Learning Motion Patterns of People for Compliant Robot Motion

Maren Bennewitz[†] Wolfram Burgard[†]
Grzegorz Cielniak[‡] Sebastian Thrun[§]

[†]*Department of Computer Science, University of Freiburg, 79110 Freiburg, Germany*
[‡]*Department of Technology, Örebro University, 70182 Örebro, Sweden*
[§]*School of Computer Science, Carnegie Mellon University, Pittsburgh PA, USA*

Abstract

Whenever people move through their environments they do not move randomly. Instead, they usually follow specific trajectories or motion patterns corresponding to their intentions. Knowledge about such patterns enables a mobile robot to robustly keep track of persons in its environment and to improve its behavior. This paper proposes a technique for learning collections of trajectories that characterize typical motion patterns of persons. Data recorded with laser-range finders is clustered using the expectation maximization algorithm. Based on the result of the clustering process we derive a Hidden Markov Model (HMM) that is applied to estimate the current and future positions of persons based on sensory input. We also describe how to incorporate the probabilistic belief about the potential trajectories of persons into the path planning process. We present several experiments carried out in different environments with a mobile robot equipped with a laser range scanner and a camera system. The results demonstrate that our approach can reliably learn motion patterns of persons, can robustly estimate and predict positions of persons, and can be used to improve the navigation behavior of a mobile robot.

1 Introduction

Recently, there has been a substantial progress in the field of service robots (see for example the book by Schraft and Schmierer [57] for an overview) and a variety of mobile robots has already been developed that are designed to operate in populated environments. These robots, for example, have been deployed in hospitals [9, 28], office buildings [1, 2, 21, 47, 62], department stores [8, 18], and museums [4, 61, 67]. Existing robotic systems are already able to perform various services such as deliver, educate, provide tele-presence [15, 20, 59], clean [23], or entertain [68]. Furthermore there are prototypes of autonomous wheelchairs [29, 33] and intelligent service robots which are designed to assist people in their homes [32, 41, 56].

Obviously, robots operating in populated environments can improve their service if they react appropriately to the activities of the people in their surrounding and do not interfere with them. This requires that robots can locate and track persons using their sensors. Furthermore, the robots need to be able to identify and potentially learn intentions of people so that they can make better prediction about their future actions. In the past, various approaches have been presented to track the positions of persons (see for example [29, 42, 58]) or to predict their short-term motions [66, 69]. These approaches assume that motion models of the persons are given. A lot of research has already been focused on the problem of learning (e.g. [14, 48, 54]) and recognizing behaviors or plans (e.g. [16, 27, 45]) of persons. Additionally, systems have been developed to detect atypical behaviors or unusual events (see for example [24, 64]).

In this paper we present an approach that, in contrast to these previous approaches which are discussed in detail in Section 6, allows a mobile robot to learn motion patterns of persons from sensor data, to utilize these learned patterns to maintain a belief about where the persons are, and to adapt the navigation behavior of the robot. Such capabilities can be useful in various kinds of situations. For example, they allow a robot to reliably predict the trajectory of a person and to avoid that the robot blocks the path of that person. Furthermore, a home-care robot can more robustly keep track of the person it is providing service to and this way increase the time it stays in the vicinity of the person, for example to support necessary interactions. Thus, the knowledge about motion patterns of a person can be quite useful for several tasks such as collision avoidance, strategic positioning, and verbal assistance.

Our approach to learning motion patterns of persons is purely probabilistic. It is motivated by the observation that people usually do not move randomly when they move through their environments. Instead, they usually engage in motion patterns, related to typical activities or specific locations they might be interested in approaching. The input to our algorithm is a set of trajectories of persons between so-called resting places where the persons typically stop and stay for a certain period of time. Such places can be desks in office environments or the TV set in home environments. Our approach clusters these trajectories into motion patterns using the EM algorithm [39]. It then derives Hidden Markov Models (HMMs) [51] from the learned motion patterns and uses these HMMs to maintain a belief about the positions of persons. Several experiments carried out with a mobile robot in different environments demonstrate that our algorithm is able to reliably learn motion patterns of persons and to use this information for an accurate estimation of the positions of the persons. Additionally, we describe how the robot can use the motion patterns for predicting future poses of the persons and how it can choose appropriate navigation actions based on these predictions.

The paper is organized as follows. The next section introduces our approach to learn typical motion patterns from observed trajectories. Section 3 contains a description of how we derive Hidden Markov Models from the learned motion patterns to predict motions of persons. In Section 4 we explain our technique to detect and identify persons using sensor data. In Section 5 we present several experiments carried out in different environments illustrating that our approach can robustly learn typical motion patterns. We also describe experimental results demonstrating that the Hidden Markov Models derived from the learned models can be used to reliably estimate and

predict the positions of multiple persons with a mobile robot based on laser and vision data. Finally, Section 6 contains a discussion of related work.

2 Learning Motion Patterns of Persons

When people perform everyday activities in their environments they do not move permanently. They usually stop at several locations denoted as resting places and stay there for a certain period of time, depending on what activity they are currently carrying out. Accordingly, we assume that the input to our algorithm is a collection of trajectories $s = \{s_1, \ldots, s_I\}$ between resting places. The output is a number of different types of motion patterns $\theta = \{\theta_1, \ldots, \theta_M\}$ a person might exhibit in its natural environment. Each trajectory s_i consists of a sequence $s_i = \{s_i^1, s_i^2, \ldots, s_i^{T_i}\}$ of positions s_i^t. Accordingly, s_i^1 is the resting place the person leaves and $s_i^{T_i}$ is the destination. The task of the algorithm presented here is to cluster these trajectories into different motion patterns. Our method is an extension of the k-Means algorithm to the multi-step case that independently applies k-Means to each step in a normalized trajectory.

2.1 Motion Patterns

We begin with the description of our model of motion patterns, which is subsequently estimated from data using EM. A motion pattern denoted as θ_m with $1 \leq m \leq M$ is represented by K probability distributions $P(x \mid \theta_m^k)$ where M is the number of different types of motion patterns a person is engaged in.

Throughout this paper we assume that the input to our algorithm consists of trajectories which have the same number of observed positions, i.e., that $T_i = T$ for all i. To achieve this, we transform the trajectories in s into a set d of I trajectories such that each $d_i = \{x_i^1, x_i^2, \ldots, x_i^T\}$ has a fixed length T and is obtained from s_i by a linear interpolation. The length T of these trajectories corresponds to the maximum length of the input trajectories in s. The learning algorithm described below operates solely on $d_1, \ldots, d_I$ and does not take into account the velocities of the persons during the learning phase. In our experiments we never found evidence that the linear interpolation led to wrong results or that the walking speed of a person depends on the typical activity it is carrying out. Note, however, that one can easily extend our algorithm to also incorporate the velocities of the persons. This can be achieved by introducing further dimensions to the state variables.

For each θ_m^k the probability distribution $P(x \mid \theta_m^k)$ is computed based on $\beta = \lceil T/K \rceil$ subsequent positions on the trajectories. Accordingly, $P(x \mid \theta_m^k)$ specifies the probability that the person is at location x after $[(k-1) \cdot \beta + 1; k \cdot \beta]$ steps given that it is engaged in motion pattern m. Thus, we calculate the likelihood of a trajectory d_i under the m-th motion pattern θ_m as

$$P(d_i \mid \theta_m) \quad = \quad \prod_{t=1}^{T} P(x_i^t \mid \theta_m^{\lceil t/\beta \rceil}). \tag{1}$$

Note that we assume consecutive positions on the trajectories to be independent.

This is generally not justified, however, in our experiments we never found evidence that this led to wrong results.

2.2 Expectation Maximization

Throughout this paper we assume that each motion pattern is represented by K Gaussian distributions with a fixed standard deviation σ. Accordingly, the application of EM leads to an extension of the k-Means algorithm [6, 13, 38, 40] to trajectories. In essence, our approach seeks to identify a model θ that maximizes the likelihood of the data.

First we have to introduce a set of *correspondence variables*, denoted c_{im}. Here i is the index of the trajectory d_i and m is the index of the motion pattern θ_m. Each c_{im} is a binary variable, that is, it is either 0 or 1. It is 1 if and only if the i-th trajectory corresponds to the m-th motion pattern. If we think of the motion pattern as a specific motion activity a person might be engaged in, c_{im} is 1 if the person was engaged in motion activity m during trajectory i.

Note that if we knew the values of the correspondence variables learning of the motion patterns would be easy. But since those values are hidden we have to compute the model which has the highest expected data likelihood.

EM is an algorithm that iteratively maximizes expected log likelihood functions by optimizing a sequence of lower bounds. In particular, it generates a sequence of models, denoted $\theta^{[1]}, \theta^{[2]}, \ldots$ of increasing log likelihood.

The optimization involves two steps: calculating the expectations $E[c_{im} \mid \theta^{[j]}, d]$ given the current model $\theta^{[j]}$, and finding the new model $\theta^{[j+1]}$ that has the maximum expected likelihood under these expectations. The first of these two steps is typically referred to as the E-step (short for: expectation step), and the latter as the M-step (short for: maximization step).

