# Socially Compliant Mobile Robot Navigation via Inverse Reinforcement Learning

Henrik Kretzschmar, Markus Spies, Christoph Sprunk, Wolfram Burgard

Department of Computer Science, University of Freiburg, Germany

**Abstract**

Mobile robots are increasingly populating our human environments. To interact with humans in a socially compliant way, these robots need to understand and comply with mutually accepted rules. In this paper, we present a novel approach to model the cooperative navigation behavior of humans. We model their behavior in terms of a mixture distribution that captures both the discrete navigation decisions, such as going left or going right, as well as the natural variance of human trajectories. Our approach learns the model parameters of this distribution that match, in expectation, the observed behavior in terms of user-defined features. To compute the feature expectations over the resulting high-dimensional continuous distributions, we use Hamiltonian Markov chain Monte Carlo sampling. Furthermore, we rely on a Voronoi graph of the environment to efficiently explore the space of trajectories from the robot's current position to its target position. Using the proposed model, our method is able to imitate the behavior of pedestrians or, alternatively, to replicate a specific behavior that was taught by tele-operation in the target environment of the robot. We implemented our approach on a real mobile robot and demonstrate that it is able to successfully navigate in an office environment in the presence of humans. An extensive set of experiments suggests that our technique outperforms state-of-the-art methods to model the behavior of pedestrians, which makes it also applicable to fields such as behavioral science or computer graphics.

## 1 Introduction

In the near future, more and more mobile robots will populate our human environments. Applications include robots that provide services in shopping malls, robotic co-workers in factories, or even assistive robots in healthcare. Traditional approaches to mobile robot navigation, such as moving on time-optimal paths, are not always desirable for such robots. Instead, robots that navigate in the same

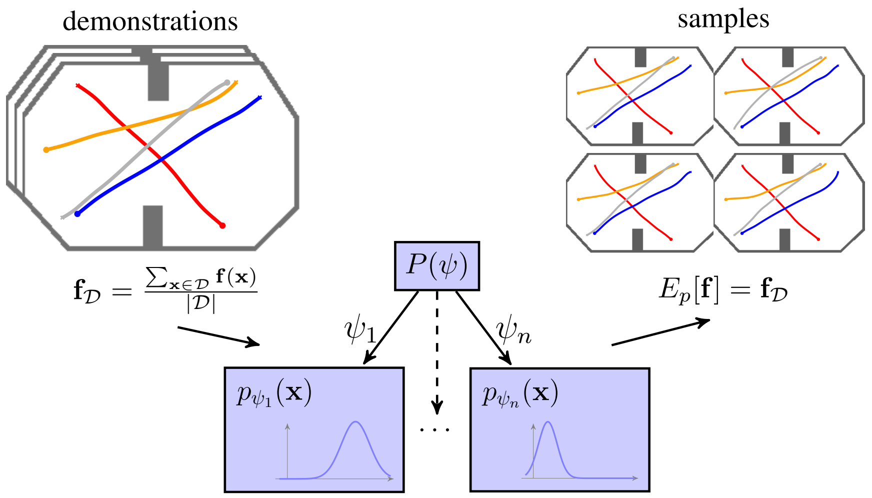

demonstrations
samples
$\mathbf{f}_{\mathcal{D}} = \frac{\sum_{\mathbf{x}\in\mathcal{D}} \mathbf{f}(\mathbf{x})}{|\mathcal{D}|}$
$P(\psi)$
$\psi_1$
$\psi_n$
$E_p[\mathbf{f}] = \mathbf{f}_{\mathcal{D}}$
$p_{\psi_1}(\mathbf{x})$
$p_{\psi_n}(\mathbf{x})$

Figure 1: Our method is able to learn a model of human cooperative navigation behavior from demonstrations. We learn the model parameters of a mixture distribution over composite trajectories to capture the discrete and continuous aspects of the behavior. The learned model generalizes to new situations and allows us to draw trajectory samples that capture the stochasticity of natural navigation behavior.

environment as humans should understand and comply with social norms that allow humans to smoothly evade each other even in crowded environments. In particular, humans tend to cooperate to avoid collisions. To enable socially compliant human-robot interaction that does not disturb nearby humans, mobile robots need to engage in such cooperative navigation. To this end, we need accurate models of the navigation behavior of interacting pedestrians.

In this paper, we propose an approach that allows a mobile robot to learn such models from observations of pedestrians. Specifically, our goal is to model the decision-making process of the pedestrians that ultimately makes them move on the continuous trajectories that we can observe. This decision-making process typically involves continuous and discrete navigation decisions. Whereas continuous navigation decisions affect the resulting trajectories in terms of distances to obstacles and higher-order dynamics such as velocities and accelerations, discrete navigation decisions determine whether to pass obstacles or other agents on the left or on the right side. In addition to that, human navigation behavior is not deterministic. Humans rather exhibit stochastic properties, i. e., their trajectories vary from run to run, which becomes apparent when they repeatedly navigate in the same situation. To capture these properties, we model the navigation behavior in terms of a joint mixture distribution over the trajectories of all the agents involved in an encounter.

