Using EM to Learn Motion Behaviors of Persons with Mobile Robots*

Maren Bennewitz† Wolfram Burgard† Sebastian Thrun‡

†*Department of Computer Science, University of Freiburg, 79110 Freiburg, Germany*
‡*School of Computer Science, Carnegie Mellon University, Pittsburgh PA, USA*

Abstract

We propose a method for learning models of people's motion behaviors in indoor environments. As people move through their environments, they do not move randomly. Instead, they often engage in typical motion patterns, related to specific locations that they might be interested in approaching and specific trajectories that they might follow in doing so. Knowledge about such patterns may enable a mobile robot to develop improved people following and obstacle avoidance skills. This paper proposes an algorithm that learns collections of typical trajectories that characterize a person's motion patterns. Data, recorded by mobile robots equipped with laser-range finders, is clustered into different types of motion using the popular expectation maximization algorithm, while simultaneously learning multiple motion patterns. Experimental results, obtained using data collected in a domestic residence and in an office building, illustrate that highly predictive models of human motion patterns can be learned.

1 Introduction

Whenever mobile robots are designed to operate in populated environments, they need to be able to perceive the people in their environment and to adapt their behavior according to the activities of the people. The knowledge of typical motion behaviors of the surrounding people can be used in several ways to improve the behavior. If a robot is able to predict the motions of a person it can, for example, choose appropriate detours that minimize the risk of collisions. Knowing where the person currently is or where it is currently going to is an important aspect in the context of a nursing robot project [12]. The goal of this project is to develop intelligent service robots than can assist people in their daily living activities.

Recently, a variety of service robots were developed that are designed to operate in populated environments. These robots, for example, are deployed in hospitals [7], museums [4], office buildings [1], and department stores [6], where they perform various services, e.g., deliver, educate, entertain [14] or assist people [13, 9]. Additionally, a variety of techniques has been developed that allows a robot to estimate the positions of people in its vicinity or to adapt its behavior accordingly. For example, the techniques described in [15] are designed to track multiple persons in the vicinity of a robot. The approach presented in [16] uses a given probabilistic model of typical motion behaviors in order to predict future poses of the persons. The system described in [8] uses a camera to estimate where persons typically walk and adapts the trajectory of the robot appropriately. [17] apply a Hidden-Markov-Model to predict the motions of moving obstacles in the environment of a robot. [10] present a system that is able to keep track of a moving target even in the case of possible occlusions by other obstacles in the environment. All the techniques described above assume the existence of a model of the motion behaviors. Our approach, in contrast, is able to learn such models and to use the learned models for the prediction of the peoples movements. The technique described in [3] uses an Abstract Hidden-Markov-Model to learn and to predict motions of a person. This approach assumes that all motions are already clustered into the corresponding motion behaviors during the learning phase. Our method extends this approach as it determines both, the clustering and the corresponding motion behaviors.

In this paper we present an approach that allows a mobile robot to learn motion patterns of persons, while they are carrying out their every-day activities. We use the popular EM-algorithm [11] to simultaneously cluster trajectories belonging to one motion behavior and to learn the characteristic motions of this behavior. We apply our technique to data recorded by mobile robots equipped with laser-range finders and demonstrate how the learned models can be used to predict the trajectory of a person in the natural environment. This paper extends out previous work described in [2] in different aspects. First, the method presented here is able to learn the number of motion patterns. Second, it includes a better approach to deal with trajectories of different length. Finally, it uses piecewise linear approximations to reduce the complexity of the learned models.

This paper is organized as follows. In the next section, we present the probabilistic representation of the motion

*This work has partly been supported by the EC under contract number IST-2000-29456.

patterns and describe how to learn them using the expectation maximization algorithm. In Section 3 we describe our application based on data recorded with laser-range finders. Section 4 presents experimental results regarding the learning process and the prediction accuracy of the learned models.

