Using EM to Learn Motion Behaviors of Persons with Mobile Robots

Maren Bennewitz   Wolfram Burgard   Sebastian Thrun

Also involved: Dirk Schulz, Dirk Hähnel and many others

Motivation

- Robots that know where people are and what they do can do better!
- Examples…

Minerva

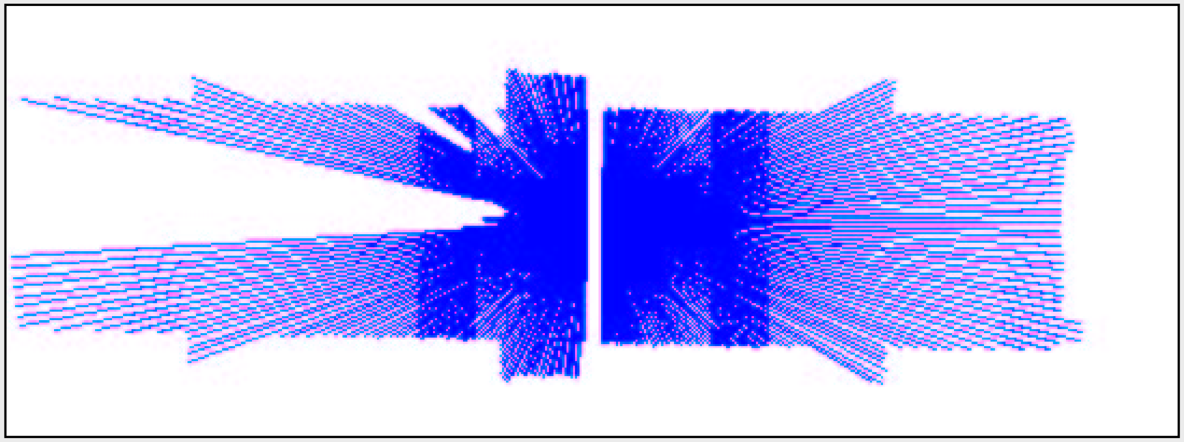
Perl:
A Nursing Robot

Albert:
An Interactive Service Robot

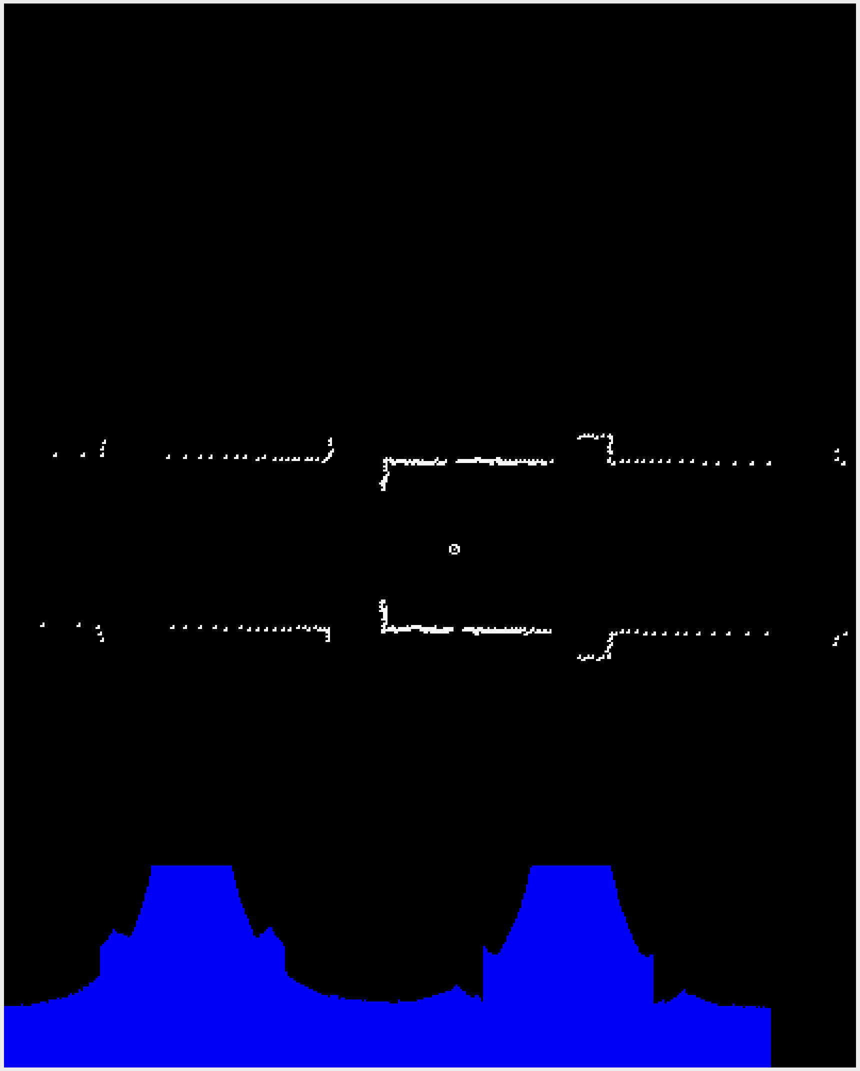
Three-Month Deployment of Albert at the HNF

Tracking People

- **Key questions**
  - How many people are there?
  - Where do they go?
- **Requirements**
  - Real time
  - No model of the environment
  - Robot in motion



Example Run



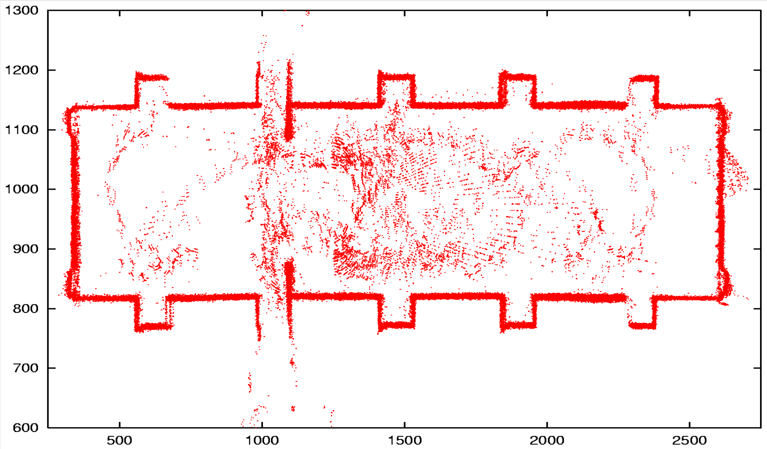
[ICRA-2001]