To calculate the expectations $E[c_{im} \mid \theta^{[j]}, d]$ we apply Bayes' rule, obeying independence assumptions between different data trajectories:

$$\begin{aligned} E[c_{im} \mid \theta^{[j]}, d] &= P(c_{im} \mid \theta^{[j]}, d) = P(c_{im} \mid \theta^{[j]}, d_i) \\ &= \eta P(d_i \mid c_{im}, \theta^{[j]}) P(c_{im} \mid \theta^{[j]}) \\ &= \eta' P(d_i \mid \theta_m^{[j]}), \end{aligned} \tag{2}$$

where the normalization constants η and η' ensure that the expectations sum up to 1 over all m. Sine the probability distributions are represented by Gaussians we obtain:

$$E[c_{im} \mid \theta^{[j]}, d] = \eta' \prod_{t=1}^{T} e^{-\frac{1}{2\sigma^2} \|x_i^t - \mu_m^{[t/\beta][j]}\|^2}. \tag{3}$$

Finally, the M-step calculates a new model $\theta^{[j+1]}$ by maximizing the expected likelihood. Technically, this is done by computing for each model component m and for each probability distribution $P(x \mid \theta_m^{k[j+1]})$ a new mean $\mu_m^{k[j+1]}$ of the Gaussian distribution. Thereby we consider the expectations $E[c_{im} \mid \theta^{[j]}, d]$ computed in the E-step:

$$\mu_m^{k[j+1]} = \frac{1}{\beta} \cdot \sum_{t=(k-1)\cdot\beta+1}^{k\cdot\beta} \frac{\sum_{i=1}^{I} E[c_{im} \mid \theta^{[j]}, d] x_i^t}{\sum_{i=1}^{I} E[c_{im} \mid \theta^{[j]}, d]}. \tag{4}$$

2.3 Monitoring Convergence and Local Maxima

The EM algorithm is well-known to be sensitive to local maxima during the search. In the context of clustering local maxima correspond to situations in which data items are associated to wrong model components or clusters. Luckily, such cases can be identified quite reliably during EM. Our approach continuously monitors two types of occurrences to detect local maxima:

Low data likelihood: If a trajectory d_i has low likelihood under the model θ, this is an indication that no appropriate model component for d_i has yet been identified that explains this trajectory.

Low model component utility: The aim of this criterion is to discover multiple model components that basically represent the same motion pattern. To detect such cases, the total data log likelihood is calculated with and without a specific model component θ_m.

Whenever the EM algorithm has converged, our approach extracts those two statistics and considers "resetting" individual model components. In particular, if a low data likelihood trajectory is found, a new model component is introduced that is initialized using this very trajectory (this is an adaption of the partition expansion presented by Li et al. [35]). At the same time the model component which has the lowest utility is eliminated from the model. If no model component exists with a utility lower than a predefined threshold our algorithm terminates and returns the current set of model components.

In our experiments we found this selective resetting and elimination strategy extremely effective in escaping local maxima. Without this mechanism, the EM frequently got stuck in local maxima and generated models that were significantly less predictive of human motions.

2.4 Estimating the Number of Model Components

The approach presented above works well in the case that the actual number of different motion patterns is known. In general, however, the correct number of motion patterns is not known in advance. Thus, we need to determine this quantity during the learning phase. If the number of model components is wrong, we can distinguish two different situations. First, if there are too few model components, there must be trajectories, that are not explained well by any of the current model components. On the other hand, if there are too many model components there must be trajectories that are explained well by different model components. Whenever the EM algorithm has converged, our algorithm checks whether the model can be improved by increasing or decreasing the number of model components. During the search, we continuously monitor the two types of occurrences mentioned above: low data likelihood and low model component utility. If a low data likelihood trajectory is found, a new model component is introduced that it initialized using this very trajectory. Conversely, if a model component with low utility is found, it is eliminated from the model.

To limit the model complexity and to avoid overfitting we use the Bayesian Information Criterion [60] to evaluate a model $\theta^{[j]}$:

$$E_c[\ln P(d, c \mid \theta^{[j]}) \mid \theta^{[j]}, d] - \frac{M^{[j]}}{2} \log_2 I. \tag{5}$$

In this term $E_c[\ln P(d, c \mid \theta^{[j]} \mid \theta^{[j]}, d]$ is the total expected data log likelihood computed as

$$E_c[\ln P(d, c \mid \theta^{[j]}) \mid \theta^{[j]}, d]$$
$$= \sum_{i=1}^{I} \left(T \cdot M \cdot \ln \frac{1}{\sqrt{2\pi}\sigma} - \frac{1}{2\sigma^2} \sum_{t=1}^{T} \sum_{m=1}^{M} E[c_{im} \mid \theta^{[j]}, d] ||x_i^t - \mu_m^{\lceil t/\beta \rceil [j]}||^2 \right). \tag{6}$$

The Bayesian Information Criterion is a popular approach to score a model during clustering. It trades off the number of model components $M^{[j]}$ multiplied by the logarithm of the number of input trajectories with the quality of the model with respect to the given data.

Our algorithm terminates and returns the model with the best overall evaluation found so far after the maximum number of iterations has been reached or when the overall evaluation cannot be improved by increasing or decreasing the number of model components.

2.5 Laser-based Data Acquisition

The EM-based learning procedure has been implemented for data acquired with laser-range finders. To acquire the data we used several laser-range scanners which were installed in the environment such that the relevant parts of the environment were covered. The laser scanners were mounted on a height of approximately 30 cm. Typical range data obtained during the data acquisition phase are depicted in Figure 1.

To determine the trajectories that are the input to our algorithm our system first extracts features which are local minima in the range scans that come from the persons' legs. Additionally, it considers changes in consecutive scans to identify the moving people. After determining the positions of the person based on the range scans we proceed with the next step and determine the so-called resting places where the person frequently stays for a while. This can easily be done by identifying time periods in which the person does not move. Then we perform a segmentation of the data into different slices in which the person moves. Finally, we compute the trajectories which are the input to the learning algorithm described above, i.e. the sequence of positions covered by the person during that motion. When computing these trajectories we ignore positions which lie closer than 15 cm to each other. A typical result of this process is shown in Figure 2.

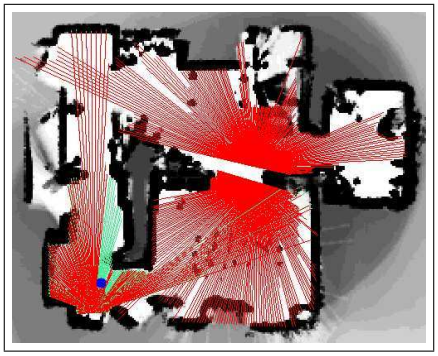

Figure 1: Typical laser range data obtained in a home environment equipped with three laser-range scanners. This data is used to extract resting places and trajectories of persons between these resting places.

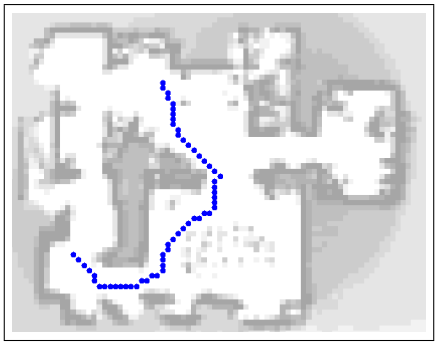

Figure 2: A single trajectory extracted from the laser data.

3 Deriving Hidden Markov Models from Learned Motion Patterns

So far we have described our approach to learning motion patterns of people. In the following we explain how to derive HMMs from the learned patterns which can be used to predict the motions of people.

As explained before, people usually do not permanently move. Rather they typically move between the resting places. Our approach assumes that the motion patterns of a person are given in the form of M trajectories θ_m each consisting of a sequence of points interconnected by line segments. Obviously, the initial point and the final point of each motion pattern must correspond to resting places. To derive a Hidden Markov Model from the given typical trajectories of a person we therefore distinguish two types of states. The first class are the initial and final states that correspond to the resting places. To connect these states we introduce so-called intermediate states which

lie on the motion patterns. In our current system we use a sequence of L_m intermediate states $\rho_m^1, \ldots, \rho_m^{L_m}$ for each motion pattern θ_m. The intermediate states are distributed over θ_m such that the distance between two consecutive states is $\Delta_\rho = 50$cm. Given this equidistant distribution of the intermediate-states and assuming a constant speed v with standard deviation σ_v of the person, the transition probabilities of this HMM depend on the length Δ_t of the time interval between consecutive updates of the HMM as well as on v and σ_v. In our current system this value is set to $\Delta_t = 0.5$ secs. Accordingly, we compute the probability $P(\rho'_m|\rho_m, \Delta_t)$ that the person is in state ρ'_m after ρ_m in the direction of the motion pattern given its current state is ρ_m and given that the time Δ_t has elapsed as:

$$P(\rho'_m|\rho_m, \Delta_t) = \int_{\rho'_m - \frac{\Delta_\rho}{2}}^{\rho'_m + \frac{\Delta_\rho}{2}} \mathcal{N}(\rho_m + v \cdot \Delta_t, \sigma_v, \rho)\, d\rho. \tag{7}$$

Here $\mathcal{N}(\rho_m + v \cdot \Delta_t, \sigma_v, \rho)$ is the quantity obtained by evaluating the Gaussian with mean $\rho_m + v \cdot \Delta_t$ and standard deviation σ_v at ρ.