As illustrated in Figure 1, this distribution comprises a probability distribution over the outcomes of the discrete decision-making process. For each outcome, a continuous distribution captures the natural variance of the agents' trajectories. We propose a feature-based maximum entropy learning approach to infer the parameters of the model that matches the observed behavior in expectation. The resulting model allows the robot to predict the behavior of multiple agents in new situations in terms of a joint probability distribution over their trajectories.

A mobile robot can use the learned model to predict the behavior of nearby pedestrians and to react accordingly. To this end, the robot maintains a probability distribution over its own actions and the actions of the pedestrians in the current situation. The most likely interaction behavior encodes a prediction of the pedestrians' behavior as well as the desired behavior of the robot itself, according to the learned model. In some applications, however, it may not be desirable to simply make the robot replicate human navigation strategies. In these cases, the proposed model also allows us to teach the robot a certain behavior by tele-operation, which is an intuitive way, especially for non-experts. During tele-operation, the robot learns how to behave during encounters with pedestrians.

Our approach assumes cooperative agents, i.e., agents that behave in a way that allows all involved agents to reach their targets as comfortably as possible. This assumption, however, does not hold in certain situations. For example, to model human behavior in evacuation scenarios, it would be necessary to introduce game-theoretic aspects, where each agent tries to increase its own utility. However, our experiments suggest that the assumption of cooperative agents is reasonable for navigation under normal circumstances, such as navigating in an office environment.

The contribution of this work is a probabilistic framework to learn the behavior of interacting agents such as pedestrians from demonstration. A key challenge of such a learning approach is the so-called forward problem, i. e., computing for a given model the expected feature values with respect to the distribution over the high-dimensional space of continuous trajectories. We propose to use Markov chain Monte Carlo (MCMC) sampling and exploit that the distributions over observed trajectories of interacting agents are highly structured. The use of a spline-based trajectory representation makes it possible to efficiently compute the gradient of the probability density, which allows our method to guide the sampling process towards regions of high probability using the Hybrid Monte Carlo (HMC) algorithm (Duane et al., 1987). Therefore, our method is able to capture the stochasticity of observed trajectories, which is in contrast to existing approaches that learn deterministic models that do not replicate well the stochastic behavior of natural agents. Furthermore, we present efficient techniques to explore the space of homotopy classes of the trajectories using a Voronoi graph of the environment. In addition, we present methods to integrate the learned model into a navigation framework for mobile robots. An extensive set of experiments suggests that our model outperforms state-of-the-art methods to model pedestrian navigation behavior. The experiments include a Turing test in which the trajectories computed by our approach were perceived as more human than the trajectories computed by other methods. The experiments also evaluate the performance of our approach in navigation tasks with a real robot. This paper builds on our previous work presented in (Kretzschmar et al., 2014; Kuderer et al., 2012, 2013, 2014) and provides a unified presentation of our approach, a more detailed description of the proposed methods, and a more comprehensive experimental evaluation.

The remainder of this paper is structured as follows. In the next section, we review work related to our approach. In Section 3, we present our learning framework to modeling the navigation behavior of pedestrians. In Section 4, we apply our model to mobile robot navigation in populated environments. In Section 5, we present a thorough experimental evaluation of our method.

## 2 Related Work

There is a wide range of literature on learning policies from demonstrations (Argall et al., 2009). Atkeson and Schaal (1997) developed one of the pioneering approaches to infer a mapping from state features to actions to directly mimic observed behavior. More recently, Ng and Russell (2000) applied inverse reinforcement learning (IRL) to recover a cost function that explains observed behavior. In particular, Abbeel and Ng (2004) suggest to match features that capture relevant aspects of the behavior that is to be imitated. However, matching features does not lead to a unique cost function.

To resolve this ambiguity, Ziebart et al. (2008) present Maximum Entropy IRL that relies on the principle of maximum entropy (Jaynes, 1978) and, hence, aims at finding the policy with the highest entropy subject to feature matching. Their method works well in discrete state-action spaces of low dimensionality. However, discretizing the state-action spaces does not scale well to continuous trajectories, especially when taking into account higher-order dynamics such as velocities and accelerations. Ratliff et al. (2006); Vernaza and Bagnell (2012); Ziebart et al. (2009) amongst others applied these indirect learning approaches to a variety of problems including route planning for outdoor mobile robots and learning pedestrian navigation behavior. Ziebart et al. (2012) cast the trajectories of pointing devices, such as a computer mouse, as a control problem in order to predict the targets of users. Kitani et al. (2012) use Maximum Entropy IRL to infer the preferences of pedestrians with respect to vision-based physical scene features such as sidewalks. Their model predicts the trajectory of a pedestrian taking into account these features using a discrete Markov decision process. Similarly, Kim and Pineau (2015) use a Bayesian inverse reinforcement learning approach to infer a cost function for socially adaptive path planning, where the cost function is expressed in terms of features that are computed based on RGB-D sensor readings.