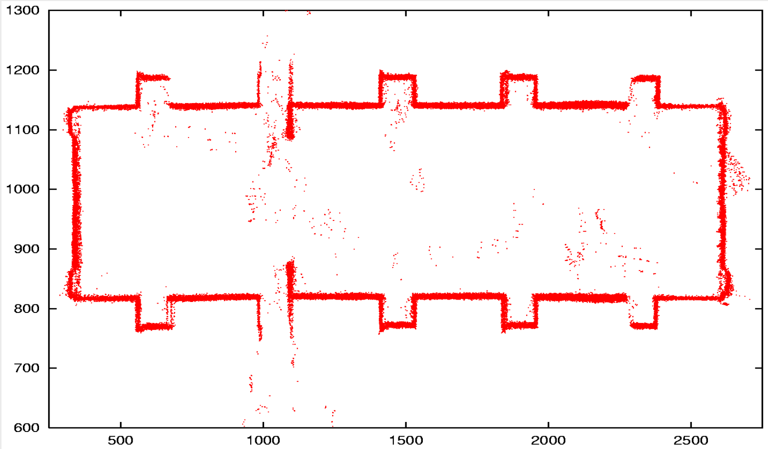
Tracking with a Moving Robot

[CVPR-2001]

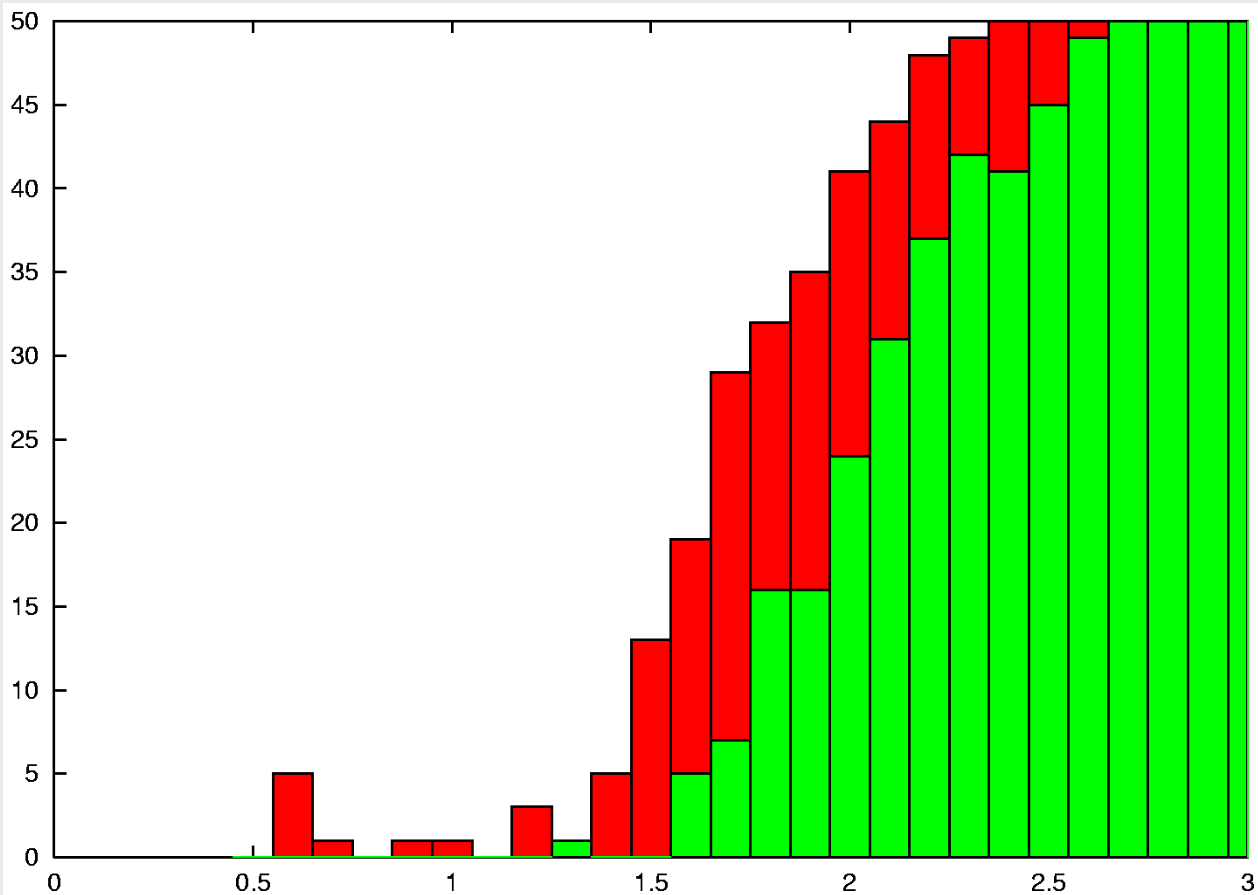
Mapping in Populated Environments

Filtering beams corresponding to persons improves maps:

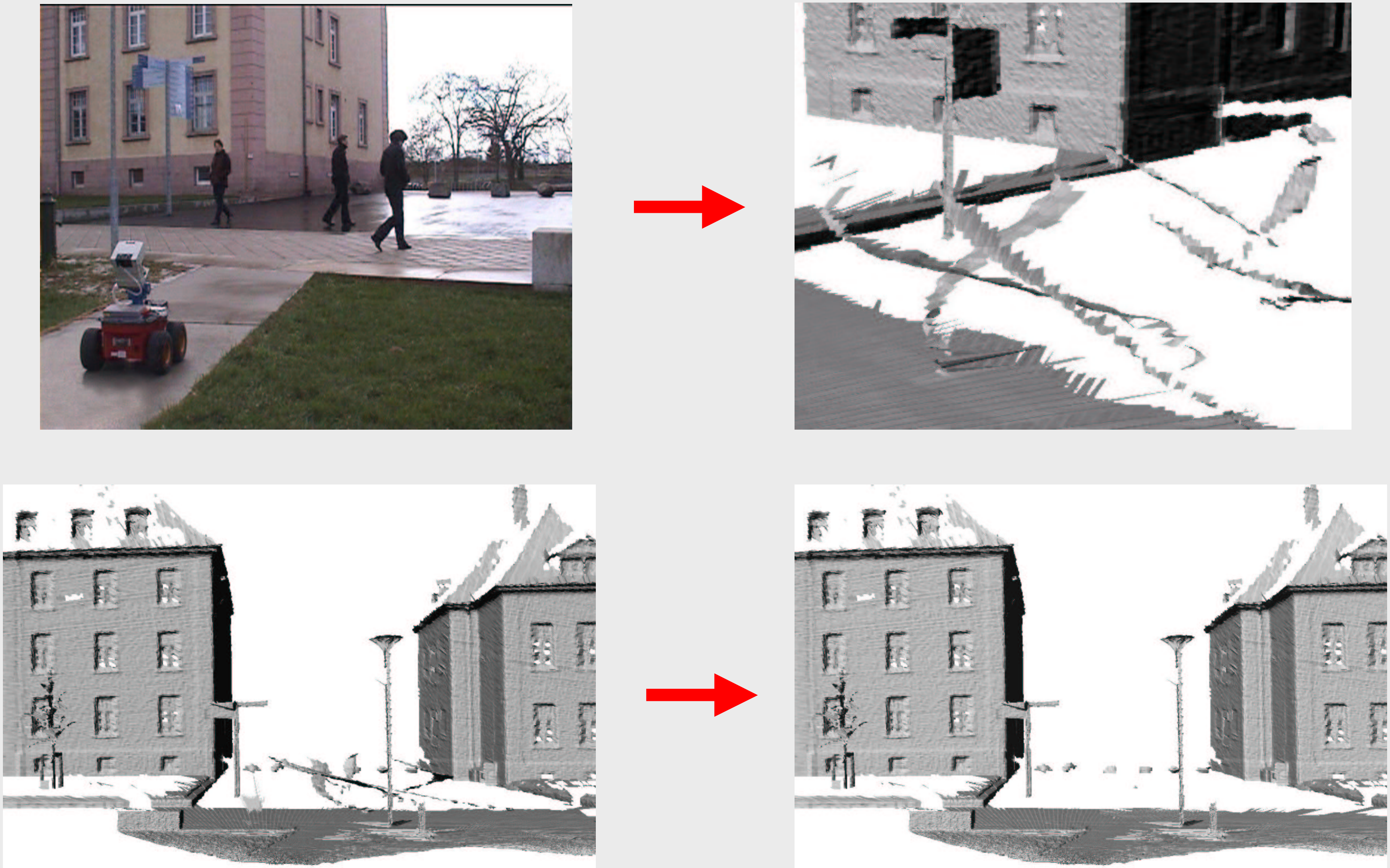




Increased Matching Accuracy by Filtering People



Learning 3d-Maps



Learning Motion Patterns

Knowledge of typical motion patterns helps robots to

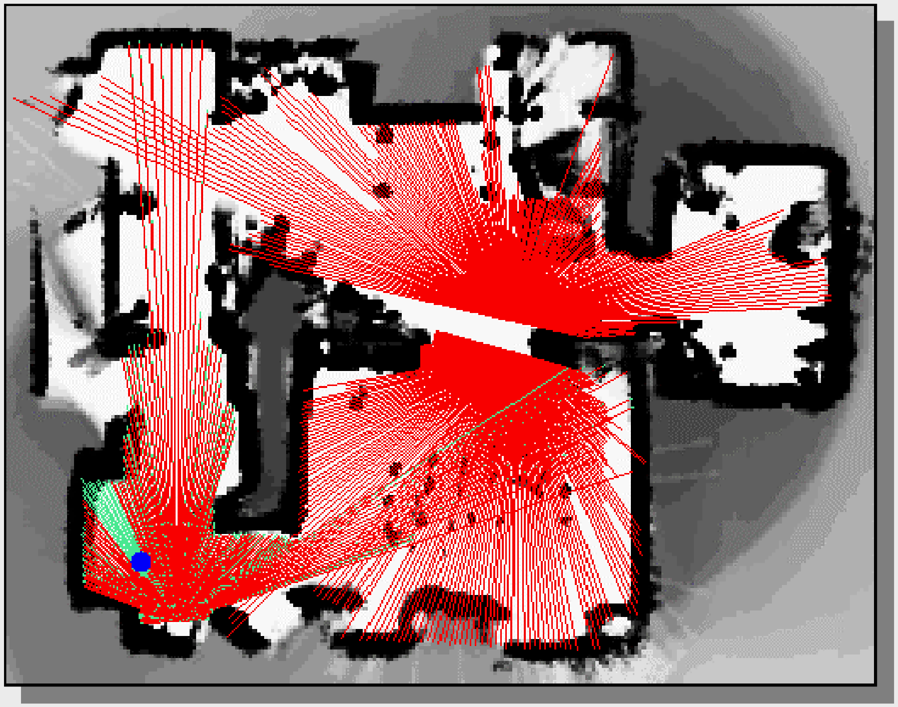
- predict behavior of persons
- avoid possible conflicts
- improve their service
- …

2D Map of a Domestic Environment, Learned by a Robot

Learning Trajectories of People in Their Homes

- **Which trajectory does the person take?**
- **Where is the person going to?**

Tracking People/Motion Segmentation



Input: Set S of data sequences $s_1, \ldots, s_N$

What we are looking for:

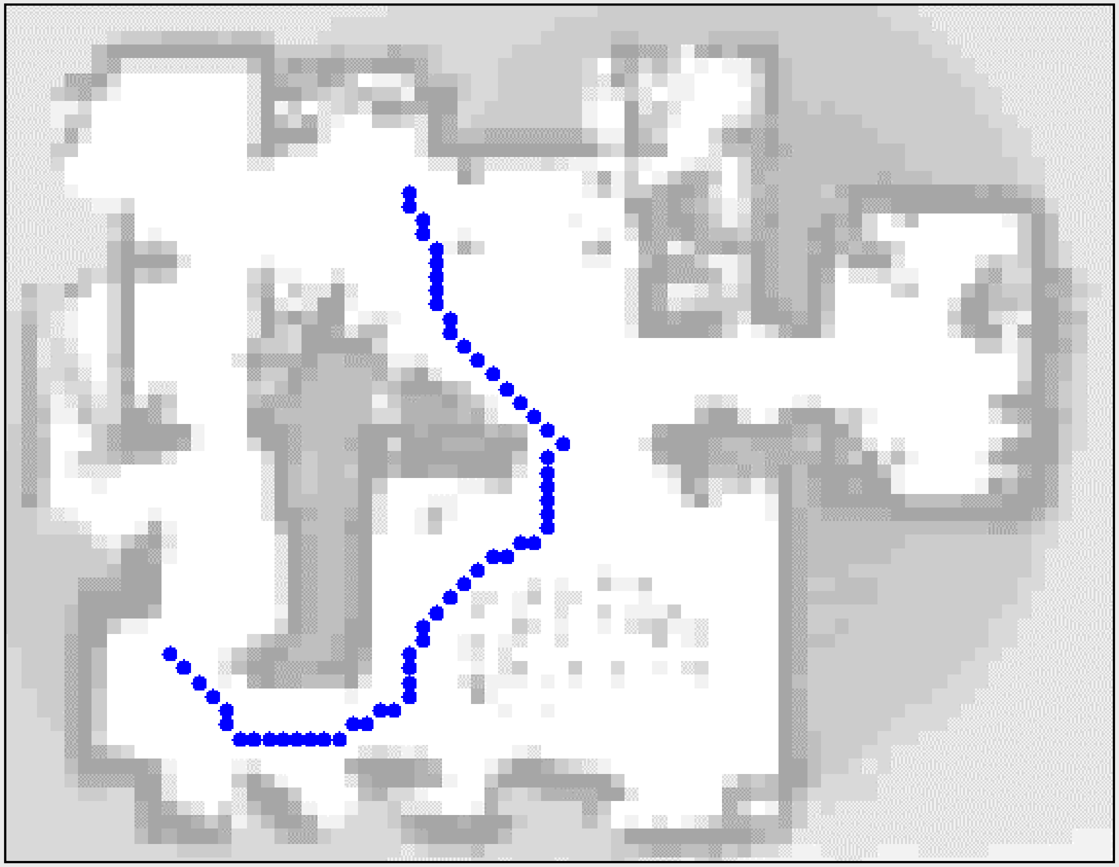
- Set θ of position-sequences $\theta_1, \ldots, \theta_M$, one for each pattern.
- Correspondence table $x_{m,n}$ telling us, which data s_n set belongs to which motion pattern θ_m.

Problem:

How can we estimate $x_{m,n}$?

Density Representation

- One Gaussian with fixed variance for every time step of every motion pattern



Formal Specification

We want to maximize

$$E_x[\log p(s, x \mid \theta)] = E[c_1 - c_2 \sum_{n=1}^{N} \sum_{m=1}^{M} x_{m,n} \log p(s_n \mid \theta_{\mathrm{m}})]$$

Linearity of $E[\ldots]$

$$= c_1 - c_2 \sum_{n=1}^{N} \sum_{m=1}^{M} E[x_{m,n}] \log p(s_n \mid \theta_{\mathrm{m}})$$

Gaussians

$$= c_1 - c_2 \sum_{n=1}^{N} \sum_{m=1}^{M} \sum_{t=1}^{T} E[x_{m,n}] \cdot \left\| s_n^t - \mu_{\mathrm{m}}^t \right\|$$

Extension of k-means clustering to trajectories!

Solution by Applying the EM-Algorithm

Maximize $E_x[\log p(s, x \mid \theta)]$ through an iterative sequence of models $\theta^1, \theta^2, \ldots$

E-Step:

$$E[x_{m,n}] \leftarrow \alpha\, p(s_n \mid \theta_{\mathrm{m}}) = \alpha \prod_{t=1}^{T} e^{-\frac{\left\| x_n^t - \mu_m^t \right\|}{2\sigma^2}}$$

The M-Step

$$\theta_m \leftarrow \arg\max_{\theta_m} \sum_{n=1}^{N} \sum_{m=1}^{M} E[x_{m,n}] \cdot \log p(s_n \mid \theta_\mathrm{m})$$

Since we have Gaussians with a fixed variance:

$$\mu_m^t \leftarrow \frac{\sum_{n=1}^{N} E[x_{m,n}] \cdot \mathrm{x}_\mathrm{n}^\mathrm{t}}{\sum_{n=1}^{N} E[x_{m,n}]}$$

Estimating the Number of Model Components

Whenever EM has converged to a (local) maximum:

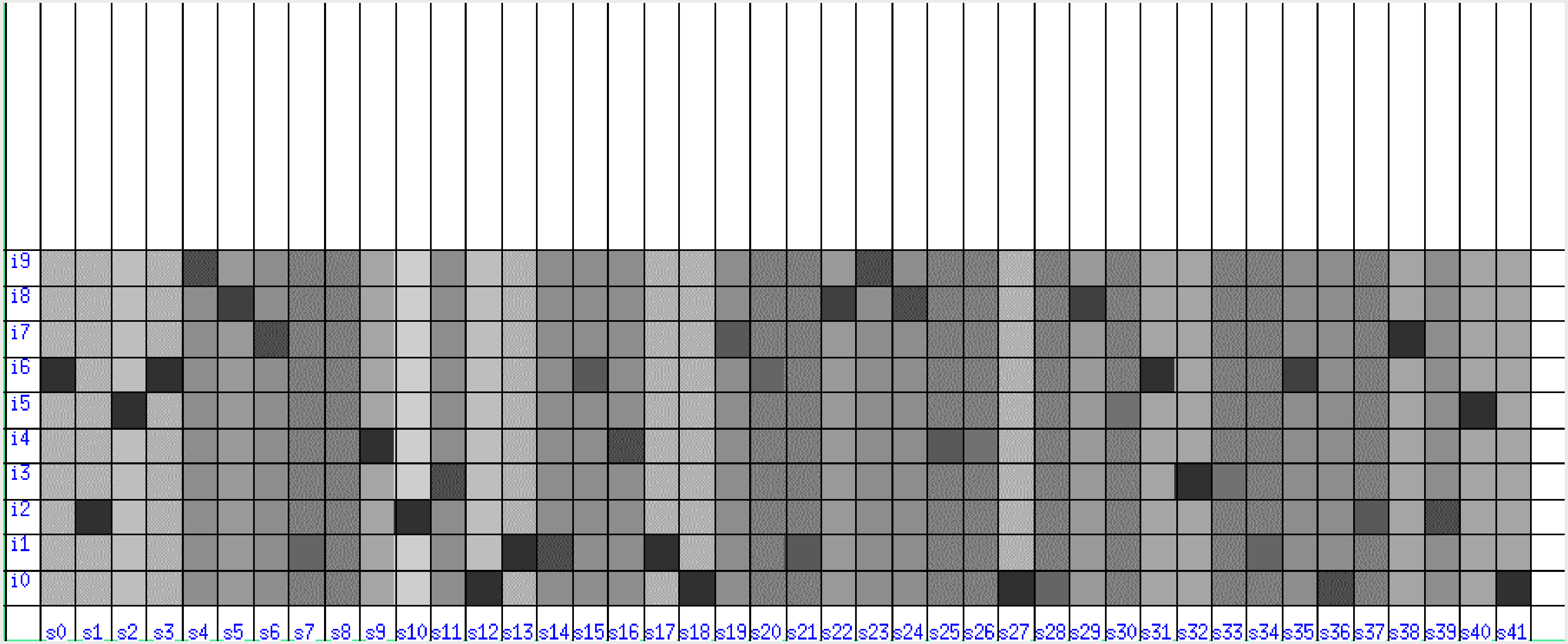
1. Try to introduce a new motion pattern for the trajectory which has the lowest likelihood under the current model.
2. Try to eliminate the motion pattern which hast the lowest utility.