The transition probabilities for the resting places are computed based on a statistics about the average time period that elapses until the person starts to move on a particular trajectory after arriving at the corresponding resting place.

4 Person Detection and Identification

To keep track of multiple persons in an environment, one in principle would have to maintain a belief over the joint state space of all persons. This approach, however, is usually not feasible since the complexity of the state estimation problem grows exponentially with the number of persons or dimensions of the state space. Additionally, learning the joint transition probability distribution would require a huge amount of training data. Therefore we approximate the posterior by factorizing the belief over the joint state space and consider independent beliefs over the states of all persons. Thus, we use an individual HMM for each person. To maintain the individual beliefs we need to be able to update the HMMs for the persons based on observations made by the robot, which requires the ability to reliably detect persons and to identify them. To achieve this, our current systems combines laser and vision information.

To detect persons in the laser-range scans obtained with the robot our system extracts features which are local minima that correspond to the persons' legs as explained above. We also need to be able to identify a person in order to appropriately update the belief about the location of that person. To achieve this we employ the vision system of our robot and learn an image database beforehand. For each person this database contains one histogram which is built from 20 images. To identify a person, we proceed as follows. Every time the laser-based people detection system reports a feature in the field of view of the camera, an image is collected and the following three steps are applied:

1. *Segmentation:* A rectangular area containing the person is extracted from the image. To determine the area in the image corresponding to a feature detected

by the laser-based people detection system, we rely on an accurate calibration between the camera and the laser. We use a perspective projection to map the 3D position of the person in world coordinates to 2D image coordinates.

2. *Color histograms:* We compute a color histogram for the area selected in the previous step. Whereas color histograms are robust with respect to translation, rotation, scale and to any kind of geometric distortions they are sensitive to varying lighting conditions. To handle this problem we consider the HSV (Hue-Saturation-Value) color space. In this color model the intensity factor can be separated so that its influence is reduced. In our current system we simply ignore this factor. Throughout all our experiments we could not find any evidence that this negatively affected the performance of the system.

3. *Database matching:* To determine the likelihood that the area extracted in the segmentation step contains a particular person, we compare the histogram computed in step 2 to all prototypes existing in the database. As a measure of similarity between a query histogram q with a prototype π in the database we use the normalized intersection norm $H(q, \pi)$ [65]. This quantity can be computed as:

$$H(q, \pi) \quad = \quad \frac{\sum_{b=1}^{B} \min(q_b, \pi_b)}{\sum_{b=1}^{B} \pi_b}, \tag{8}$$

where q and π are color histograms both having B bins. One advantage of this norm is that it also allows to compare partial views, i.e.,when the person is close to the camera and only a part of it is visible.

As an application example consider the situation depicted in the left image of Figure 3. In this particular situation two persons (person B and person C) were walking along the corridor within the perceptual field of the robot. The right image of Figure 3 shows the estimate of the laser-based people detection system at the same point in time. The corresponding image obtained with the robot's camera is shown in the left image of Figure 4. The two segments of the image that correspond to the two features detected by the laser-based people detection system are also shown in this image. Figure 5 depicts the resulting color histograms of the two extracted segments. The right image of Figure 4 plots the similarities between these histograms and the three prototypes stored in the data base.

Since we consider independent beliefs over the states of the persons we have to determine which feature is caused by which person and we have to update each HMM depending on the likelihood that the corresponding person has been observed. For that purpose we apply Joint Probabilistic Data Association Filters (JPDAFs) [5].

Let $\xi^t = \{\xi_1^t, \ldots, \xi_R^t\}$ denote the state of the R persons we are tracking at time t. Each ξ_r^t is a random variable ranging over the state space of a single person. A measurement at time t is denoted as $z^t = \{z_1^t, \ldots, z_{S^t}^t\}$. Here S^t is the number of features detected at time t. In our case, each z_s^t is the position of a feature provided by the laser-based people detector together with the corresponding similarity values provided by the vision system. If the detected feature is not in the field of view of the

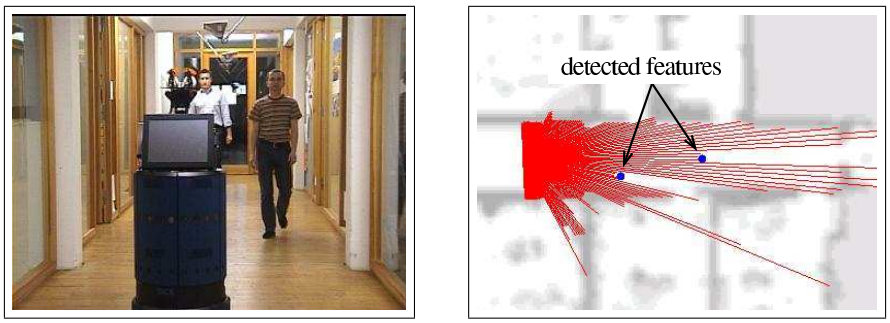

Figure 3: Typical scene with two persons walking along the corridor (left image) and corresponding estimate of the laser-based people detection system (right image).

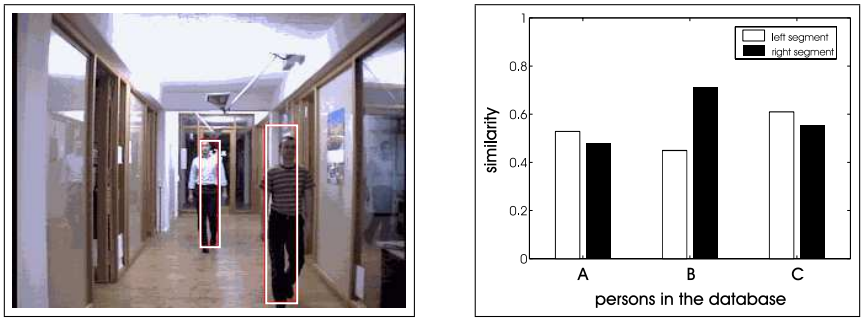

Figure 4: Segmentation of the two persons from the image grabbed with the camera of the robot (left image) and the similarity between the color histograms (see Figure 5) of the extracted segments and the data base prototypes (right image).

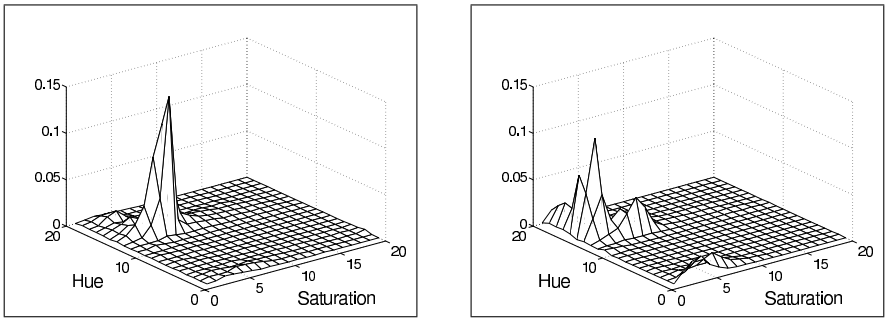

Figure 5: Corresponding color histograms for the left and right segment shown in Figure 4 (left image).

camera we assume that the similarity values are uniformly distributed over all data base images.

Whenever new sensory input arrives we follow the idea of Joint Probabilistic Data Association Filters and integrate the single features according to the assignment probability λ_{sr} that feature s corresponds to person r [58]:

$$P(\xi_r^t \mid z^{(1:t)}) \quad = \quad \eta \sum_{s=0}^{S^t} \lambda_{sr} P(z_s^t \mid \xi_r^t)\, P(\xi_r^t \mid z^{(1:t-1)}). \tag{9}$$

Here η is a normalization factor and $z^{(1:t)}$ denotes the sequence of all measurements up to time t.

