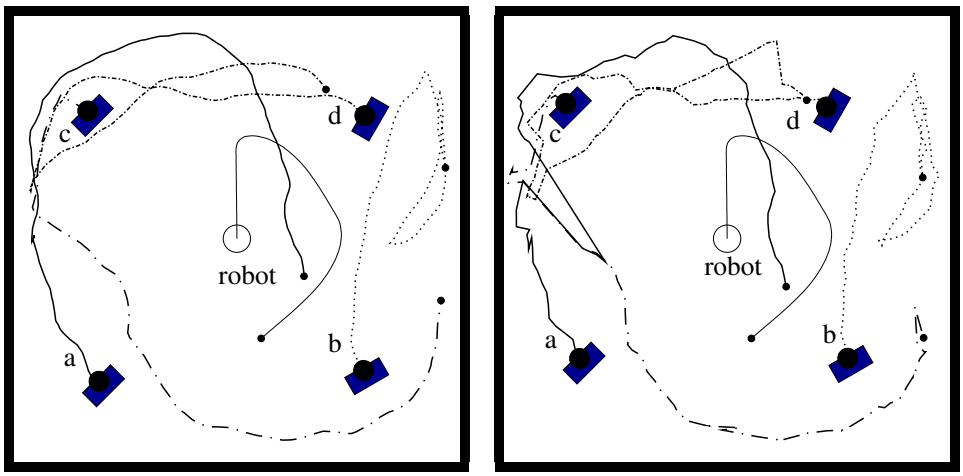

Figure 10: Tracking four objects with a moving robot. The real trajectories are shown on the left side. The right image contains the estimated paths.

determined the number of objects in 90% of all scans.

Additionally, we carried out several simulation experiments. In one experiment reported here the task of the system was to track four moving objects while the robot was moving at speeds up to 25 cm/s. The objects moved with a constant velocity of 50 cm/s and changed their movement direction according to a Gaussian distribution with a standard deviation of 45 degrees.

The left image of Figure 10 shows a typical fraction of this experiment. The right image of the same figure shows the estimates of the positions of the objects

Table 1: Tracking errors for the different objects in the experiment depicted in Figure 10.

Track	RMS error [cm]	std. dev. [cm]	min. error [cm]	max. error[cm]
a	34.79	56.04	8.20	282.76
b	22.98	24.11	6.70	37.78
c	24.52	37.60	5.40	170.73
d	28.87	33.87	8.69	90.17

in this situation. The average, mean, minimum, and maximum tracking errors for the four different objects are given in Table 1. Please note that the huge maximum error for track a comes from an occlusion by track c when both objects are to the left of the robot. According to our motion model, the samples for a immediately spread out in the area occluded by c. After a few steps, some of the samples for track a get close to c and therefore receive support from the feature corresponding to c. Accordingly, the SJPDAF infers a slight probability that the feature corresponding to c actually corresponds to a. Since the samples in the occluded area are almost uniformly distributed whereas the samples close to c are concentrated, the estimate for a coincides with the estimate for c for a few steps. However, as soon as the feature for a becomes visible again the SJPDAF re-localizes a and correctly keeps track of it afterwards.

6.3 Advantage of the Occlusion Handling

To explore 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 this experiment, two persons were walking in the corridor of our department (see Figure 11) with speeds of 60 cm/s and 80 cm/s. As can be seen in the figure, one person temporarily occludes the other person. From this data set we created 40 different sequences that were used for the quantitative analysis. For all 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 (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%).

6.4 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 and provides a significantly increased robustness.

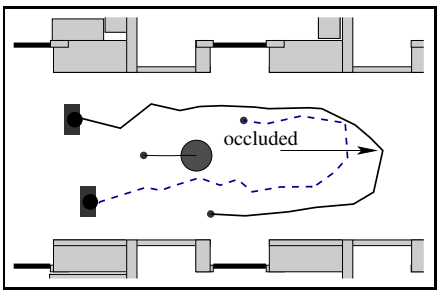

Figure 11: Estimated trajectories of two persons in a situation in which one person temporarily occludes the other. The arrow indicates the point in time, when the occlusion occurred.