2 Learning Motion Patterns

Our approach to discovering typical motion patterns of people is strictly statistical, using the popular EM-algorithm to find different types of activities that involve physical motion throughout the natural environment. The input to our routine is a collection of trajectories $d = \{d_1, \ldots, d_N\}$ (called: the data). The output is a number of different types of motion patterns $\theta = \{\theta_1, \ldots, \theta_M\}$ a person might exhibit in their natural environment. Each trajectory d_i consists of a sequence $d_i = \{x_i^1, x_i^2, \ldots, x_i^{T_i}\}$ of positions x_i^t. Accordingly, x_i^1 is the first position covered by the person and $x_i^{T_i}$ is the final destination. Throughout this paper we assume that all trajectories have the same length T. In our current system we choose T as the maximum length of all trajectories. A trajectory d_i of length $T_i < T$ is extended by linear interpolation.

2.1 Motion Patterns

We begin with the description of our model of motion patterns, which is subsequently estimated from data using EM. Within this paper we assume that a person engages in M different types of motion patterns. A motion pattern, denoted θ_m with $1 \leq m \leq M$, is represented by K probability distributions $p(x \mid \theta_m^k)$. The mean of each probability distribution 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 he or she is engaged in motion pattern m. Thus, we calculate the likelihood of a trajectory d_i under the m-th motion model θ_m as

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

Please note that the approach described in [2] is a special instance of this scheme, as it corresponds to $\beta = 1$.

2.2 Expectation Maximization

In essence, our approach seeks to identify a model θ that maximizes the likelihood of the data. To define the likelihood of the data under the model θ, it will be useful to introduce a set of *correspondence variables*, denoted c_{im}. Here i is the index of a trajectory d_i, and m is the index of a motion model θ_m. Each correspondence 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 model as a specific motion activity a person might be engaged in, c_{im} is 1 if person was engaged in motion activity m in trajectory i.

In the sequel, we will denote the set of all correspondence variables for the i-th data item by c_i, that is, $c_i = \{c_{i1}, \ldots, c_{iM}\}$. For any data item i, the fact that exactly one correspondence is 1 translates to $\sum_{m=1}^{M} c_{im} = 1$.

Throughout this paper we assume that each motion pattern is represented by T Gaussian distributions with a fixed standard deviation σ. Accordingly, the application of EM leads to an extension of the fuzzy k-Means Algorithm (see e.g. [5]) to trajectories.

The goal is to find the set of motion patterns which has the highest data likelihood. How this likelihood is computed is explained in [2]. Since the logarithm is a monotonic function we can maximize the log likelihood instead of the likelihood which is given by:

$$\begin{aligned} \ln p(d, c \mid \theta) \quad = \quad & \sum_{i=1}^{N} \Bigg(T \cdot M \cdot \ln \frac{1}{\sqrt{2\pi}\sigma} \\ & - \frac{1}{2\sigma^2} \cdot \sum_{t=1}^{T} \sum_{m=1}^{M} c_{im} \| x_i^t - \mu_m^{\lceil t/\beta \rceil} \|^2 \Bigg). \end{aligned} \tag{2}$$

Since the correspondence variables c are not observable in the first place the common approach is to integrate over them, that is, to optimize the expected log likelihood $E_c[\ln p(d, c \mid \theta) \mid \theta, d]$ instead. Since the expectation is a linear operator we can move it inside the expression, so that we finally get:

$$\begin{aligned} E_c[\ln p(d, c \mid \theta) \mid \theta, d] = & \sum_{i=1}^{N} \Bigg(T \cdot M \cdot \ln \frac{1}{\sqrt{2\pi}\sigma} \\ & - \frac{1}{2\sigma^2} \cdot \sum_{t=1}^{T} \sum_{m=1}^{M} E[c_{im} \mid \theta, d] \| x_i^t - \mu_m^{\lceil t/\beta \rceil} \|^2 \Bigg), \end{aligned} \tag{3}$$

where $E[c_{im} \mid \theta, d]$ depends on the model θ and the data d.

Optimizing (3) is not an easy endeavor. EM is an algorithm that iteratively maximizes expected log likelihood 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. Mathematically, the standard method is to turn (3) in a so-called Q-function which depends on two models, θ and θ'. In accordance with (3), this Q-function is factored as follows [2]:

$$\begin{aligned} Q(\theta' \mid \theta) = & \sum_{i=1}^{N} \Bigg(T \cdot M \cdot \ln \frac{1}{\sqrt{2\pi}\sigma} \\ & - \frac{1}{2\sigma^2} \cdot \sum_{t=1}^{T} \sum_{m=1}^{M} E[c_{im} \mid \theta, d] \| x_i^t - \mu_m'^{\lceil t/\beta \rceil} \|^2 \Bigg). \end{aligned} \tag{4}$$

The sequence of models is then given by calculating

$$\theta^{[j+1]} \quad = \quad \underset{\theta'}{\operatorname{argmax}}\, Q(\theta' \mid \theta^{[j]}) \tag{5}$$

starting with some initial model $\theta^{[0]}$. Whenever the Q-function is continuous as in our case, the EM algorithm converges at least to a local maximum.