To compute λ_{sr} one considers so-called joint association events. Each such event ψ is a set of pairs $(s,r) \in \{0,\ldots,S^t\} \times \{1,\ldots,R\}$ that uniquely determines which feature is assigned to which person. Note that in the JPDAF framework, z_0^t is used to model situations in which a person r has not been detected, i.e., no feature has been obtained for r. This is represented in the association event as $(0,r)$. We additionally have to consider false alarms which occur if a feature is not caused by any of the persons that are tracked using the HMMs. Let γ denote the probability that an observed feature is a false alarm. The number of false alarms contained in an association event ψ is given by $\phi = S^t - A$ where $A = R - \|\{(0,\cdot) \in \psi\}\|$ is the number of persons to which features have been assigned. Thus, γ^ϕ is the probability assigned to all false alarms in z^t given ψ. Finally, let Ψ_{sr} denote the set of all valid joint association events which assign feature s to person r. At time t the JPDAF computes the probability that feature s is caused by person r according to:

$$\lambda_{sr} = \sum_{\psi \in \Psi_{sr}} \left[\eta'\, \gamma^\phi \prod_{(j,i) \in \psi} \int P(z_j^t \mid \xi_i^t) P(\xi_i^t \mid z^{(1:t-1)}) d\xi_i^t \right]. \tag{10}$$

Again, η' is a normalizer. Since we use HMMs to represent the belief about the states of the persons the integration in (10) corresponds to summing over all states of the HMM for the particular person.

It remains to describe how the term $P(z_j^t \mid \xi_i^t)$ is computed. As explained above, each z_j^t consists of the position y_j^t of the feature j at time t and the similarity measure $H(q_j^t, \pi_i)$ between the query histogram q_j^t of the corresponding segment in the camera image and the database histogram of person i. In our current system we use the following approximation to compute the likelihood $P(y_j^t, H(q_j^t, \pi_i) \mid \xi_i^t)$, which has turned out to yield satisfactory results in practice:

$$P(z_j^t \mid \xi_i^t) = P(y_j^t, H(q_j^t, \pi_i) \mid \xi_i^t) = H(q_j^t, \pi_i)\; P(y_j^t \mid \xi_i^t). \tag{11}$$

Here $P(y_j^t \mid \xi_i^t)$ is the probability that the laser-based people detection system reports a feature detection at location y_j^t given that the person is in state ξ_i^t. We compute this quantity using a mixture of a uniform distribution and a bounded Gaussian with mean y_j^t. Note that we also take into account visibility constraints, i.e., states that are occluded are regarded as states outside the bounded Gaussian. In the case that no feature

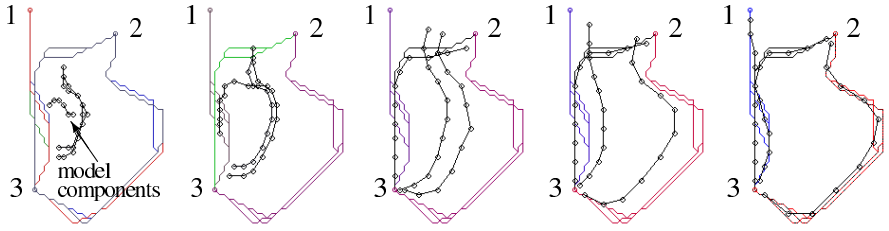

Figure 6: Trajectories of three different motion patterns and evolution of the model components during different iterations of the EM algorithm. The means of the three model components are indicated by circles and the numbers indicate the three resting places.

has been obtained for a person we use the likelihood of false negative observations for such states that are within the range of the robot's sensors. For all other states we use the average likelihood that the robot does not detect a person given it is outside the robot's sensor range.

5 Experimental Results

The experiments described in this section are designed to illustrate that our algorithm can learn complex motion patterns of persons in different types of environments. We also demonstrate that the HMMs derived from the learned motion patterns allow a robust estimation of the positions of multiple persons and that the behavior of a mobile robot can be improved by predicting the motions of persons based on the learned motion patterns.

5.1 Learning Results

To analyze the ability of our approach to learn different motion patterns from a set of trajectories we performed extensive experiments in different environments. This included a domestic residence, an office environment, and a large hallway. The following section describes an experiment using data collected in the home environment. In this experiment the actual number of motion patterns is known beforehand. In the second set of experiments, in which we use the data obtained in the office environment, the number of motion patterns is unknown and has to be determined during the clustering process.

To get the random initial model we initialize the expectations with a unimodal distribution for each trajectory, i.e., for each d_i the expectations $E[c_{i1} \mid \theta^{[0]}, d], \ldots, E[c_{iM^{[0]}} \mid \theta^{[0]}, d]$ form a distribution with a unique randomly chosen peak. In all experiments we set the parameter β to 5 which means that the mean of each probability distribution is computed based on 5 subsequent positions on the trajectories. The standard deviation

σ was set to 170 cm. We experimentally found out that these values yield good results.

5.1.1 Known Number of Motion Patterns

To see how our EM-based learning procedure works in practice consider Figure 6 (see also Extension 1). In this example, a model for nine trajectories belonging to three different motion patterns has to be learned. There are three trajectories leading from resting place 3 to resting place 1, three trajectories leading from 3 to 2, and three trajectories leading from 2 to 3.

The leftmost image shows the initial model (the means of the three model components are indicated by circles). In the following images one can see the evolution of the model components during different iterations of the EM algorithm. Finally, the rightmost image shows the model components after convergence of the EM algorithm. As can be seen, the trajectories have been approximated quite well by the individual model components.

5.1.2 Unknown Number of Motion Patterns

In the remaining experiments the task was to correctly learn the motion patterns of the persons along with their number. In principle, one could start our algorithm with a single model component and just introduce in each iteration (after convergence of the EM) a new model component for the trajectory which has the lowest likelihood given the current model. When the overall evaluation cannot be improved anymore by increasing the number of components, the system automatically alternates decreasing and increasing operations until the evaluation of the best model cannot be improved any more. However, to speed up the process we usually start our algorithm with a model that contains one component for six trajectories in the input data. This reduces the learning time since typically fewer increasing operations are needed. In general, it is not easy to guess a good initial value of the number of motion patterns. Even if there is a heuristic about the correct number of motion patterns, initializing our algorithm with this number does not automatically lead to a small number of iterations. This is because the EM often gets stuck in local maxima which means that there exist redundant model components that basically represent the same motion pattern. Those redundant model components are first eliminated before new model components will be introduced for trajectories with low likelihood.

We applied our algorithm to data recorded in our office environment (see map in Figure 8). From the collected data we extracted 129 trajectories. Figure 7 shows the model complexity and model evaluation for one run in which we started with 20 different model components. As can bee seen from the figure, the algorithm decreased the number of model complexity until only 17 (non-redundant) components remained. This is because, the model contained components with very low utilities. Afterwards it increased the number of model components to improve the model evaluation. Finally, it terminated with the model correctly representing all 49 different motion patterns. The trajectories of the learned model can be seen in Figure 8. The identified resting places are indicated by numbers.

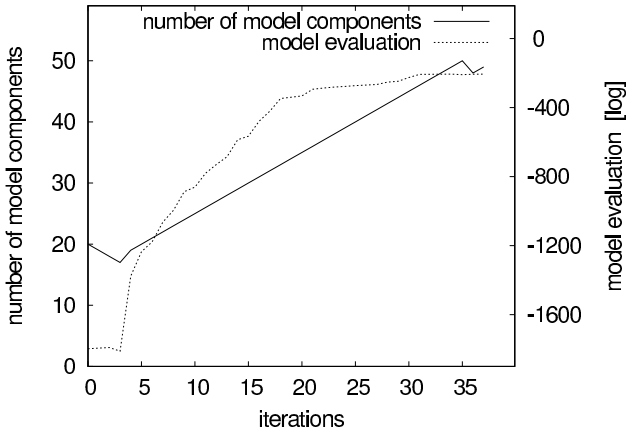

Figure 7: Evolution of the number of model components and the overall evaluation of the model during the application of our algorithm. In this case a model for 129 trajectories collected in our office environment (see map in Figure 8) has to be learned.

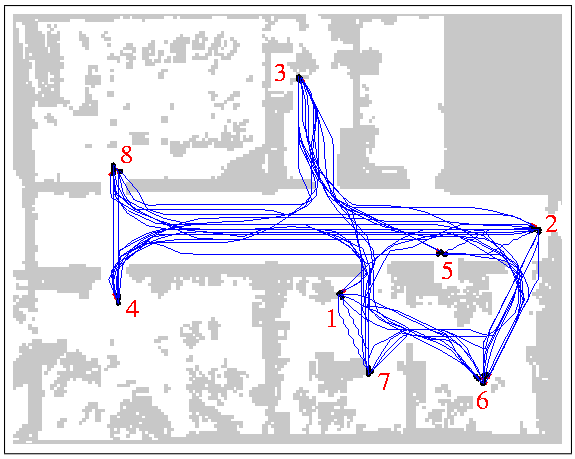

Figure 8: Trajectories of the 49 learned motion patterns in the office environment as well as the identified resting places.

To evaluate the performance of our approach we carried out a series of experiments using several data sets. In each experiment we chose a random set of trajectories and counted the number of correct classifications. It turned out that our algorithm was able to learn the correct model in 96% of all cases. We furthermore did not discover evidence that the number of model components we initialized our algorithm with has an influence on the overall result.