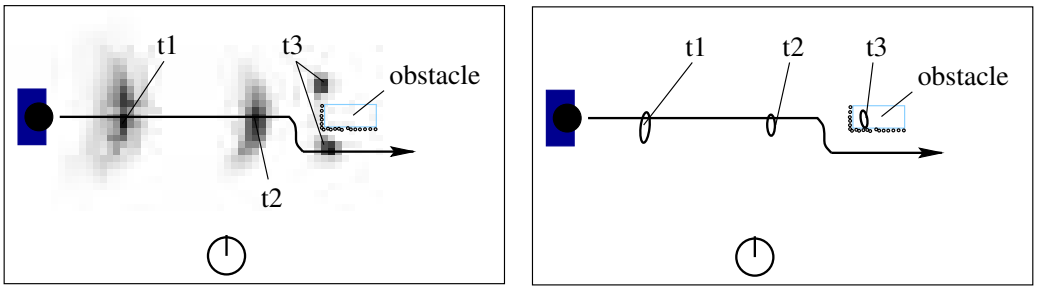

Figure 12: Tracking one object approaching a static obstacle; using a particle filter (left) and using a Kalman filter (right). The arrow indicates the trajectory taken by the object. The Kalman filter erroneously predicts that the object moves into the obstacle.

6.4.1 Prediction Accuracy

One advantage of particle filters lies in their high flexibility during the prediction step. For example, they also allow to incorporate obstacles during the prediction. Figure 12 shows a typical situation. 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 12 also contains the probability densities of the particle filter 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

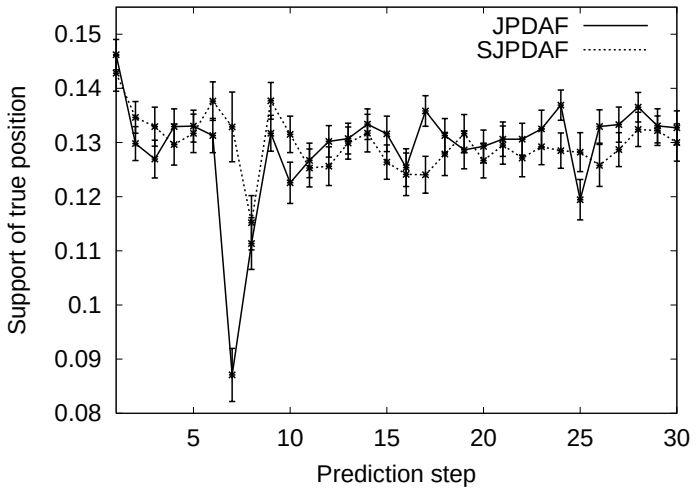

Figure 13: Probability mass of the predicted state densities at the true locations of the object for the experiment illustrated in Figure 12. The false prediction of the Kalman filter leads to a low support at the true location.

the same points in time by the standard JPDAF using a Kalman filter are shown in the right image of Figure 12. 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 moves through the obstacle. Our SJPDAF, in contrast, correctly detects that the person must pass the obstacle either on the left or on the right. This is indicated by the bimodal distribution shown in the left image of Figure 12. As a result, the SJPDAF provides a better support for the next measurement than the standard JPDAF.

To quantitatively evaluate the prediction accuracy of SJPDAFs and JPDAFs, we ran this experiment a 100 times in simulation. During these experiments the object passed the obstacle always at time step 7 but randomly either on the left or on the right. Figure 13 shows the average probability mass of the predicted state density within a region of 80 cm $\times$ 80 cm around the true position of the object plotted over each time step. Whereas the support of the JPDAF for the true position decreases drastically, the SJPDAF concentrates more probability mass at the possible locations to the left and right of the obstacle. Accordingly, the SJPDAF provides a significantly better support for the true position than the JPDAF. Please note that person again changes its motion direction in step 8. Therefore, both fil-