In particular, 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:

$$\begin{aligned} E[c_{im} \mid \theta^{[j]}, d] &= p(c_{im} \mid \theta^{[j]}, d) \\ &= p(c_{im} \mid \theta^{[j]}, d_i) \\ &= \eta p(d_i \mid c_{im}, \theta^{[j]}) p(c_{im} \mid \theta^{[j]}) \\ &= \eta' p(d_i \mid \theta_m^{[j]}), \end{aligned} \tag{6}$$

where the normalization constants η and η' ensure that the expectations sum up to 1 over all m. If we combine (1) and (6) exploiting the fact that the distributions are represented by Gaussians we obtain:

$$E[c_{im} \mid \theta^{[j]}, d_i] = \eta' \prod_{t=1}^{T} e^{-\frac{1}{2\sigma^2}\|x_i^t - \mu_m^{\lceil t/\beta \rceil [j]}\|^2}. \tag{7}$$

Finally, the M-step calculates a new model $\theta^{[j+1]}$ by maximizing the expected likelihood. Technically, this is done by computing for every motion pattern m and for each probability distribution $p(x \mid \theta_m^k)$ 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:

$$\mu_m^{k[j+1]} = \frac{1}{\beta} \cdot \sum_{t=(k-1)\cdot\beta+1}^{k\cdot\beta} \frac{\sum_{i=1}^{N} E[c_{im} \mid \theta^{[j]}, d] x_i^t}{\sum_{i=1}^{N} E[c_{im} \mid \theta^{[j]}, d]} \tag{8}$$

2.3 Estimating the Number of Model Components

Since in general the correct number of motion patterns is not known in advance, we need to determine this quantity during the learning phase. If the number of motion patterns is wrong, we can distinguish two different situations. First, if there are too few motion patterns, then there must be trajectories, that are not explained well by any of the current motion patterns. On the other hand, if there are too many motion patterns then there must be trajectories that are explained well by different model components. Thus, whenever the EM algorithm appears to have converged, we check whether the overall data likelihood can be improved by increasing or decreasing the number of model components. During the search, we therefore continuously monitor two types of occurrences:

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

2. Low motion pattern utility: If we can remove a model component θ_m without reducing the overall data likelihood, this indicates that θ contains a similar motion pattern and that θ_m is a duplicate. To detect such a redundant model component θ_m, we calculate the data log likelihood with and without θ_m. Technically, this involves executing the E step twice, once with and once without θ_m.

Whenever the EM has converged to a local maximum, our approach extracts those two statistics and considers deleting individual model components or introducing new ones. In particular, if a low data likelihood trajectory is found, a new model component is introduced that is initialized using this very trajectory. Conversely, if a motion pattern with low utility is found, it is eliminated from the model.

To limit the model complexity we use a penalty term $M^{[j]}\alpha$, where $M^{[j]}$ is the number of components of the model $\theta^{[j]}$ and α is an adjustable parameter that was set to 1.7 in all our experiments. This term ensures that we do not end up with a model that over-fits the data, which in the worst case is one with an individual motion pattern for every trajectory. Particularly, we compute

$$\begin{aligned} &(E_c[\ln p(d, c \mid \theta^{[j]}) \mid \theta^{[j]}, d] - M^{[j]}\alpha) - \\ &\quad (E_c[\ln p(d, c \mid \theta^{[j+1]}) \mid \theta^{[j+1]}, d] - M^{[j+1]}\alpha) \end{aligned}$$

where $E_c[\ln p(d, c \mid \theta^{[i]} \mid \theta^{[i]}, d]$ (see Equation 3) is the total expected data log likelihood. According to this difference we increase or decrease the number of model components.