Inspired by Abbeel and Ng (2004) and Ziebart et al. (2008), our approach aims to find maximum entropy distributions that match observed feature values. However, in contrast to the abovementioned techniques, our approach reasons about joint probability distributions over trajectories of multiple agents whose actions potentially mutually affect each other. In addition to that, our method reasons about the agents' trajectories in continuous state spaces including their higher-order dynamics.

In many inverse reinforcement learning approaches, estimating the feature expectations is a challenging problem, especially in high-dimensional state spaces of continuous trajectories. To this end, Boularias et al. (2011) apply importance sampling to compute the gradient of the model parameters while Vernaza and Bagnell (2012) constrain the features to have a certain low-dimensional structure. Kalakrishnan et al. (2013) propose to assume the demonstrations to be locally optimal and sample continuous trajectories by adding Gaussian noise to the model parameters. Kuderer et al. (2012) approximate the feature expectations using Dirac delta functions at the modes of the distribution. This approximation, however, leads to suboptimal models when learning from imperfect human navigation behavior since its stochasticity is not sufficiently captured. In practice, this method underestimates the feature values and thus favors samples from highly unlikely homotopy classes. In contrast to that, the approach that we present in this paper efficiently estimates the feature expectations using Hybrid Monte Carlo sampling (Duane et al., 1987), which is applicable to arbitrary differentiable features. Hybrid Monte Carlo ideas are also used by Ratliff et al. (2009b) in the context of globally robust trajectory optimization.

In contrast to probabilistic learning approaches, many researchers have proposed models to capture the complex navigation behavior of humans (Bitgood and Dukes, 2006; Christensen and Pacchierotti, 2005; Hall, 1966; Yoda and Shiota, 1996, 1997). Zambrano et al. (2012) divides these methods to model human navigation behavior into steering models and optimization models.

Steering models describe human navigation behavior as a dynamical system in which a set of rules determines the agent's immediate action given its current state in the environment (Helbing and Johansson, 2009; Helbing and Molnar, 1995; Johansson et al., 2007; Lerner et al., 2007; Müller et al., 2008; Warren, 2006). The social forces method presented by Helbing and Molnar (1995) models pedestrian motion behavior in terms of forces that correspond to internal objectives of humans, such as the desire to reach a target and to avoid obstacles. Subsequently, several authors used parameter learning methods to fit the social forces model to observed crowd behavior (Helbing and Johansson, 2009; Johansson et al., 2007). Although the social forces model seems to perform well at simulating crowds, we found that the model has shortcomings in predicting the movements of individual pedestrians, particularly during evasive maneuvers. Pandey and Alami (2009) and Kirby et al. (2009) implement predefined social rules on a robot that operates in an environment populated by pedestrians. Similarly, Müller et al. (2008) proposed a method that allows a mobile robot to follow people that walk in the same direction as the robot. Lerner et al. (2007) infer a database of navigation rules from video data. Optimization models cast pedestrians as utility-optimizing agents that minimize a cost function comprising relevant properties of human navigation (Arechavaleta et al., 2008; Hoogendoorn and Bovy, 2003; Mombaur et al., 2010; Pham et al., 2007). There are methods that minimize the walking discomfort in terms of accelerations and distances to other pedestrians (Hoogendoorn and Bovy, 2003), maximize smoothness of the trajectory (Pham et al., 2007), or minimize the derivative of the curvature (Arechavaleta et al., 2008). Mombaur et al. (2010) present an optimization model that allows a humanoid robot to imitate human-like trajectories. Some of the abovementioned optimization approaches adapt the parameters of the models such that the resulting trajectories resemble training examples, which is known as inverse optimal control. In contrast to these non-probabilistic methods, we capture the stochasticity of human navigation behavior by modeling the behavior in terms of a probability distribution over the agents' trajectories. This also allows our method to learn from non-optimal demonstrations.

Trautman and Krause (2010) demonstrate that mobile robot navigation fails in densely populated environments unless the robot takes into account the interaction between itself and the humans. In contrast to many other approaches that model agents independently of the others, their proposed method takes into account mutual interaction to plan composite trajectories. In Trautman et al. (2013), the authors use their method to navigate a robot through a crowded cafeteria. Similarly, van den Berg et al. (2009) present an approach to reciprocal collision avoidance that allows a set of mobile robots to navigate without collisions. The social forces model (Helbing and Molnar, 1995) as well as the data-driven approach presented by Lerner et al. (2007) implicitly model cooperative navigation behavior. We explicitly model cooperative behavior by jointly predicting the trajectories of all interacting agents.

The abovementioned approaches aim at modeling the navigation behavior of humans and can therefore be used to foster efficient and socially compliant mobile robot navigation in populated environments. Several reactive collision avoidance methods were successfully applied to mobile robot navigation in crowded or dynamic environments. Such methods include the dynamic window approach by Fox et al. (1997), the velocity obstacles by Fiorini and Shillert (1998) as well as its extension to multiple obstacles, the reciprocal velocity obstacles (RVO) (van den Berg et al., 2009). To achieve more human-like behavior, Guy et al. (2010) extend RVO by introducing response and observation time to other agents. Whereas the approaches described above seek to avoid dynamic obstacles such as pedestrians, they do not consider human predictive abilities, which sometimes results in unnatural robot movements.