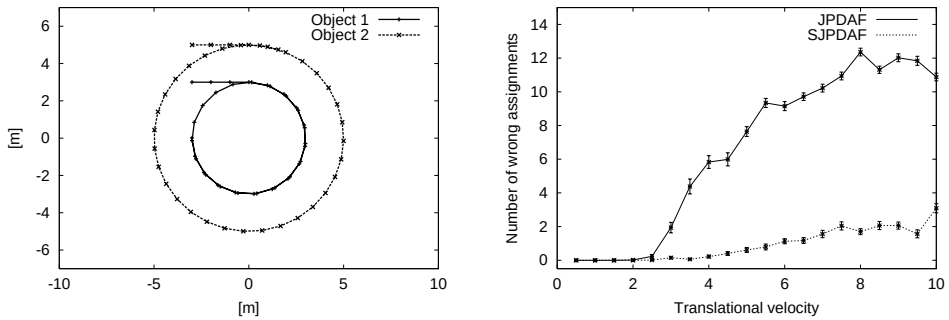

Figure 14: Trajectory of two objects moving on concentric circles (left image) and average number of wrong associations depending on translational velocity (right image).

ters have a reduced support for the true position of the person. Since the SJPDAF cannot exploit an obstacle as in step 7, the support of the true position is reduced and similar to that of the JPDAF.

6.4.2 Non-Linearities

If the objects being tracked follow a non-linear motion model, the correct uncertainty of the object state estimates is no longer normally distributed. In such situations, the standard JPDAF applies Extended Kalman Filters (EKFs) to approximate the covariance of the object state estimates. The approximation is then based on a linearization of the motion model. However, if observations arrive at low rates, rather large linearization errors can result. In contrast to this, the SJPDAF directly applies the non-linear motion model to the samples and this way achieves a more accurate approximation of the state uncertainty. As a result, feature to object assignments calculated by the SJPDAF are significantly more robust than the assignments obtained by the standard JPDAF.

To demonstrate this effect, we carried out an additional simulation experiment (see Figure 14). Here, two objects first move in parallel on a straight line and then continue on two concentric circles at the same constant speed. We examined the robustness of the tracking algorithms by counting the number of data association failures. To represent the state of each object we used a five-tuple $\langle x, y, \theta, v, \omega \rangle$ where x, y, and θ are the pose and v and ω are the translational and rotational velocities. We furthermore assumed Gaussian noise in both velocities. We re-

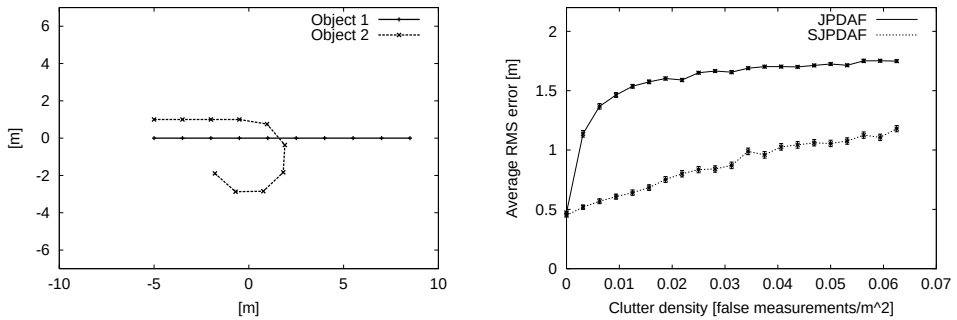

Figure 15: Trajectory of the two objects being tracked (left image) and resulting average RMS error for increasing clutter densities (right image).

peated this experiment for different translational velocities v with a fixed standard deviation of 0.1 m/s. As the standard deviation of the rotational velocity we used r/v, where r is the constant radius of the circular trajectory. For each value of v we performed 100 Monte Carlo runs to determine the average number of wrong assignments. Observations were integrated at a rate of one observation per second.

The results for the JPDAF and the SJPDAF are depicted in the right part of Figure 14. As can be seen from the figure, the SJPDAF shows significantly lower number of association errors if the speed of the objects exceeds 2.5 m/s.