Additionally, we store the model with the highest overall evaluation $E_c[\ln p(d, c \mid \theta^{[i]}) \mid \theta^{[i]}, d] - M^{[i]}\alpha$ encountered before the learning phase. If the maximum number of iterations is reached or if the overall evaluation cannot be improved after increasing and decreasing the model complexity our algorithms stops and returns the model with the best value.

In most of the experiments carried out with different data sets our approach correctly clustered the trajectories into the corresponding categories (see also the experiments section). Without this mechanism, however, EM frequently got stuck in local maxima and generated models that were significantly less predictive of human behavior.

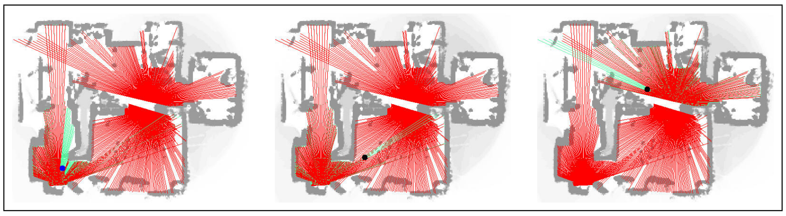

Figure 1: *Typical data sets obtained with three robots tracking a person in a home environment.*

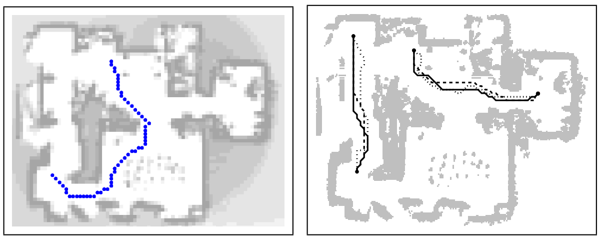

Figure 2: *A single trajectory extracted from the laser data (left image) and trajectories of two different classes of motion behaviors (right image).*

3 Laser-based Implementation

The EM-based learning procedure has been implemented for data acquired with laser-range finders. To acquire the data we used three Pioneer I robots which we installed in the environments. The robots were aligned so that they covered almost the whole environment. 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 we first extract the position of the person in the range scans. We locate changes in consecutive laser-range scans and use local minima in the distance histograms of the range scans. In a second step we identify resting places and perform a segmentation of the data into different slices in which the person moves. Furthermore, we smooth the data to filter out measurement noise. Finally, we compute the trajectories, 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 the left image of Figure 2.

4 Experimental Results

To evaluate the capabilities of our approach, we performed extensive experiments. The first set of experiments described here demonstrates the ability of our approach to learn different motion patterns from a set of trajectories. Then we analyze the classification performance of learned models. In all experiments we set the parameter β to 5 which we experimentally found out to yield good results regarding the success rate of learning correct models.

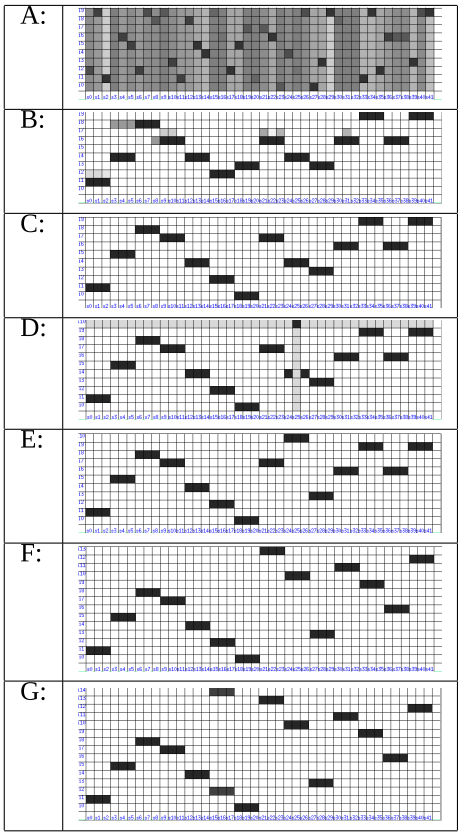

Figure 3: *Expectations $E[c_{im}|\theta^{[j]}, d]$ computed in the different iterations of the EM-algorithm.*

4.1 Application of EM

In the first experiment, we applied our approach to learn a motion model for 42 trajectories recorded in a home environment (see Figure 1). We started our algorithm with a model size of $M = 10$. Figure 3 shows for different rounds of the EM the resulting expectations $E[c_{i1} \mid \theta^{[j]}, d], \ldots, E[c_{iM^{[j]}} \mid \theta^{[j]}, d]$ for every trajectory d_i under the current model $\theta^{[j]}$ (the darker the more likely). The x-axis of each plot contains the trajectories $d_1, \ldots, d_N$ and the y-axis contains the different model components $\theta_i^{[j]}$, for $i = 1, \ldots, M^{[j]}$.