5.1.3 Discussion

Note that the output of our learning algorithm depends on the standard deviation σ and on the number of probability distributions K used to represent the motion patterns. It is clear that smaller values of σ will result in a higher number of model components. Furthermore, if there is only a relatively small number of trajectories for one motion pattern compared to the other motion patterns, our algorithm tends to underestimate the number of model components and to assign these trajectories to other clusters. Due to the assumption that all model components have the same length, our algorithm prefers to cluster longer trajectories into single components rather than short trajectories. This is because the distance between long trajectories and their cluster is typically higher than for short trajectories. A solution to this problem would be to additionally estimate the number of probability distributions constituting each particular motion pattern. This aspect will be subject of future research.

Furthermore, there are several alternative clustering approaches that might be applicable to our problem of learning motion patterns. For the purpose of comparison we also implemented the classical k-Means algorithm. This algorithm differs from the approach used in this paper in that the correspondence variables have binary values only and that each data trajectory d_i is assigned to the "closest" model component, i.e., the model component θ_m which has the highest probability that d_i is observed given the person is engaged in θ_m. In our experiments it turned out that the classical k-Means algorithm yields a similar success rate compared to our approach. The advantage of our technique lies in the probabilistic framework and its correspondence to the EM algorithm as a general optimization procedure.

5.2 Tracking Persons

The experiments described in this section are designed to illustrate that our approach can be used to robustly estimate the positions of multiple persons, even if they are not in the field of view of the robot. Finally, we present experiments illustrating that the belief about current and future positions of persons can be used by a mobile robot to adapt its navigation behavior. To analyze the applicability of the HMMs for the prediction of the locations of a person we performed several experiments with our B21r robot Albert in our office environment. The Hidden Markov Model used to carry out these experiments was computed based on data recorded over two hours in our office environment. During the acquisition phase the average speed of the person was v=107 cm/sec with a standard deviation of σ_v=25 cm/sec. The possible transitions of the Hidden Markov Model that was derived from the learned motion patterns in the office environment is shown in Figure 9. Whereas the numbered squares indicate the eight resting places, the small circles on the trajectories are the intermediate states.

5.2.1 Tracking a Single Person

The first experiment is designed to illustrate that our approach is able to reliably estimate the position of a person in its environment. In this experiment, a single person was moving in our department and the task of the robot, which itself did not move, was to

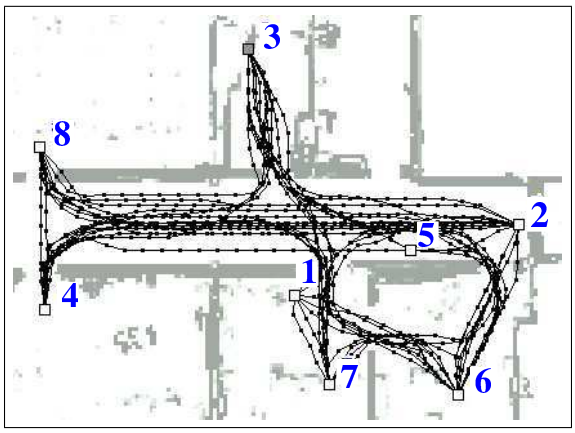

Figure 9: Possible Transitions of the Hidden Markov Model derived from learned motion patterns. The numbered squares indicate the eight resting places. The small circles on the trajectories correspond to the intermediate states.

estimate the positions of this person. Especially, we were interested in the probability that the person stayed at the correct resting place.

Figure 10 shows a scene overview (left image) for an experiment in which the person walked through the environment. The robot could only cover a part of the environment with its sensors but even though it was able to maintain and update the belief about the position of the person at any point in time. The right image of Figure 10 depicts the results of the laser-based feature detection system. The left image of Figure 11 shows the corresponding beliefs about the position of the person after integrating the observation. In this case we did not use vision information since there was only one person in the environment of the robot. The grey dot corresponds to the position of the feature provided by the laser-based people detector. The size of the squares of the states of the HMM represents the probability that the person is currently in the corresponding state. Additionally, the resting places are labeled with the probability that the person stays at this particular place. When the robot observed the person as it walked through the corridor it assigned high likelihood to the states close to the detected feature. After the person entered the room and moved outside the field of view of the robot most of the probability mass "wandered" to resting place 7 (see right image of Figure 11) which was according to the transition probabilities encoded in the HMM.

Figure 12 plots for different resting places the probability that the person stayed at this particular place over time. The dashed-dotted, dashed, and solid curve correspond to resting places 3, 7, and 6 respectively. Whereas the x-axis represents the individual time steps, the y-axis indicates the probability. The graph also includes the ground truth, which is indicated by the corresponding horizontal line-pattern at the .9 level. As can be seen from the figure, the system can reliably determine the location of the person. During this experiment the robot predicted the correct position of the person in 93% of the time.

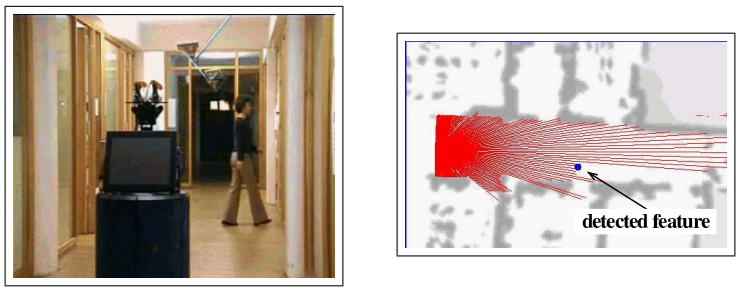

Figure 10: Robot Albert tracking a person while it is moving through the environment. The left image shows a scene overview and the right images depicts the corresponding results of the laser-based feature detection system.

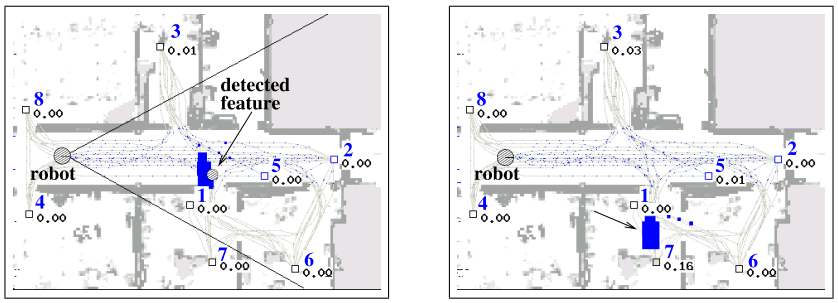

Figure 11: The left image shows the corresponding belief over the position of the person after integrating the observation shown in Figure 10. The size of the squares represents the probability that the person is currently in the corresponding state and the resting places are labeled with the probability that the person stays currently there. Even if the robot does not observe the person any more it is able correctly infer that the person is going to resting place 7 (right image).

Figure 13 shows another experiment with a moving robot (see also Extension 2). Here the robot traveled along the corridor and looked into one of the offices where it detected person A (see left image of Figure 13). The evolution of the probabilities for person A to be at resting places 3, 6, and 7 is plotted in the right image of this figure. Whereas the robot initially was rather uncertain as to where person A was, the probability of resting place 3 seriously increased after the detection.

5.2.2 Estimating the Locations of Multiple Persons

As an application example with multiple persons consider the situation depicted Figure 14 (see also Extension 3). In this experiment the database of the robot consisted of three persons. For all three persons we used identical motion patterns and transition probabilities in the HMMs. In the situation described here the robot was initially quite

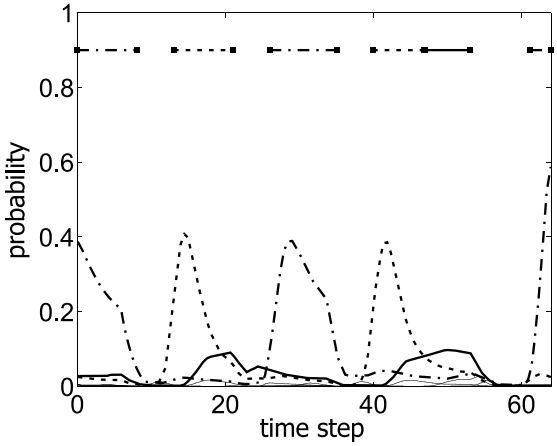

Figure 12: Evolution of the probability of the person to be at the different resting places over the time. The ground truth is indicated by the horizontal line-pattern at the .9 level.