6.4.3 Clutter

The use of particle filters also improves the robustness of multi-object tracking in cluttered environments. To demonstrate this effect, we carried out a third simulation experiment, during which we generated different amounts of false measurements. Again, the systems had to track two objects that moved in parallel for 5 seconds. After that, one of the objects took a circular trajectory with a diameter of 4 m (see left image of Figure 15). Both objects, for which we used the same linear motion model, moved at a constant velocity of 1 m/s. We assumed white noise with a standard deviation of 0.1 m/s in the translational velocity and of 0.5 rad in rotational velocity. New observations were integrated every 1.5 seconds and measurement errors were assumed to be normally distributed with a standard deviation of 0.2 m in the x- and y-direction.

We carried out 400 Monte Carlo for different clutter densities. The resulting average RMS error wrt. the density of false measurements in the surveillance re-

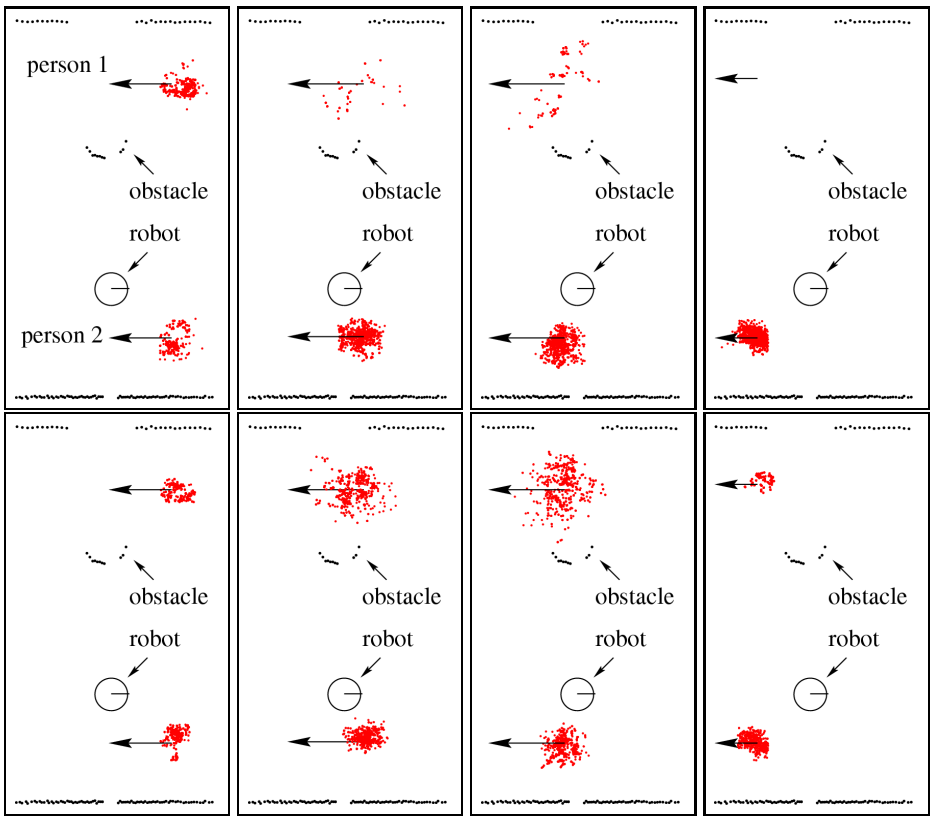

Figure 16: Tracking two persons with one sample set (top row) and with two sample sets (bottom row). The arrows indicate the position of the persons and their movement direction.

gion is depicted in the left part of Figure 15. As can bee seen from the figure, the RMS error increases quickly in case of the standard JPDAF. This is mainly because the JPDAF loses track of the object moving on the circle. In contrast to that, the SJPDAF shows significantly smaller errors. Please note that this experiment confirms an observation already made by Lin et al. [40]. The authors showed that the particle filter performs significantly better in situations in which large intermediate state estimation errors occur.