To enhance the readability we grouped the trajectories belonging to the same motion pattern. Please note that there are exactly three trajectories for each motion pattern in this particular data set.

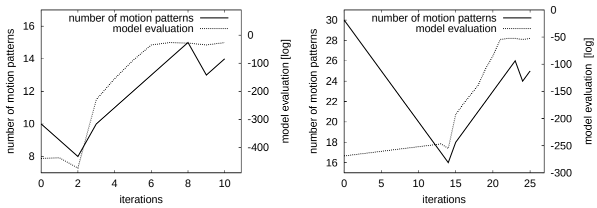

Figure 4: *Evolution of the number of model components and the overall evaluation of the model for the home (left) and the office environment (right).*

Since a uniform distribution of $E[c_{im} \mid \theta, d]$ represents a local maximum in the log likelihood space, the EM-algorithm immediately would get stuck if we initialize the expectations uniformly. To avoid this we initially use a unimodal distribution of the expectations for each trajectory, i.e., for each d_i the expectations $E[c_{i1} \mid \theta^{[0]}, d], \ldots, E[c_{iM} \mid \theta^{[0]}, d]$ form a distribution with a unique peak. The location of the mode, however, is chosen randomly.

The topmost image of Figure 3 labeled A shows the initialization of the expectations. The second image (plot B) shows the expectations after one iteration. In the situation corresponding to C the EM has converged to a local maximum in the log likelihood space given 10 different motion patterns. As can be seen from the figure, there are four model components that explain two different motion patterns. In the next step (image D) our algorithm therefore tries to improve the data likelihood by introducing a new model component to which it assigns the highest probability for trajectory 25 which has the lowest likelihood given the current model. The situation after the EM has again converged to a local maximum is shown in E. As before, another motion pattern is introduced. We omit the corresponding images for the sake of brevity. The correct classification is found after our algorithm has introduced four additional model components (plot F). As can be seen from the figure, the system has determined a model in which all trajectories are correctly clustered into motion patterns. Nevertheless, our algorithm still tries to further improve this model by removing and adding a model component. However, since none of these operations increases the overall evaluation, our algorithm terminates and outputs the model corresponding to F. Plot G shows the expectations for a model with 15 components. As can be seen from the histogram, three trajectories are assigned to two different model components. Because of the penalty term our algorithm prefers the lower complexity model corresponding to histogram F.

Figure 2 (right) shows the corresponding trajectories of two different motion behaviors after the convergence of the EM. Obviously, the trajectories are correctly clustered.

Figure 4 (left) shows for the same data set but for a different initialization of the expectations the evolution of the overall evaluation $E_c[\ln p(d, c \mid \theta^{[i]}) \mid \theta, d^{[i]}] - M^{[i]}\alpha$ as well as the number of model components. As can be seen from the figure, our algorithm first tries to reduce the model complexity. This is because the model contains components with very low utilities.

We additionally applied our algorithm to data recorded in our office environment (the map is depicted in Figure 5 (left)). Figure 4 (right) shows the model complexity and model evaluation for one run in which we started with 30 different motion patterns. As can bee seen from the figure, the algorithm decreases the model complexity until only 16 (non-redundant) components remain. Afterwards it increases this number to improve the model evaluation. Finally, it terminates with the model correctly representing 25 different motion patterns.

To evaluate the performance of our approach we carried out 100 experiments for each data set. In every 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 100% of the cases using the data recorded in the home environment and in 91% of the cases using the data recorded in the office environment. The left image of Figure 5 shows two trajectories that our algorithm falsely classified in all situations in which it failed. As can be seen from the figure, both trajectories are extremely similar although they actually belong to different motion behaviors.