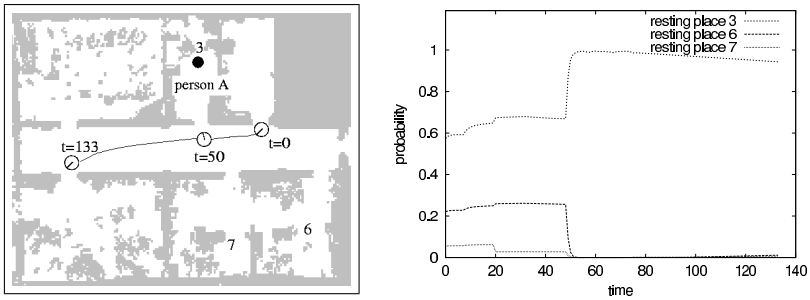

Figure 13: Trajectory of the robot including the detection of person A in his office (left image) as well as the corresponding evolution of the probability that person A is at the resting places 3, 6, or 7 (right image).

certain that persons A and B were in the room containing resting place 3. Then the robot observed a person leaving the room (see images in the first row of Figure 14). The grey circles labeled with names indicate the position provided by the laser-based feature detection system. Since the robot did not get vision information at this particular time step it was uncertain who the person was. Note that we use uniform similarity values for all database images in such situations. The intensity of the circle represents the similarity between the extracted segment and the database images of the person corresponding to the HMM (the darker the more likely). Again, the size of the squares represents the probability that the person is currently in the corresponding state and the probabilities of the resting places are indicated by numbers. In the images shown in the second row a second person entered the corridor. Now the robot received vision

information and updated the individual HMMs according to the data association probabilities computed by the JPDAF (see images in the third row). During the next time steps both persons left the field of view of the robot but nevertheless the robot was able to maintain an adequate belief about their positions.

5.3 Improving the Navigation Behavior of a Mobile Robot

In the experiments described above we demonstrated that our approach can be used to learn motion patterns of persons and how to use such information to maintain a belief about the positions of the persons. The following experiment is designed to show that a robot, which takes into account different motion patterns during path planning, performs significantly better than a robot that just relies on a linear prediction of the movements of persons.

In particular, we use the learned motion patterns to predict the movements of detected persons and consider the forecasted trajectories during path planning. In our current system we apply the A^* algorithm [46] to determine the minimum-cost path in the three dimensional configuration time-space of the robot. The environment is represented using an occupancy grid map [43]. Each cell $\langle x, y \rangle$ of this grid stores the probability $P(occ_{x,y})$ that the corresponding area in the environment is occupied. The cost for traversing a cell $\langle x, y \rangle$ is proportional to its occupancy probability $P(occ_{x,y})$. To incorporate the robot's belief about future movements of detected persons, we additionally discount a cell $\langle x, y \rangle$ according to the probability, that one of the persons covers $\langle x, y \rangle$ at a given time t.

Consider the situation depicted in the left image of Figure 15. The robot was moving along the corridor from left to right to reach its target location, which is labeled C in the figure. At the position labeled with A the robot detected a person approaching it. According to the learned motion patterns the person was most probably walking to resting place 3. The motion patterns which had a sufficiently high probability are depicted in Figure 16 (the thicker the trajectory the higher the probability that the person will follow this specific trajectory). Since the probabilities of the motion patterns with target locations 4 and 8 were very low, the additional costs introduced to the configuration time-space did not prevent the robot from driving further along the corridor. Thus, the robot moved to the location labeled B in the left image of Figure 15 and waited there until the person entered the room in the north. After that it moved to its target location, which is labeled C.

We repeated this experiment with a system that does not use the motion patterns and instead only linearly predicts the trajectory of persons. The trajectory of the robot in this experiment is shown in the right image of Figure 15. After the robot detected the person it continued to move and simultaneously replanned its path using the configuration time space computed based on a linear prediction of the movements of the person. When it detected that it would block the path of the person it turned around and moved to the location labeled B in the right image of Figure 15. After it noticed that the person disappeared the robot continued to its designated target location C. We also performed this experiment with a collision avoidance system that does not predict the motions of persons [63]. In a situation like the one considered here, the person always had to slow down because the robot was blocking its path.

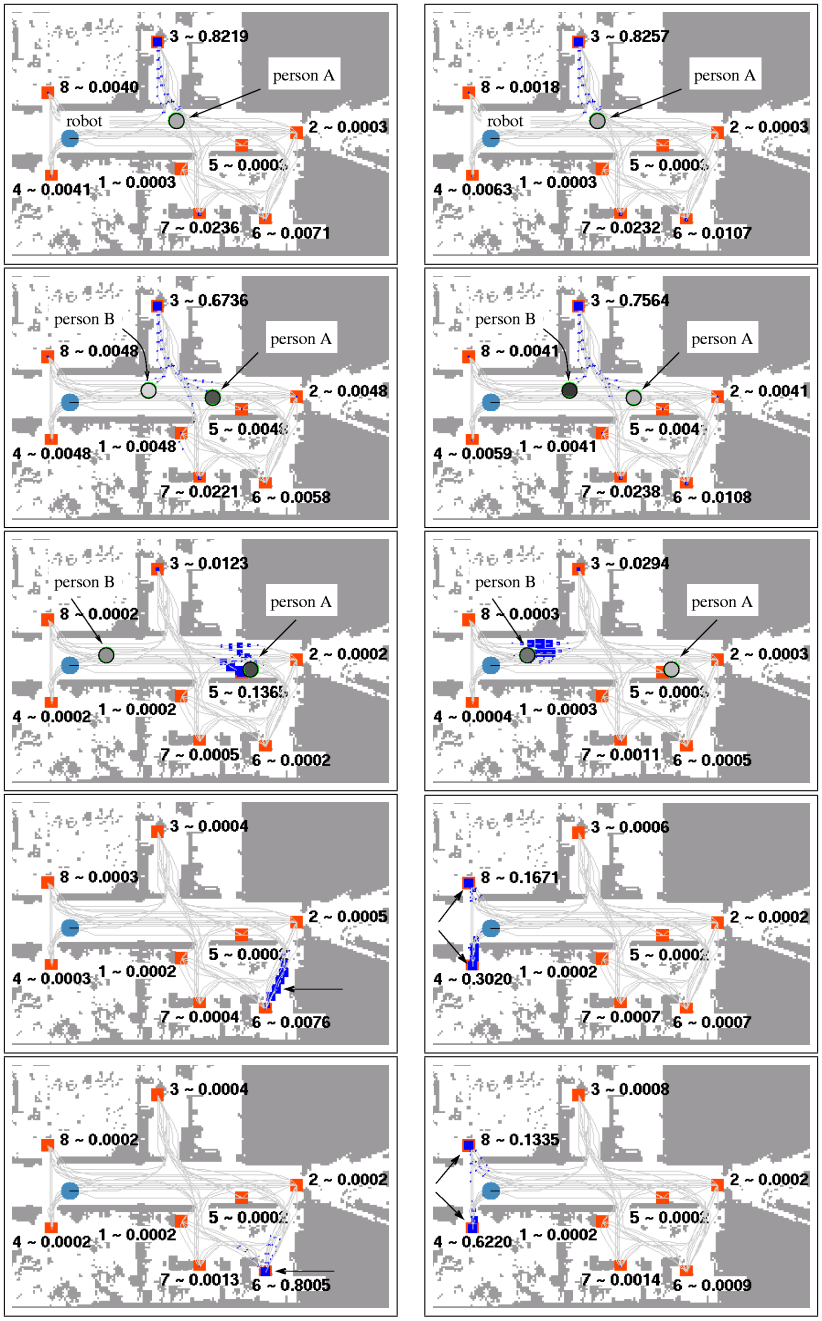

Figure 14: This figure shows an experiment with two persons. Whereas the left column depicts the belief about the position of person A the right column shows the belief about the position of person B. The circles labeled with names are detected features. The grey value of each circle represents the similarity to the person corresponding to the HMM.

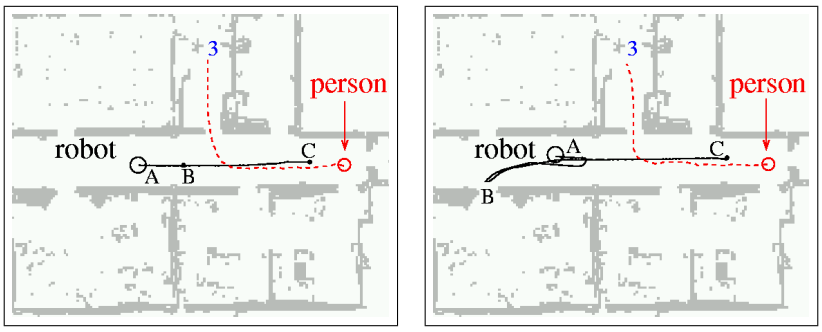

Figure 15: In the left image the robot used the motion patterns to predict future poses of persons. At the position labeled A the robot observed a person approaching it. Based on the learned motion patterns it inferred that the person will enter the room to the north to go to resting place 3 with very high likelihood (see Figure 16). Accordingly, the robot moved forward and waited at the position labeled B until the person left the corridor. In the right image the the robot used a linear prediction of the trajectory of the person. It anticipated that it will block the person's way and thus it moved to the position labeled B. After the person left the corridor the robot continued approaching its target location.