6.5 Advantage of the SJPDAF over Standard Particle Filters

In the past, single state particle filters have also been used for tracking multiple objects [30, 35]. 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 un-occluded object.

Figure 16 shows an example situation in which the robot is tracking two persons. Whereas one person is visible all the time, the second person is temporarily occluded by a static obstacle. The upper row of the figure shows the evolution of the samples using 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 concentrate on the un-occluded object so that 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.

7 Summary and Conclusions

Mobile robots are now in a state where they can safely be deployed in populated environments. In this situation, the ability to know where persons are and where they are going to becomes increasingly important. Knowledge about the trajectories of persons can be used to to improve the navigation behavior and the interaction capabilities. In this paper we presented a new technique for keeping track of multiple moving objects. Our approach uses SJPDAFs, a sample-based variant of joint probabilistic data association filters. It also includes a recursive Bayesian filter to deal with varying numbers of objects. The use of particle filters has several desirable advantages. SJPDAFs can represent arbitrary densities over the state spaces of the individual objects which leads to more accurate estimates especially in the prediction step. The fact that particle filters can easily deal with nonlinear systems results in a more robust behavior and fewer data association errors. Finally, the SJPDAF can more robustly deal with situations with a huge amount of clutter.

Our algorithm has been implemented and evaluated in practice using a mobile robot equipped with laser range-finders. Additionally, we performed a variety of simulation experiments. All results demonstrate that SJPDAFs are able to reliably and accurately keep track of a varying number of objects. They also illustrate that our algorithm outperforms other techniques developed so far.

Despite the encouraging results, there still are several warrants for future research. For example, the current system uses fixed sample sizes for the particle filters. In this context, techniques to adaptively change the sample size according to the underlying uncertainty [19] might be attractive to improve the overall efficiency. Additionally, our current system randomly introduces samples whenever a new object has been discovered. In such a situation more intelligent sampling techniques might lead to better results and a faster convergence. A further topic for future work regards the distinction between groups of objects and individual objects. Sometimes it is desirable to track a whole group of people rather than the individual persons. For example, tracking them individually might be too time consuming or just impossible because we cannot extract all individual features from the sensor data. In such situations it might be more desirable to track groups of people as a whole. However, this requires a new data association algorithm, which is able to handle both individual objects and clusters of objects.

8 Acknowledgments

This work is sponsored in part by the IST Programme of the Commission of the European Communities under contract numbers IST-1999-12643 and IST-2000-29456 and by the National Science Foundation (CAREER grant number 0093406) and by DARPA (MICA program).

References

[1] J. K. Aggarwal and Q. Cai. Human motion analysis: A review. *Computer Vision and Image Understanding: CVIU*, 73(3):428–440, 1999.

[2] K.O. Arras and S.J. Vestli. Hybrid, high-precision localization for the mail distributing mobile robot system MOPS. In *Proc. of the IEEE International Conference on Robotics & Automation (ICRA)*, 1998.

[3] H. Asoh, S. Hayamizu, I. Hara, Y. Motomura, S. Akaho, and T. Matsui. Socially embedded learning of office-conversant robot jijo-2. In *Proceedings of IJCAI-97*. IJCAI, Inc., 1997.

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

[5] M. Bennewitz, W. Burgard, and S. Thrun. Learning motion patterns of persons for mobile service robots. In *Proc. of the IEEE International Conference on Robotics & Automation (ICRA)*, 2002.

[6] D. Beymer and Konolige K. Tracking people from a mobile platform. In *IJCAI-2001 Workshop on Reasoning with Uncertainty in Robotics*, 2001.

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

[8] H. Bui, S. Venkatesh, and G. West. Tracking and surveillance in wide-area spatial environments using the Abstract Hidden Markov Model. *Intl. J. of Pattern Rec. and AI*, 2001.

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

[10] I. J. Cox, M. L. Miller, R. Danchick, and G. E. Newnam. A comparison of two algorithms for determining ranked assignments with application to multi-target tracking and motion correspondence. *IEEE Trans. on Aerospace and Electronic Systems*, 33(1):295–301, 1997.