Our model considers trajectories in distinct homotopy classes, which result from the different choices of how to bypass obstacles or other agents. Other authors also investigated methods to compute homotopically distinct paths for mobile robot navigation. Bhattacharya et al. (2012) perform an A$^*$ search on an arbitrary graph representation of the environment that they augment with the $H$-signature to capture topological information. However, their graph may contain multiple paths to the goal of the same homotopy class. To lower the computational burden, we compute an abstract graph in which each path corresponds to a unique homotopy class. Demyen and Buro (2006) propose a method for efficient triangulation-based path planning that searches an abstract graph that represents the environment. Similar to our work, the resulting path through this graph is then mapped back to a trajectory in the original 2D environment. Whereas Demyen and Buro (2006) assume a polygonal representation of the environment, we allow arbitrary obstacles on grid maps, which enables us to incorporate online real-world sensor data. Vernaza et al. (2012) point out that the vector of winding angles around each obstacle is invariant for all trajectories of a given homotopy class. We also use the vector of winding angles to compute a unique identifier that fully describes the homotopy class of a composite trajectory, which is useful during navigation.

For mobile robot navigation we utilize the model of interactive navigation behavior that we present in this paper. To find the most likely trajectory in each homotopy class, we apply gradient-based optimization. Similarly, many authors presented approaches to find optimal trajectories with respect to a given cost function in the context of path planning for mobile robots. For instance, Sprunk et al. (2011) use a spline-based representation of the trajectories and optimize the corresponding control points to find time-optimized, curvature continuous trajectories that obey acceleration and velocity constraints. Similarly, Gulati et al. (2009) optimize trajectories for an assistive mobile robot with respect to user comfort. Ratliff et al. (2009a) present a general framework for trajectory optimization, which they apply to high-dimensional motion planning for robots. It is well-known that such gradient-based optimization methods often fail to find globally optimal solutions since they are prone to get stuck in local minima. Kalakrishnan et al. (2011) therefore propose to use stochastic trajectory optimization to overcome these local minima. However, large state spaces due to complex settings make it infeasible to efficiently find globally optimal solutions by uniformly sampling trajectories. In contrast to that, our model explores the state space by simultaneously searching regions that belong to different homotopy classes, which each correspond to a local minimum.

## 3 Modeling Human Cooperative Navigation Behavior

The objective of our approach is to learn a model of human cooperative navigation behavior from observed trajectories. We model the decision-making process that ultimately leads to the observed trajectories of the agents as a joint probability distribution over the trajectories of all the agents. We assume that the observed trajectories are samples drawn from this distribution. We furthermore assume that the probability distribution depends on cartain features of the trajectories. Our goal is to find the probability distribution that explains the observed trajectories in terms of feature expectations. To this end, we apply the principle of maximum entropy, which we discuss in the following.

### 3.1 The Principle of Maximum Entropy and Feature Matching

In general, the problem of learning from demonstration is to find a model that explains the observed demonstrations and that generalizes to new situations. If we can model the observed behavior in terms of a probability distribution, learning translates to finding the distribution from which the observed samples $\mathbf{x}_k \in \mathcal{D}$ are drawn. To capture the relevant properties of the behavior, we define a feature vector

$$\mathbf{f} : \mathcal{X} \to \mathbb{R}^n \tag{1}$$

that maps states $\mathbf{x} \in \mathcal{X}$ to a vector of real values. This allows us to compute empirical feature values $\mathbf{f}_\mathcal{D}$ of the demonstrations

$$\mathbf{f}_\mathcal{D} = \frac{1}{|\mathcal{D}|} \sum_{\mathbf{x}_k \in \mathcal{D}} \mathbf{f}(\mathbf{x}_k), \tag{2}$$

that encode the properties of the observed behavior we want to learn. Following Abbeel and Ng (2004), we aim to find the distribution $p(\mathbf{x})$ that matches these empirical feature values in expectation:

$$\mathbb{E}_{p(\mathbf{x})}[\mathbf{f}(\mathbf{x})] = \mathbf{f}_\mathcal{D}. \tag{3}$$

In general, however, there is not a unique distribution that matches the features. Ziebart et al. (2008) resolve this ambiguity by applying the principle of maximum entropy (Jaynes, 1978), which states that the distribution with the highest entropy represents the given information best since it does not favor any particular outcome besides the observed constraints.