Select model θ which has the highest evaluation

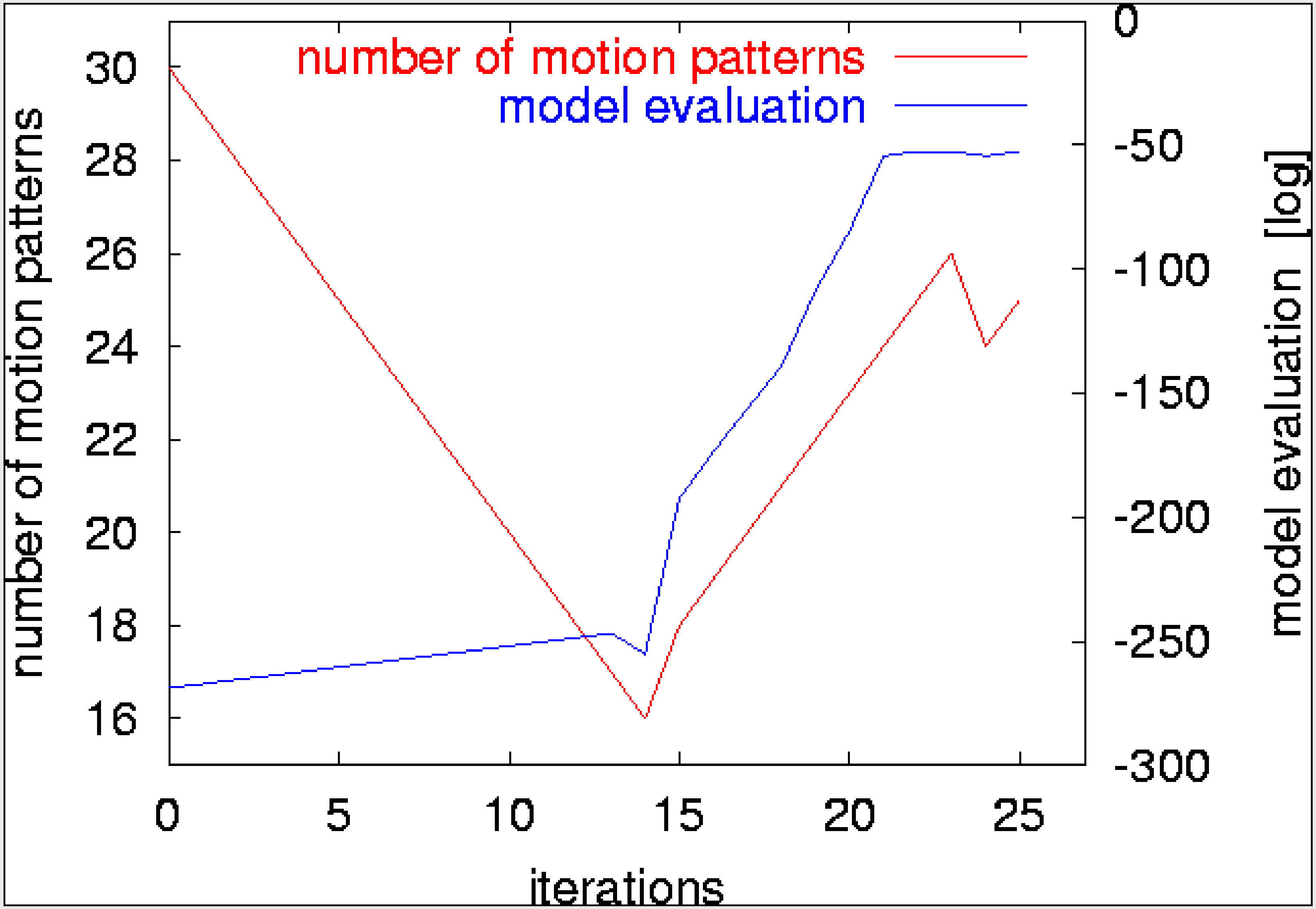
$$E_x[\log p(s, x \mid \theta)] - M\alpha$$