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

[12] 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. *IEEE Transactions on PAMI*, 18(2):138–150, February 1996.

[13] T. Darrell, B. Moghaddam, and A. P. Pentland. Active face tracking and pose estimation in an interactive room. In *Proc. of the IEEE Sixth International Conference on Computer Vision*, pages 67–72, 1996.

[14] F. Dellaert, D. Fox, W. Burgard, and S. Thrun. Monte Carlo localization for mobile robots. In *Proc. of the IEEE International Conference on Robotics & Automation (ICRA)*, 1999.

[15] A. Doucet, N. de Freitas, and N. Gordon, editors. *Sequential Monte Carlo Methods in Practice*. Springer Verlag, New York, January 2001.

[16] H. Endres, W. Feiten, and G. Lawitzky. Field test of a navigation system: Autonomous cleaning in supermarkets. In *Proc. of the IEEE International Conference on Robotics & Automation (ICRA)*, 1998.

[17] J. F. Engelberger. Health-care robotics goes commercial: The 'helpmate' experience. *Robotica*, 11:517–523, 1993.

[18] A. Fod, A. Howard, and M. J. Matarić. Laser-based people tracking. In *Proc. of the IEEE International Conference on Robotics & Automation (ICRA)*, 2002.

[19] D. Fox. KLD-Sampling: Adaptive particle filters. In *In Advances in Neural Information Processing Systems 14 (NIPS)*, 2002.

[20] 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.

[21] D. Fox, S. Thrun, F. Dellaert, and W. Burgard. Particle filters for mobile robot localization. In Doucet et al. [15].

[22] H.H. González-Baños, C.Y. Lee, and J.C. Latombe. Real-time combinatorial tracking of a target moving unpredictably among obstacles. In *Proc. of the IEEE International Conference on Robotics & Automation (ICRA)*, 2002.

[23] N.J. Gordon. A hybrid bootstrap filter for target tracking in clutter. *IEEE Transactions on Aerospace and Electronic Systems*, 33(1):353–358, January 1997.

[24] 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.

[25] H.-M. Gross, H.-J. Boehme, and A. König. Vision-based Monte-Carlo self-localization for a mobile service robot acting as shopping assistant in a home store. In *Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2002.

[26] 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.

[27] D. Hähnel, D. Schulz, and W. Burgard. Map building with mobile robots in populated environments. In *Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2002.

[28] I. Horswill. Polly: A vision-based artificial agent. In *Proc. of the National Conference on Artificial Intelligence (AAAI)*, 1993.

[29] J. Illmann, B. Kluge, and E. Prassler. Statistical recognition of motion patterns. In *Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2002.

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

[31] B. Jung and G.S. Sukhatme. A region-based approach for cooperative multi-target tracking in a structural environment. In *Proc. of the IEEE International Conference on Robotics & Automation (ICRA)*, 2002.

[32] R.E. Kahn, M.J. Swain, P.N. Prokopowicz, and R.J. Firby. Gesture recognition using the perseus architecture. Technical Report TR-96-04, University of Chicago, 19, 1996.

[33] 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.

[34] B. Kluge, C. Koehler, and E. Prassler. Fast and robust tracking of multiple moving objects with a laser range finder. In *Proc. of the IEEE International Conference on Robotics & Automation (ICRA)*, 2001.

[35] E. Koller-Meier and F. Ade. Tracking multiple objects using the condensation algorithm. *Journal of Robotics and Autonomous Systems*, 34(2-3):93–105, 2001.

[36] D. Kortenkamp, E. Huber, and R. P. Bonasso. Recognizing and interpreting gestures on a mobile robot. In *Proc. of the American Conference on Artificial Intelligence*, 1996.

[37] E. Kruse and F. Wahl. Camera-based monitoring system for mobile robot guidance. In *Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 1998.