Following Ziebart et al. (2008), we are interested in the distribution that matches the feature expectations, as given in Equation (3), without implying any further assumptions. In this section, we outline how their approach can be applied to continuous spaces. The principle of maximum entropy states that the desired distribution maximizes the differential entropy

$$\operatorname*{argmax}_p H(p) = \operatorname*{argmax}_p \int_{\mathbf{x}} -p(\mathbf{x}) \log p(\mathbf{x}) d\mathbf{x}, \tag{4}$$

subject to the constraint

$$\int_{\mathbf{x}} p(\mathbf{x})\, d\mathbf{x} = 1, \tag{5}$$

that enforces the distribution to integrate to one, and subject to the constraints

$$\forall i\; f_{i\mathcal{D}} = \mathbb{E}_{p(\mathbf{x})}[f_i(\mathbf{x})] = \int_{\mathbf{x}} p(\mathbf{x}) f_i(\mathbf{x}) d\mathbf{x} \tag{6}$$

that enforce the empirical features values to match the expected feature values for all features $f_i$. Introducing Lagrangian multipliers $\alpha$ and $\theta_i$ for these constraints yields the maximization problem

$$p^\star, \alpha^\star, \boldsymbol{\theta}^\star = \operatorname*{argmax}_{p,\alpha,\boldsymbol{\theta}} h(p, \alpha, \boldsymbol{\theta}), \tag{7}$$

where

$$\begin{aligned} h(p,\alpha,\boldsymbol{\theta}) = & \int_{\mathbf{x}} -p(\mathbf{x}) \log p(\mathbf{x})\, d\mathbf{x} \\ & - \alpha \left( \int_{\mathbf{x}} p(\mathbf{x})\, d\mathbf{x} - 1 \right) - \sum_i \theta_i \left( \int_{\mathbf{x}} p(\mathbf{x}) f_i(\mathbf{x})\, d\mathbf{x} - f_{i\mathcal{D}} \right). \end{aligned} \tag{8}$$

Applying the Euler-Lagrange equation from the calculus of variations (see (Fox, 1987, Sec. 1.4)) to Equation (8) implies that the probability distribution $p^\star(\mathbf{x})$ has the structure

$$p_{\boldsymbol{\theta}}(\mathbf{x}) = \frac{1}{Z(\boldsymbol{\theta})} \exp(-\boldsymbol{\theta}^T \mathbf{f}(\mathbf{x})), \tag{9}$$

where $Z(\boldsymbol{\theta})$ is a normalization factor to satisfy Equation (5). Thus, the structure of the distribution that maximizes entropy under the constraint of feature matching depends only on the features. However, the parameter vector $\boldsymbol{\theta}^\star$ depends on the training samples $\mathbf{x}_k \in \mathcal{D}$. Unfortunately, it is not feasible to compute $\boldsymbol{\theta}^\star$ analytically. However, we can apply gradient-based optimization techniques to determine $\boldsymbol{\theta}^\star$. The gradient is given by the derivative of Equation (8) with respect to the parameter vector:

$$\frac{\partial}{\partial \boldsymbol{\theta}} h(p, \alpha, \boldsymbol{\theta}) = \mathbb{E}_{p_{\boldsymbol{\theta}}(\mathbf{x})}[\mathbf{f}(\mathbf{x})] - \mathbf{f}_{\mathcal{D}}. \tag{10}$$

There is also a different point of view that leads to the same result. If we assume an exponential family distribution, as given in Equation (9), the log-likelihood of the observed behavior $\mathcal{D}$ is given by

$$L_{p_{\boldsymbol{\theta}}}(\mathcal{D}) = \log \frac{1}{Z(\boldsymbol{\theta})} \exp(-\boldsymbol{\theta}^T \mathbf{f}_{\mathcal{D}}), \tag{11}$$

and its derivative with respect to $\boldsymbol{\theta}$ is given by

$$\frac{\partial}{\partial \boldsymbol{\theta}} L_{p_{\boldsymbol{\theta}}}(\mathcal{D}) = \int_{\mathbf{x}} p_{\boldsymbol{\theta}}(\mathbf{x}) \mathbf{f}(\mathbf{x})\, d\mathbf{x} - \mathbf{f}_{\mathcal{D}} = \mathbb{E}_{p_{\boldsymbol{\theta}}(\mathbf{x})}[\mathbf{f}(\mathbf{x})] - \mathbf{f}_{\mathcal{D}}.$$

Consequently, the problem of finding the maximum entropy distribution subject to feature matching is equivalent to maximizing the likelihood of the training data when assuming an exponential family distribution (Jaynes, 1978).

To summarize, our goal is to find the distribution that matches, in expectation, the feature values of a set of demonstrations. By applying the principle of maximum entropy it follows that this distribution is an exponential family distribution Equation (9). Therefore, finding the desired distribution translates to computing the parameter vector $\boldsymbol{\theta}^\star$ that leads to feature matching. Computing $\boldsymbol{\theta}^\star$ analytically is not feasible, but we can compute the gradient with respect to these parameters and, consequently, apply gradient-based optimization.

## 3.2 Modeling Continuous Navigation Decisions

We apply the principle of maximum entropy and feature matching as introduced above to learn the navigation behavior of interacting agents by observing a set $\mathcal{D}$ of their trajectories $\mathbf{x} \in \mathcal{X}$. To this end, we define a feature vector $\mathbf{f}$ that captures the properties of the navigation behavior that we want to take into account. We are interested in the distribution $p(\mathbf{x})$ over the trajectories that matches the observations $\mathcal{D}$ in terms of these features, as expressed in Equation (3), which we restate here for convenience:

$$\mathbb{E}_{p(\mathbf{x})}[\mathbf{f}(\mathbf{x})] = \mathbf{f}_{\mathcal{D}}. \tag{12}$$

Applying the principle of maximum entropy, which we discussed in Section 3.1, leads to an exponential family distribution

$$p_{\boldsymbol{\theta}}(\mathbf{x}) = \frac{1}{Z(\boldsymbol{\theta})} \exp(-\boldsymbol{\theta}^T \mathbf{f}(\mathbf{x})), \tag{13}$$

where $Z(\boldsymbol{\theta})$ is a normalization factor. The term

$$\boldsymbol{\theta}^T \mathbf{f}(\mathbf{x}) = \sum_i \theta_i f_i(\mathbf{x}) \tag{14}$$

in the exponent can be interpreted as a cost function that depends on a weighted sum of feature values. Hence, our model assumes that the agents are exponentially more likely to choose trajectories with lower cost.