4.2 Predicting Trajectories

To evaluate the capability of our learned models to predict human motions we performed a series of experiments. In each experiment we randomly chose fractions of test trajectories and computed the likelihood of the correct motion pattern. Figure 5 (right) shows the average likelihood of the correct motion behavior depending on the length of the observed fraction. As can be seen from the figure, the classification results are quite good and our approach yields models allowing a mobile robot to reliably identify the correct motion pattern. For example, if the robot observes 50% of a trajectory, then the probability of the correct motion behavior is about 0.6 in both environments.

Figure 5 (center) illustrates for one trajectory of the person in the office environment the evolution of the set of possible motion behaviors. Shown in grey are the means of four different motion patterns. The black line corresponds to the trajectory of the person which was observed for the first time at the position labeled S. In the beginning there are four possible motion behaviors (W, B, D, M) to which the trajectory might belong. When location 1 is reached the motion behavior W can be eliminated from the set of hypotheses because the corresponding likelihood gets too low. Thus, even if the system is not able to uniquely determine the intended goal location, it can already predict that the person will follow the corridor during the next steps. When the person reaches location 2

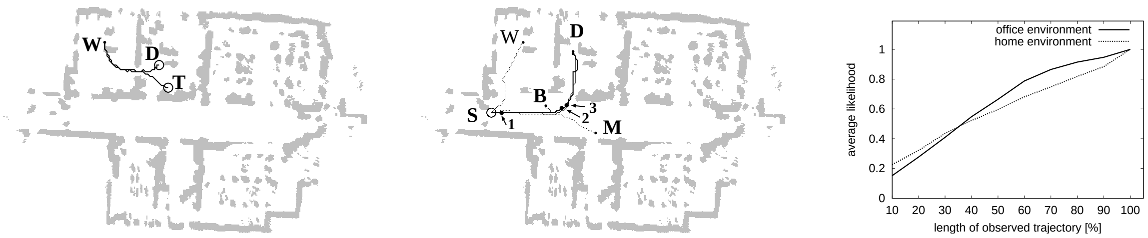

Figure 5: *The two classes of trajectories which sometimes are classified into the same motion behavior (left), an example of motion prediction (center) and the average likelihood of the correct motion behavior after observing fractions of trajectories (right).*

the system can also exclude the motion behavior B. Finally, when the person reaches position 3, C becomes unlikely and D becomes the most probable motion behavior. This illustrates, that the results of the prediction are useful even in situations in which there are ambiguities about the actual intention of the person.

5 Conclusions

In this paper we presented a method for learning motion behaviors of persons in indoor environments. Our approach applies the popular EM-algorithm to cluster similar behaviors into single patterns. Thereby it is able to estimate the number of motion patterns. Additionally, it can deal with trajectories of different length. The output of our algorithm is a collection of motion patterns, each corresponding to a principle motion behavior of a person. Using the resulting motion patterns our system can predict the motions of persons based on observations made by the robot.

Our approach has been implemented and applied to range data recorded with mobile robots equipped with laser-range sensors. In practical experiments we demonstrated that our method is able to reliably learn typical motion behaviors of a person in a domestic residence as well as in an office building. We furthermore described how to use the resulting models to predict the motions of persons in the vicinity of the robot.

References

[1] 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 Int. Joint Conference on Artificial Intelligence (IJCAI)*, 1997.

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

[3] 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.

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

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

[7] 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.

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

[9] 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.

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

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

[12] N. Roy, G. Baltus, D. Fox, F. Gemperle, J. Goetz, T. Hirsch, D. Magaritis, M. Montemerlo, J. Pineau, Schulte J., and S. Thrun. Towards personal service robots for the elderly. In *Proc. of the Workshop on Interactive Robotics and Entertainment*, 2000.

[13] 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.

[14] R.D. Schraft and G. Schmierer. *Serviceroboter*. Springer Verlag, 1998. In German.

[15] D. Schulz, W. Burgard, D. Fox, and A.B. Cremers. Tracking multiple moving targets with a mobile robot using particle filters and statistical data association. In *Proc. of the Int. Conference on Robotics & Automation (ICRA)*, 2001.

[16] 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 Int. Conference on Intelligent Robots and Systems (IROS)*, 1995.

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