where M = #model components, α = penalty term

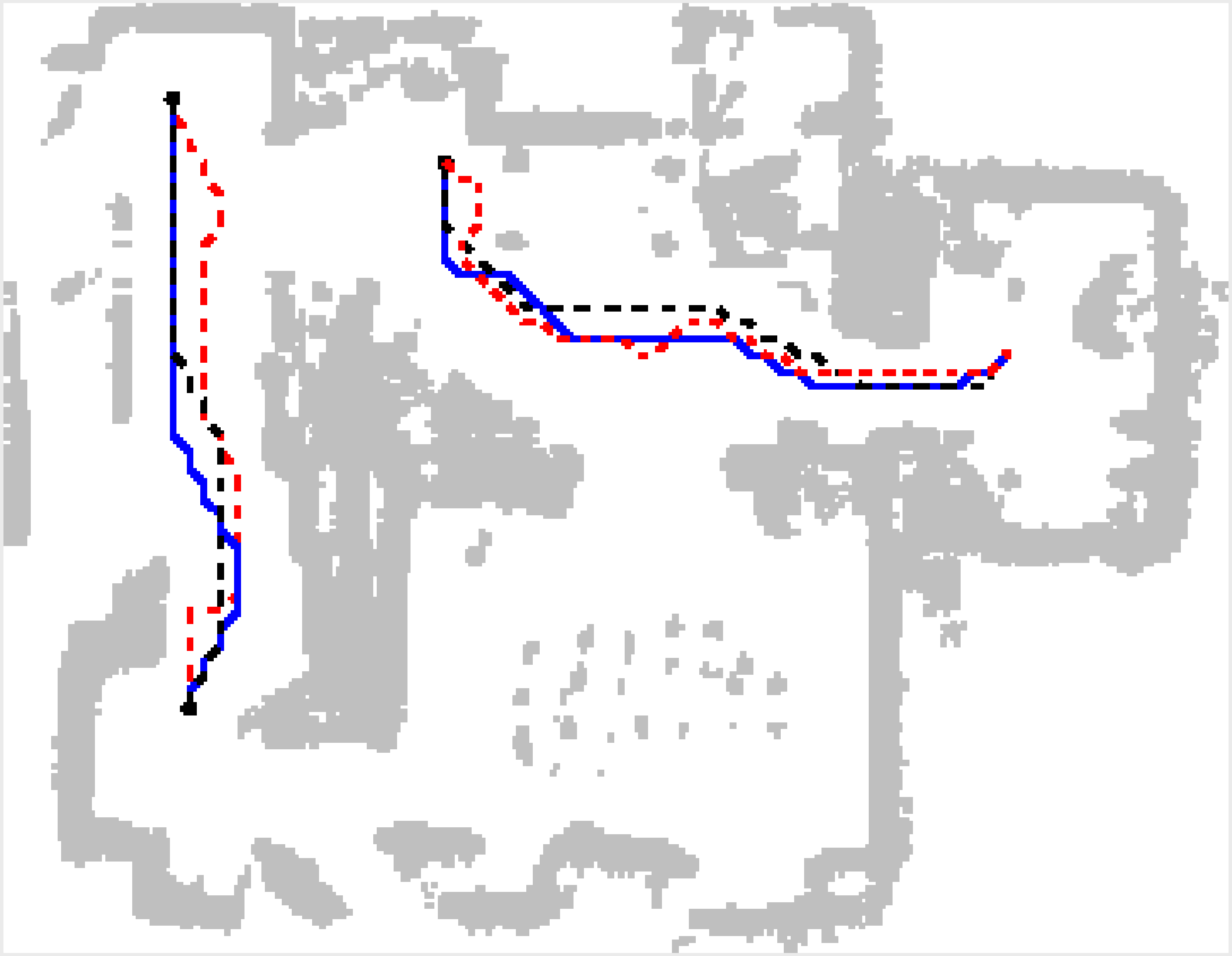
Application of EM



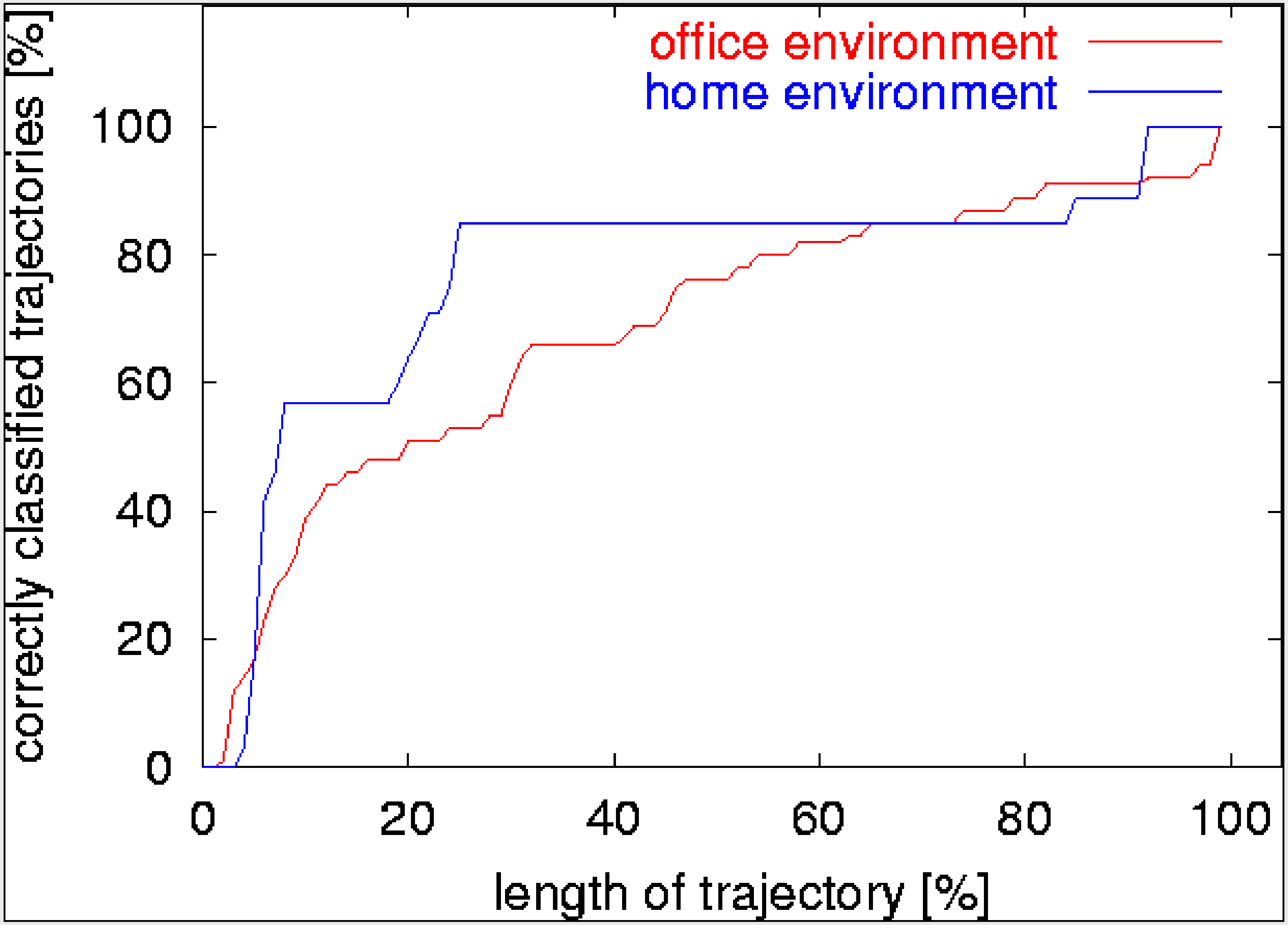
Model Selection



Clustering Results