Learning the navigation behavior in this context boils down to finding the feature weights $\boldsymbol{\theta}$ that lead to feature matching, as expressed in Equation (12). As described in the previous section, we find these feature weights using gradient-based optimization, where the gradient is given by Equation (10). After convergence of the optimization process, the gradient vanishes and thus the probability distribution $p_{\boldsymbol{\theta}}(\mathbf{x})$ matches the empirical feature values.

In the following, we present a continuous representation of the agents' trajectories and introduce features that capture physical properties of the trajectories such as velocities and accelerations. We furthermore propose a method to compute the feature expectations with respect to the continuous probability distributions that are necessary to find the parameters of the probability distribution.

### 3.2.1 Spline-Based Representation

We represent the planar movements of an agent $a_i \in A$ in terms of its trajectory $x^{a_i}(t)$ that continuously maps time $t$ to the continuous configuration space $\mathbb{R}^2$ of the agent at time $t$. This function continuously defines the position of the agent $a_i$ over time. In general, the space of continuous trajectories in $\mathbb{R}^2$ has infinite dimensionality. To represent the trajectories in a finite-dimensional space, we use cubic splines. Specifically, for each interval $[t_j, t_{j+1}]$ with $t_0 \leq t_1 \leq \cdots \leq t_m$ the restriction of $x^{a_i}(t)$ to $[t_j, t_{j+1}]$ is a two-dimensional polynomial of degree three. This allows us to parameterize the trajectories by a finite set of control points $x^{a_i}(t_j)$ and $\dot{x}^{a_i}(t_j)$ for each $j \in \{0, \ldots, m\}$, which fully specifies the trajectory. In our representation, two consecutive spline segments "share" control points. Hence, we have equal position and velocity at the intersections. We can furthermore efficiently compute the positions $x^{a_i}(t)$, the velocities $\dot{x}^{a_i}(t)$, and the accelerations $\ddot{x}^{a_i}(t)$ at each time step in closed form. To represent the joint behavior of all agents $a_i \in A$, we consider composite trajectories

$$\mathbf{x} = (x^{a_1}(t), \ldots, x^{a_n}(t)) \in \mathcal{X}. \tag{15}$$

A composite trajectory encodes the interactive navigation behavior in a certain situation, i. e., how the agents evade each other. Note, that our model considers probability distributions over these composite trajectories. This captures the cooperative, joint behavior of all agents involved in the interaction process.

### 3.2.2 Features

According to recent studies (Hoogendoorn and Bovy, 2003), pedestrians seem to consider various criteria for navigation, for example time of travel, velocities, accelerations, and proximity to other pedestrians. We model these criteria in terms of features

$$f : \mathcal{X} \mapsto \mathbb{R} \tag{16}$$

that map composite trajectories $\mathbf{x} \in \mathcal{X}$ to real numbers $f(\mathbf{x})$. In particular, we propose to use the following features that model an intent to reach the target positions energy efficiently, taking into account velocities and clearances when avoiding obstacles and other agents.

**Time** The incentive of a group $A$ of pedestrians to reach a certain target position as fast as possible (Mombaur et al., 2010) is captured by the feature

$$f^A_{\text{time}}(\mathbf{x}) = \sum_{a \in A} \int_t 1 \; dt. \tag{17}$$

**Acceleration** Pedestrians typically aim to walk efficiently, avoiding unnecessary accelerations (Hoogendoorn and Bovy, 2003; Mombaur et al., 2010). Integrating the squared acceleration over the trajectory yields the feature

$$f^A_{\text{acceleration}}(\mathbf{x}) = \sum_{a \in A} \int_t \|\ddot{x}^a(t)\|^2 \; dt. \tag{18}$$

**Velocity** Pedestrians tend to walk at a certain velocity that is uncomfortable to exceed (Helbing and Molnar, 1995). We therefore use the feature

$$f^A_{\text{velocity}}(\mathbf{x}) = \sum_{a \in A} \int_t \|\dot{x}^a(t)\|^2 \; dt \tag{19}$$

that integrates the squared velocity over the trajectories.