[38] S. M. Lavalle, H. H. González-Banos, G. Becker, and J.-C. Latombe. Motion strategies for maintaining visibility of a moving target. In *Proc. of the IEEE International Conference on Robotics & Automation (ICRA)*, 1997.

[39] S. Lenser and M. Veloso. Sensor resetting localization for poorly modelled mobile robots. In *Proc. of the IEEE International Conference on Robotics & Automation (ICRA)*, 2000.

[40] X. Lin, T. Kirubarajan, Y. Bar-Shalom, and S. Maskell. Comparison of EKF, pseudomeasurement filter, and particle filter for a bearing-only target tracking problem. In *Proceedings of SPIE Vol. 4728*, 2002.

[41] M. Lindström and J.-O. Eklundh. Detecting and tracking moving objects from a mobile platform using a laser range scanner. In *Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2001.

[42] 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.

[43] 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.

[44] M. Montemerlo, S. Thun, D. Koller, and B. Wegbreit. FastSLAM: A factored solution to simultaneous mapping and localization. In *Proc. of the National Conference on Artificial Intelligence (AAAI)*, 2002.

[45] M. Montemerlo, S. Thun, and W. Whittaker. Conditional particle filters for simultaneous mobile robot localization and people-tracking. In *Proc. of the IEEE International Conference on Robotics & Automation (ICRA)*, 2002.

[46] 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.

[47] K. Murphy and S. Russell. Rao-blackwellised particle filtering for dynamic bayesian networks. In Doucet et al. [15].

[48] R. Murrieta-Cid, H.H. González-Baños, and B. Tovar. A reactive motion planner to maintain visibility of unpredictable targets. In *Proc. of the IEEE International Conference on Robotics & Automation (ICRA)*, 2002.

[49] L. E. Parker. Cooperative motion control for multi-target observation. In *Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 1997.

[50] P. Pirjanian and M. Matarić. Multi-robot target acquisition using multiple objective behaviour coordination. In *Proc. of the IEEE International Conference on Robotics & Automation (ICRA)*, 2000.

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

[52] C. Rasmussen and G.D. Hager. Joint probabilistic techniques for tracking multi-part objects. In *Proc. of the International Conference on Computer Vision and Pattern Recognition (CVPR)*, 1998.

[53] R. Simmons, R. Goodwin, K. Haigh, S. Koenig, and J. O'Sullivan. A layered architecture for office delivery robots. In *Proc. of the First International Conference on Autonomous Agents*, Marina del Rey, CA, 1997.

[54] S. Tadokoro, M. Hayashi, Y. Manabe, Y. Nakami, and T. Takamori. On motion planning of mobile robots which coexist and cooperate with human. In *Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 1995.

[55] S. Thrun, M. Bennewitz, W. Burgard, A.B. Cremers, F. Dellaert, D. Fox, D. Hähnel, C. Rosenberg, N. Roy, J. Schulte, and D. Schulz. MINERVA:

A second generation mobile tour-guide robot. In *Proc. of the IEEE International Conference on Robotics & Automation (ICRA)*, 1999.

[56] N. Vlassis, B. Terwijn, and B. Kröse. Auxiliary particle filter robot localization from high-dimensional sensor observations. In *Proc. of the IEEE International Conference on Robotics & Automation (ICRA)*, 2002.

[57] S. Waldherr, S. Thrun, R. Romero, and D. Margaritis. Template-based recognition of pose and motion gestures on a mobile robot. In *Proc. of the National Conference on Artificial Intelligence (AAAI)*, 1998.

[58] J. Wolf, W. Burgard, and H. Burkhardt. Robust vision-based localization for mobile robots using an image retrieval system based on invariant features. In *Proc. of the IEEE International Conference on Robotics & Automation (ICRA)*, 2002.

[59] C. R. Wren, A. Azarbayejani, T. Darrell, and A. P. Pentland. Pfinder: Real-time tracking of the human body. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(7):780–785, 1997.

[60] Q. Zhu. Hidden Markov model for dynamic obstacle avoidance of mobile robot navigation. *IEEE Transactions on Robotics and Automation*, 7(3), 1991.