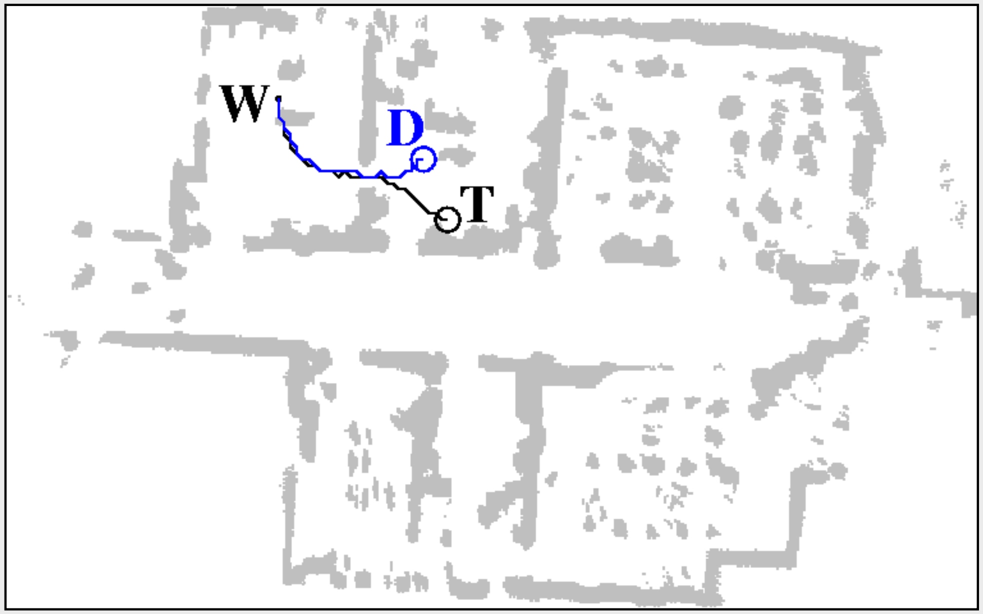


Prediction Accuracy

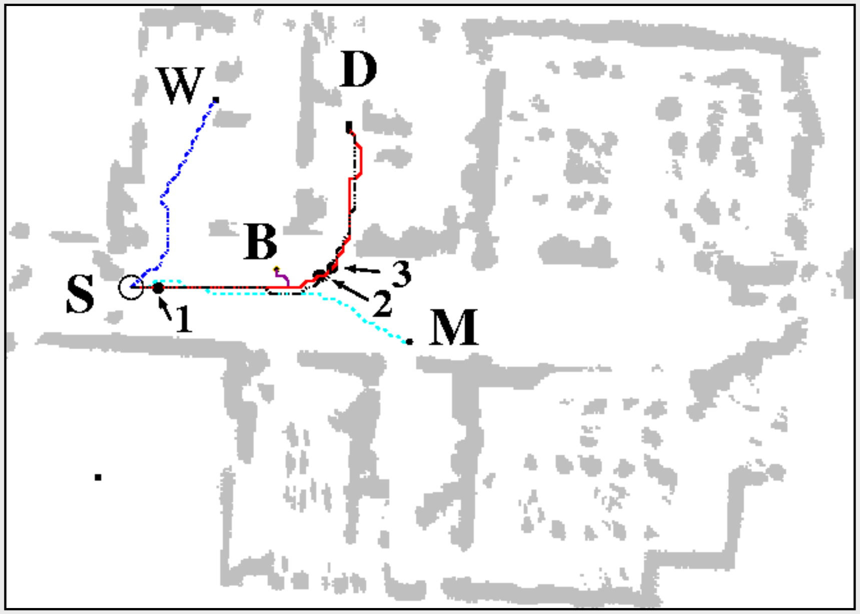


Why it's sometimes difficult ...

during learning:



during classification:

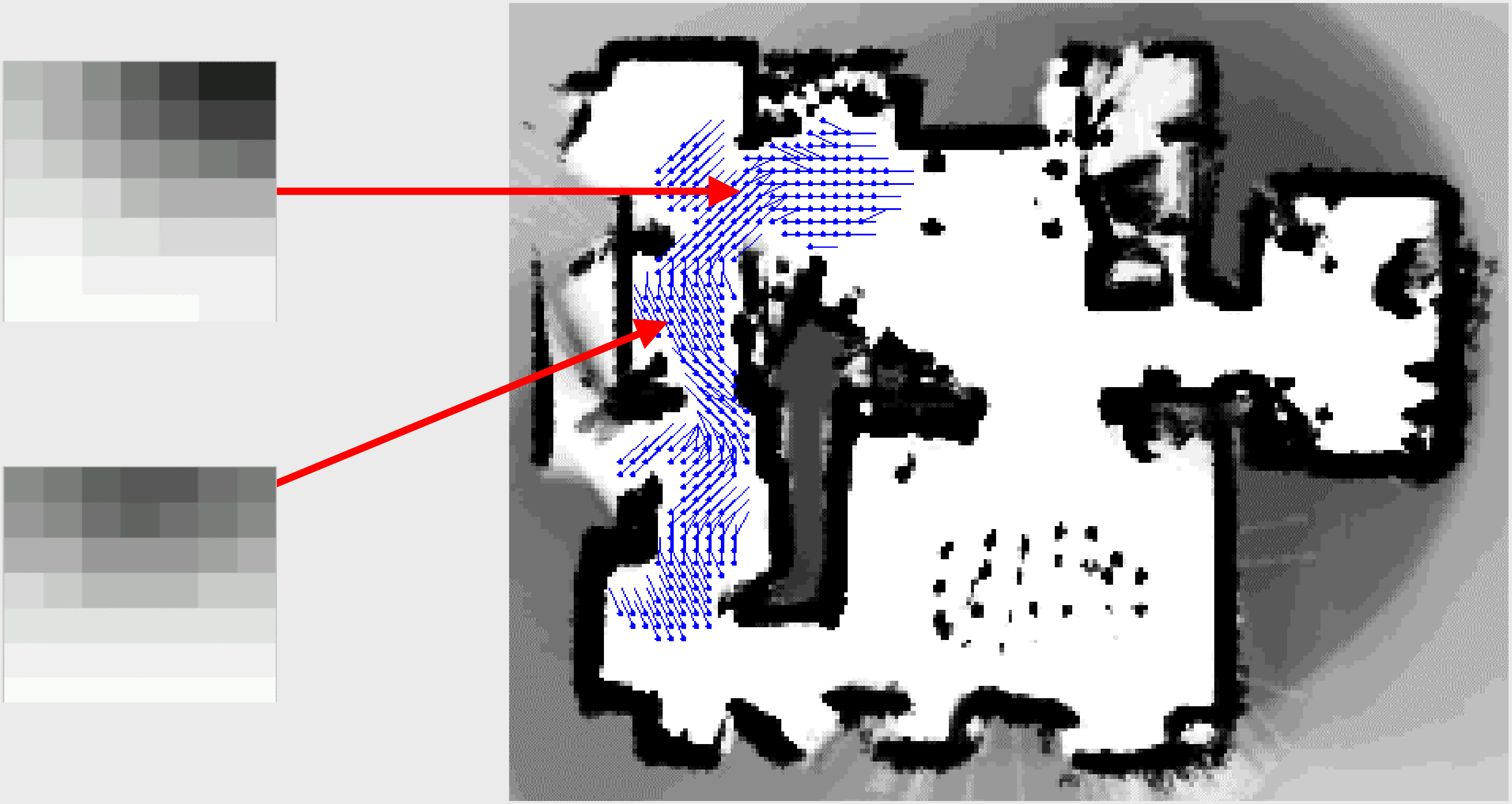


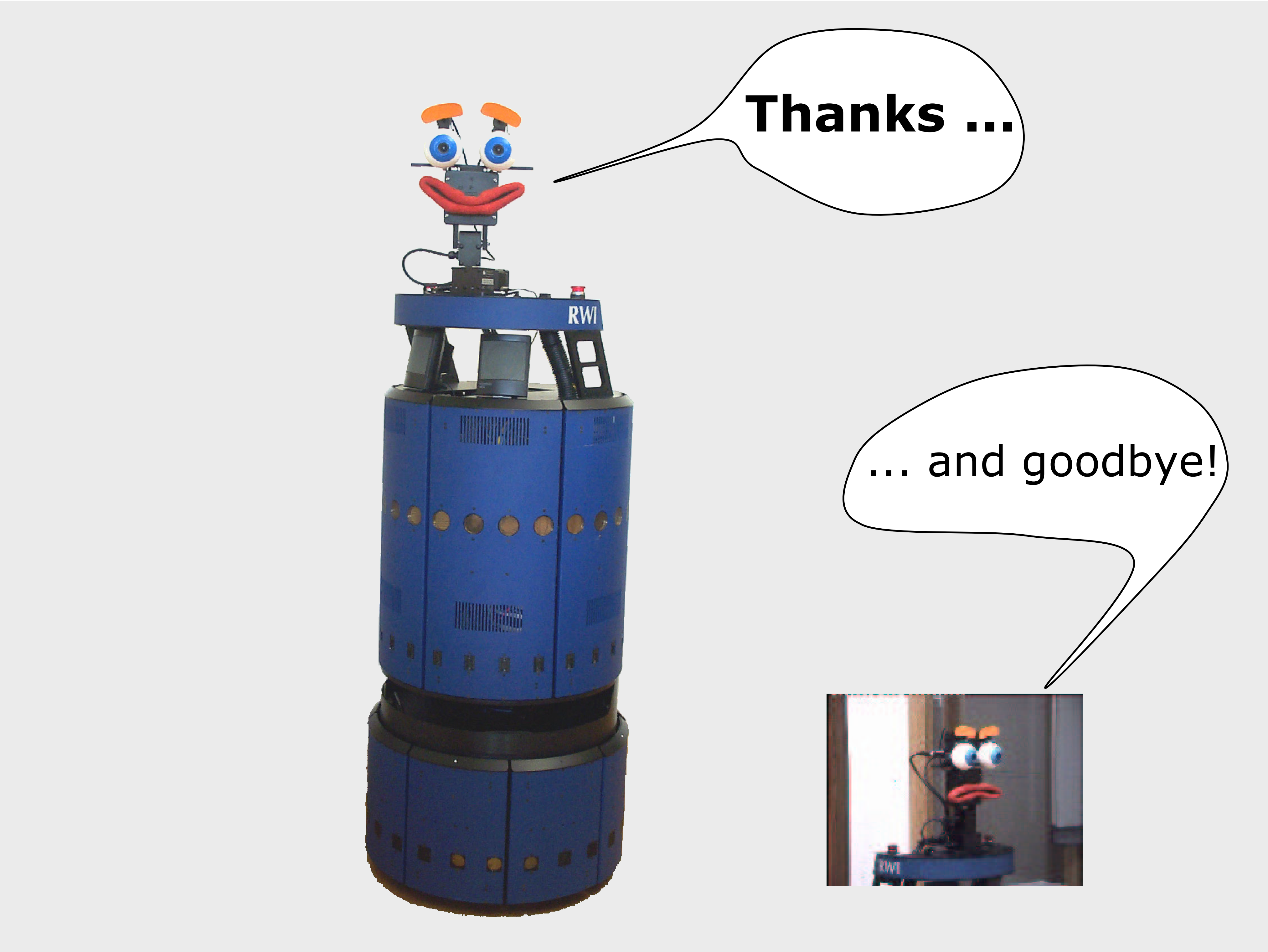
... because there are serious overlaps!

Conclusions and Future Work

- **Technique to learn motion patterns of people in home and office environments.**
- **Learning more abstract patterns (lower complexity models, e.g. linear piecewise approximations)**
- **Adapting the robot's behavior according to the predicted behavior**
- **Applications**

Example: Markov Chains




Thanks ...
RWI
... and goodbye!