**Clearance to other agents** Pedestrians tend to evade other pedestrians. We assume that the evasive maneuvers depend on the distances between the agents, yielding the feature

$$f^{\text{phys}}_{\text{distance}}(\mathbf{x}) = \sum_{a \in A} \sum_{b \neq a} \int_t \frac{1}{\|x^a(t) - x^b(t)\|^2} \, dt. \tag{20}$$

**Collision avoidance with respect to static obstacles** Finally, pedestrians avoid walls and other static obstacles, leading to the feature

$$f^{A}_{\text{obstacle}}(\mathbf{x}) = \sum_{a \in A} \int_t \frac{1}{\|x^a(t) - o^a_{\text{closest}}(t)\|^2} \, dt, \tag{21}$$

where $o^a_{\text{closest}}$ is the position of the closest obstacle to agent $a$ at time $t$.

### 3.2.3 Computing Feature Expectations

As discussed in Section 3.1, learning the model parameters $\boldsymbol{\theta}$ requires computing the feature expectations $\mathbb{E}_{p_{\boldsymbol{\theta}}(\mathbf{x})}[\mathbf{f}(\mathbf{x})]$ of the resulting probability distribution $p_{\boldsymbol{\theta}}(\mathbf{x})$. In general, however, inference about distributions over continuous trajectories is not analytically tractable. Monte Carlo sampling methods yet provide means to approximate the expectations using a set of sample trajectories drawn from the distribution. In particular, Markov chain Monte Carlo (MCMC) methods (Bishop, 2006) allow us to obtain samples from high-dimensional distributions. These methods aim to explore the state space by constructing a Markov chain whose equilibrium distribution is the target distribution.

Most notably, the widely-used Metropolis-Hastings algorithm (Hastings, 1970) generates a Markov chain in the state space using a proposal distribution and a criterion to accept or reject the proposed steps. This proposal distribution and the resulting acceptance rate, however, have a dramatic effect on the mixing time of the algorithm, e. g., the number of steps after which the distribution of the samples can be considered to be close to the target distribution. In general, it is difficult to design a proposal distribution that leads to satisfactory mixing. As a result, efficient sampling from complex high-dimensional distributions is often not tractable in practice.

Our approach exploits the structure of the distributions over composite trajectories to enable efficient sampling. First, the navigation behavior of physical agents shapes the trajectories according to certain properties such as smoothness and goal-directed navigation, which are captured by the features. As a result, the distributions over the composite trajectories of the same homotopy class are highly peaked. Exploiting the gradient of the probability densities allows us to guide the sampling process towards these regions of high probability. To this end, assuming that the physical features are differentiable with respect to the parameterization of the trajectories allows us to compute the gradient of the density $p_\psi(\mathbf{x})$.

In particular, we use the Hybrid Monte Carlo algorithm (Duane et al., 1987), which takes into account the gradient of the density to sample from the distributions $p_\psi(\mathbf{x})$. The algorithm considers an extended target density $p_\psi(\mathbf{x}, \mathbf{u})$ to simulate a fictitious physical system, where $\mathbf{u} \in \mathbb{R}^n$ are auxiliary momentum variables. The method constructs a Markov chain by alternating Hamiltonian dynamical updates and updates of the auxiliary variables, utilizing the gradient of the density $p_\psi(\mathbf{x})$ with respect to $\mathbf{x}$. After performing a number of these "frog leaps", Hybrid Monte Carlo relies on the Metropolis-Hastings algorithm to accept the candidate samples $\mathbf{x}^\star$ and $\mathbf{u}^\star$ with probability

$$\min\Big(1, \frac{\tilde{p}_\psi(\mathbf{x}^\star)\exp(-\frac{1}{2}\mathbf{u}^{\star T}\mathbf{u}^\star)}{\tilde{p}_\psi(\mathbf{x}^{(\tau)})\exp(-\frac{1}{2}\mathbf{u}^{\tau T}\mathbf{u}^\tau)}\Big), \tag{22}$$

where the normalizer $Z_p$ in the distribution $p_\psi(\mathbf{x}) = \tilde{p}_\psi(\mathbf{x})/Z_p$ vanishes.

In theory, the resulting Markov chain explores the entire state space $\mathcal{X}$, and therefore enables sampling from the distribution $p_{\boldsymbol{\theta}}(\mathbf{x})$. In practice, however, the mixing time when sampling from the kind of distributions over composite trajectories that we are considering can be obstructive. This follows from the observation that the distributions over composite trajectories are highly peaked. The regions of smooth, collision-free trajectories are surrounded by regions of very low probability, for example where two agents get close to each other. Thus, in practice the Markov chain is unlikely to cover the whole space of composite trajectories in reasonable time. In the next section, Section 3.3, we will present a model of human navigation behavior that naturally captures continuous as well as discrete navigation decisions and that also allows efficient sampling for learning its parameters.

## 3.3 Modeling Discrete Navigation Decisions

In the previous section, we presented an approach to model continuous navigation behavior in terms of features such as velocities, accelerations, and clearances when passing obstacles and other agents. In addition to that, the decision-making process of interacting agents that ultimately leads to the observed trajectories often also comprises discrete decisions. The model we present in this section allows us to model both, the discrete as well as the continuous navigation decisions. For instance our model allows us to learn cultural preferences of passing on a specific side, or how acceptable it is to split groups of people that belong together.