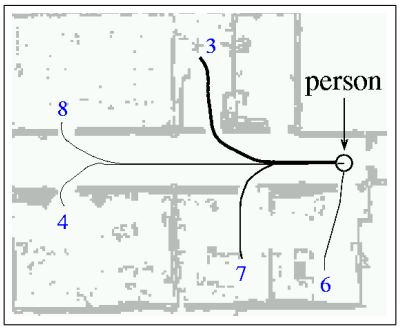

Figure 16: Most probable trajectories of the person in the experiment shown in Figure 15. The thicker the trajectory the higher the probability that the person follows this motion pattern.

We performed ten similar experiments for each of the prediction strategies and measured the time needed to complete the navigation task. The average time for both systems is shown in Figure 17. As can be seen from the graph the time can be significantly reduced when taking into account learned motion patterns compared to the approach that only performs a linear prediction.

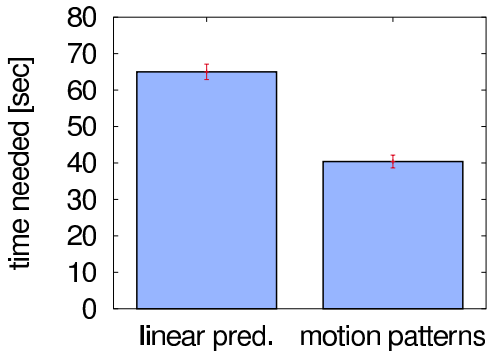

Figure 17: Average time needed to complete the navigation task when the robot performs linear prediction (left column) and when it uses the learned motion patterns (right column) for ten experiments.

6 Related Work

A variety of techniques has been developed that allows a robot to estimate the positions of people in its vicinity or to predict future poses. The approaches presented by Tadokoro et al. [66] and Zhu [69] use given probabilistic motion models in order to predict future poses of observed objects. Since these approaches do not learn typical motion patterns, they can only predict short-term motions. The techniques developed by Schulz et al. [58] and Montemerlo et al. [42] are able to robustly track multiple persons in the sensor range of a mobile robot equipped with laser-range finders using particle filters. Kluge et al. [29] also implemented a system to track people but they do not apply a motion model to the objects. Thus, they cannot reliably keep track of individual moving objects over time and deal with temporary occlusion. The same drawback has the approach presented by Lindström and Eklundh [36] which uses Gaussian hypotheses to detect and track moving objects from a moving platform in furnished rooms. Rosencrantz et al. [55] introduced variable-dimension particle filters to track the location of moving objects even if they are temporarily occluded. However, they are not able to estimate the probability of certain positions when the objects are outside the sensor range. They can only predict that the persons are in specific areas (rooms). There are also many vision-based techniques that are designed to keep track of moving objects and which show robust behavior even in the case of temporary occlusions (for example [53, 37]). Feyrer and Zell [10] use a combination of vision and laser data to detect persons in front of stationary background and to track them. Furthermore, they use an artificial potential field to pursue them in real-time thereby avoiding collisions with obstacles. Fod et al. [11] use multiple statically mounted laser-range finders and apply Kalman filters to maintain an estimate of the positions of people in everyday environments whereas Krumm et al. [30] deployed multiple stationary cameras for multi-person tracking in a living room. Lavalle et al. [34] presented a system that is

able to follow of a moving target. They explicitely take into account possible short-term occlusions by static obstacles in the environment. González-Baños et al. [17] presented a modified version that works without a prior map of the environment. Furthermore, some multi-robot systems have been developed that keep track of multiple moving targets [44, 25, 49] or surround a moving target [50]. Riley and Veloso [52] use predefined opponent models in the RoboCup domain to reason about the opponents' strategy and predict their behavior. Depending on the inferred strategy they adapt their own team play. Foka and Trahanias [12] suggested to predict the movements of people using manually defined "hot points" which the people might be interesting in approaching. Ehrenmann et al. [7] proposed the interpretation of dynamic gestures to command robots and Kasper et al. [26] presented an approach to improve the behavior of a robot by following the activities of a teacher.

The majority of the existing position tracking techniques assume that the models of the motion behavior of the objects to be tracked are given. Our approach, in contrast, is able to learn such models and to use the learned models for the long-term prediction of the persons' movements. Our system is able to maintain an estimate about the positions of multiple persons even if they are not in the sensor range of the robot for a long period of time.

The system described by Kruse and Wahl [31] uses a camera system mounted on the ceiling to track persons in the environment and to learn where the people usually walk in their workspace. A collision probability field similar to a potential field [3] is computed which incorporates the average motion behavior of the persons. Accordingly, their system can only make a short-term prediction of future poses of persons. Johnson and Hogg [24] learn probability density functions of typical object trajectories to detect atypical behaviors. Compared to the work presented here, they do not apply a technique to estimate the number of different motion patterns. The goal of the work by Stauffer and Grimson [64] is also to detect unusual events. They learn codebooks of a given number of prototypes. Rosales and Sclaroff [54] analyze 3D trajectories to learn typical classes of actions like walking, running, and biking. Oliver et al. [48] use data obtained from various sensors as input to an Layered HMM and infer the state of a user's activity. Galata et al. [14] use Variable Length Markov Models (VLMMs) to model structured behaviors. One problem to be solved in the context of VLMMs is the estimation of the optimal size of the time window in order to correctly predict the next states. We follow a conservative approach and assume all steps in the past to be relevant. Nguyen et al. [45] recently proposed to use an Abstract Hidden Markov mEmory Model (AHMEM) to infer intentions of persons. The idea of an AHMEM is to model higher level behaviors by a stochastic sequence of more simple behaviors at the lower levels. The authors apply an EM-based learning method for labeled trajectories to determine the transition probabilities for the states at the lowest level (grid cells) and assume that the landmarks the persons want to approach are given. Our approach in contrast applies an unsupervised clustering method to the observed trajectories and is also able to automatically infer resting places which correspond to the landmarks in the AHMEM. Illmann et al. [22] apply a fusion of omnidirectional vision and information of a laser-range finder to extract basic motion patterns like straight motion, wandering aimlessly, or entering a queue. Their goal is to predict short-term motions of surrounding people using the learned motion patterns so that a mobile robot can chose

adequate behaviors. Finally, Guralnik and Haigh [19] use sequential pattern mining to learn typical behaviors of humans in their homes. They installed 10-20 sensors of different types in a home. Their algorithm uses this data to learn sequences of rooms in which room the person was acting. Their algorithm uses domain knowledge to extract sequences of rooms the person was acting in. These sequences are then analyzed by a human expert to identify complex behavior models. The approach described here can be seen as an alternative way of learning the sequences of rooms that does not require domain knowledge.

7 Conclusions

In this article we presented a method for learning and utilizing motion patterns of persons. Our approach applies the EM algorithm to cluster trajectories recorded with laser-range sensors into a collection of motion patterns. We furthermore introduced a method for automatically deriving an HMM from these typical motion patterns of persons. To update the resulting HMMs based on laser-range data and vision information we apply Joint Probabilistic Data Association Filters. We also described how to incorporate predicted trajectories of persons into the path planning process.

Our approach has been implemented and successfully applied to trajectories recorded in different environments. Practical experiments carried out with a mobile robot in different environments demonstrate that our method is able to learn typical motion patterns of persons and to reliably use them for maintaining a probabilistic belief about the current positions of persons. We furthermore presented experiments illustrating that the behavior of a mobile robot can be improved by predicting the motions of persons based on learned motion patterns.

8 Acknowledgments

This work has partly been supported by the IST Programme of the European Communities under contract numbers IST-2000-29456 and HPMT-CT-2001-00251, by the German Science Foundation (contract number SFB-TR 08), and by the National Science Foundation under the ITR Program. We furthermore would like to thank Cyrill Stachniss and Julio Pastrana for helping us to carry out the experiments with our mobile robot.

References

[1] 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.

[2] H. Asoh, S. Hayamizu, I. Hara, Y. Motomura, S. Akaho, and T. Matsui. Socially embedded learning of office-conversant robot Jijo-2. In *Proc. of the International Joint Conference on Artificial Intelligence (IJCAI)*, 1997.

[3] J. Barraquand, B. Langois, and J.C. Latombe. Numerical potential field techniques for robot path planning. *IEEE Transactions on Robotics and Automation, Man and Cybernetics*, 22(2):224–241, 1992.

[4] 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.

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

[6] R. Duda, P. Hart, and D. Stork. *Pattern Classification*. Wiley-Interscience, 2001.

[7] M. Ehrenmann, T. Lütticke, and R. Dillmann. Dynamic gestures as an input device for directing a mobile platform. In *Proc. of the IEEE International Conference on Robotics & Automation (ICRA)*, 2001.

[8] 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.

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

[10] S. Feyrer and A. Zell. Robust real-time pursuit of persons with a mobile robot using multisensor fusion. In *6th International Conference on Intelligent Autonomous Systems (IAS-6)*, 2000.

[11] A. Fod, A. Howard, and M.J. Matarić. A laser-based people tracker. In *ICRA*, 2002.

[12] A.F. Foka and P.E. Trahanias. Predictive autonomous robot navigation. In *Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 490–495, 2002.

[13] E. Forgy. Cluster analysis of multivariate data: Efficiency vs. interpretability of classifications. *Biometrics*, 21(3), 1965.

[14] N. Galata, A.and Johnson and D. Hogg. Learning variable length markov models of behaviour. *Computer Vision and Image Understanding (CVIU) Journal*, 81(3), 2001.

[15] K. Goldberg and R. Siegwart. *Robots on the Web: Physical Interaction through the Internet*. MIT-Press, 2001.

[16] R.P. Goldman, W. Geib, and C.A. Miller. A new model of plan recognition. In *Proc. of the Conference on Uncertainty in Artificial Intelligence (UAI)*, 1999.

[17] 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.

[18] H.-M. Gross and H.-J. Boehme. Perses – a vision-based interactive mobile shopping assistant. In *Proc. of the IEEE International Conference on Systems, Man and Cybernetics*, 2000.

[19] V. Guralnik and K.Z. Haigh. Learning models of human behaviour with sequential patterns. In *Proc. of the AAAI-02 workshop "Automation as Caregiver"*, 2002.

[20] G. Hirzinger, B. Brunner, J. Dietrich, and J. Heindl. ROTEX the first remotely controlled robot in space. In *Proc. of the IEEE International Conference on Robotics & Automation (ICRA)*, 1994.

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

[22] 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.

[23] M. Jäger and B. Nebel. Dynamic decentralized area partitioning for cooperating cleaning robots. In *Proc. of the IEEE International Conference on Robotics & Automation (ICRA)*, 2002.

[24] N. Johnson and D. Hogg. Learning the distribution of object trajectories for event recognition. In *British Machine Vision Conference*, 1995.

[25] 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.

[26] M. Kasper, G. Fricke, K. Stuernagel, and E. von Puttkamer. A behavior-based mobile robot architecture for learning from demonstration. *Journal of Robotics and Automation*, 34(2-3):153–164, 2001.

[27] H. Kautz and J.F. Allen. Generalized plan recognition. In *Proc. of the National Conference on Artificial Intelligence (AAAI)*, 1986.

[28] S. King and C. Weiman. Helpmate autonomous mobile robot navigation system. In *Proc. of the SPIE Conference on Mobile Robots*, pages 190–198, Boston, MA, November 1990. Volume 2352.

[29] B. Kluge, C. Köhler, 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.

[30] J. Krumm, S. Harris, B. Meyers, B. Brumitt, M. Hale, and S. Shafer. Multi-camera multi-person tracking for Easyliving. In *Proc. of Third IEEE International Workshop on Visual Surveillance*, 2000.

[31] F. Kruse, E. und Wahl. Camera-based monitoring system for mobile robot guidance. In *Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1248–1253, 1998.

[32] G. Lacey and K. Dawson-Howe. The application of robotics to a mobility aid for the elderly blind. *Journal of Robotics and Autonomous Systems (RAS)*, 23:245–252, 1998.

[33] A. Lankenau and T. Röfer. Smart wheelchairs – state of the art in an emerging market. *Zeitschrift KI mit Schwerpunkt Autonome Mobile Systeme*, 4, 2000.

[34] 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.

[35] C. Li, G. Biswas, M. Dale, and P. Dale. Building models of ecological dynamics using hmm based temporal data clustering. In *Proc. of the Fourth International Conference on Intelligent Data Analysis*, 2001.

[36] 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.

[37] 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.

[38] J. MacQueen. Some methods for classification and analysis of mulivariate observations. In *Proc. of the Fifth Berkeley Symposium on Math., Stat. and Prob.*, pages 281–296, 1967.

[39] G.J. McLachlan and T. Krishnan. *The EM Algorithm and Extensions*. Wiley Series in Probability and Statistics, 1997.

[40] T.M. Mitchell. *Machine Learning*. McGraw-Hill, 1997.

[41] M. Montemerlo, J. Pineau, N. Roy, S. Thrun, and V. Verma. Experiences with a mobile robotic guide for the elderly. In *Proc. of the AAAI National Conference on Artificial Intelligence*, 2002.

[42] M. Montemerlo, S. Thrun, 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.

[43] H.P. Moravec and A.E. Elfes. High resolution maps from wide angle sonar. In *Proc. IEEE Int. Conf. Robotics and Automation*, pages 116–121, 1985.

[44] 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.

[45] N.T. Nguyen, H.H. Bui, S. Venkatesh, and G. West. Recognising and monitoring high-level behaviours in complex spatial environments. In *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2003.

[46] N. J. Nilsson. *Principles of Artificial Intelligence*. Springer Publisher, Berlin, New York, 1982.

[47] I. Nourbakhsh, R. Powers, and S. Birchfield. Dervish: An office-navigating robot. *AI Magazine*, 16(2), 1995.

[48] N. Oliver, E. Horvitz, and A. Garg. Layered representations for learning and inferring office activity from multiple sensory channels. In *Proc. of the International Conference on Multimodal Interfaces (ICMI)*, 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] L.R. Rabiner and B.H. Juang. An introduction to hidden markov models. *IEEE ASSP Magazine*, 3(1):4–16, 1986.

[52] P. Riley and M. Veloso. Recognizing probabilistic opponent movement models. In A. Birk, S. Coradeschi, and S. Tadokoro, editors, *RoboCup-2001: The Fifth RoboCup Competitions and Conferences*. Springer Verlag, Berlin, 2002.

[53] R. Rosales and S. Sclaroff. Improved tracking of multiple humans with trajectory prediction and occlusion modeling. In *Proc. of the IEEE Workshop on Interpretation of Visual Motion*, 1998.

[54] R. Rosales and S. Sclaroff. A framework for heading-guided recognition of human activity. *Computer Vision and Image Understanding (CVIU) Journal*, 2003. to appear.

[55] M. Rosencrantz, G. Gordon, and S. Thrun. Locating moving entities in dynamic indoor environments with teams of mobile robots. In *Proc. of the Second Joint International Conference on Autonomous Agents & Multi Agent Systems (AAMAS)*, 2003.

[56] C. Schaeffer and T. May. Care-o-bot - a system for assisting elderly or disabled persons in home environments. In *Assistive technology on the threshold of the new millenium*. IOS Press, Amsterdam, 1999.

[57] R.D. Schraft and G. Schmierer. *Service Robots*. Springer Verlag, 1998.

[58] D. Schulz, W. Burgard, D. Fox, and A.B. Cremers. People tracking with a mobile robot using sample-based joint probabilistic data association filters. *International Journal of Robotics Research (IJRR)*, 2003.

[59] D. Schulz, W. Burgard, D. Fox, S. Thrun, and A.B Cremers. Web interfaces for mobile robots in public places. *IEEE Robotics and Automation Magazine*, 7(1), 2000.

[60] G. Schwarz. Estimating the dimension of a model. *The Annals of Statistics*, 6(2):461–464, 1978.

[61] R. Siegwart, K.O. Arras, B. Jensen, R. Philippsen, and N. Tomatis. Design, implementation and exploitation of a new fully autonomous tour guide robot. In *In Proceedings of the 1st International Workshop on Advances in Service Robotics (ASER)*, 2003.

[62] 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*, 1997.

[63] C. Stachniss and W. Burgard. An integrated approach to goal-directed obstacle avoidance under dynamic constraints for dynamic environments. In *Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2002.

[64] C. Stauffer and W.E.L. Grimson. Learning patterns of activity using real-time tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):747–757, 2000.

[65] M. Swain and D. Ballard. Color indexing. *International Journal of Computer Vision*, 7(1), 1991.

[66] 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)*, pages 518–523, 1995.

[67] S. Thrun, M. Beetz, M. Bennewitz, W. Burgard, A. B. Cremers, D. Dellaert, D. Fox, D. Hähnel, C. Rosenberg, J. Schulte, and D. Schulz. Probabilistic algorithms and the interactive museum tour-guide robot Minerva. *International Journal of Robotics Research*, Vol. 19, Number 11:972–999, 2000.

[68] T. Weigel, J.-S. Gutmann, M. Dietl, A. Kleiner, and B. Nebel. CS Freiburg: Coordinating robots for successful soccer playing. *IEEE Transactions on Robotics and Automation*, 2002.

[69] Q. Zhu. Hidden Markov model for dynamic obstacle avoidance of mobile robot navigation. *IEEE Transactions on Robotics and Automation*, 7(3):390–397, 1991.

9 Index to Multimedia Extensions

The multimedia extensions to this article can be found online by following the hyperlinks from http://www.ijrr.org/.

Extension	Type	Description
1	Video	Application of EM: This video shows the evolution of the model components during an application of EM. (100 KB)
2	Video	Verifying a Person's Location: In this experiment the robot updates its belief about the position of a person while it is moving. (3.4 MB)
3	Video	Tracking the Positions of Multiple Persons: This video shows the evolution of the belief about the positions of two persons. (5.9 MB)