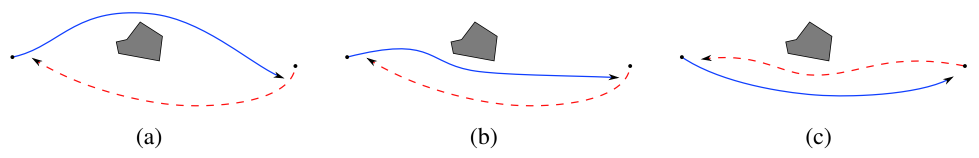


Figure 2: Example of three homotopically distinct composite trajectories in the same situation. (a vs. b): the blue agent passes the solid obstacle on opposite sides. (b vs. c): the two agents pass each other on different sides.

In the following, we first review the concept of homotopic trajectories. We then present a method to model the discrete and the continuous decision-making process of interacting agents in terms of a mixture distribution. We introduce a set of features to capture the discrete navigation decisions of the agents. Finally, we compute the expected feature values of the discrete probability distribution, which are needed for learning the model parameters.

### 3.3.1 Mixture Distribution

Figure 2 illustrates three distinct composite trajectories that capture the possible interaction of two agents from the same initial situation. The composite trajectories depicted in (a) and (b) differ in one of the agent's decision to bypass the static obstacle on different sides. In contrast to that, according to the trajectories depicted in (b) and (c), the agents pass each other on distinct sides. We can capture these different possible ways to bypass static obstacles and other agents with the concept of homotopy. Two composite trajectories are homotopic if and only if they can be continuously transformed into each other without any collisions. Hence, the depicted trajectories (a), (b), and (c) are non-homotopic. Consequently, the agents' discrete decisions of bypassing other agents and static obstacles on the left side or on the right side partition the space $\mathcal{X}$ of composite trajectories into homotopy classes $\psi \in \Psi$.

We model these discrete decisions using a probability distribution over the homotopy classes of the composite trajectories. Specifically, as illustrated in Figure 3, we use a mixture distribution that comprises a discrete probability distribution $P(\psi)$ over the homotopy classes $\psi \in \Psi$. For each of the homotopy classes $\psi \in \Psi$, the mixture distribution models the agents' continuous navigation decisions in terms of a continuous probability distribution $p_\psi(\mathbf{x})$, where $\mathbf{x} \in \psi$, as discussed in Section 3.2. We assume that the continuous navigation behavior is independent of the homotopy classes, which means that the feature weights $\boldsymbol{\theta}$ that give rise to the continuous distributions $p_\psi(\mathbf{x})$ are the same for all the homotopy classes $\psi$.

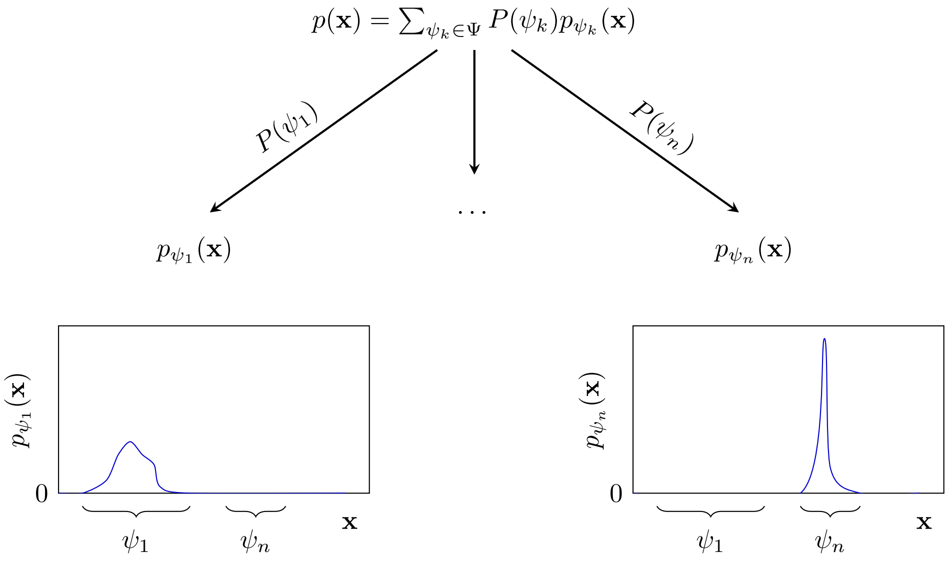


Figure 3: Mixture distribution to model the agents' discrete and continuous navigation behavior. The mixture density function $p(\mathbf{x})$ is given by a weighted sum of probability density functions $p_\psi(\mathbf{x})$ that capture the agents' continuous navigation decisions conditioned on the homotopy classes $\psi \in \Psi$. The weights assign a probability $P(\psi)$ to each homotopy class $\psi$ of the space of composite trajectories. The figure illustrates the high-dimensional space $\mathcal{X}$ of composite trajectories in one dimension.

For learning the feature weights $\boldsymbol{\theta}$ we need to compute feature expectations with respect to the distribution $p_\psi(\mathbf{x})$ of a given homotopy class $\psi$ only, as opposed to the space of all composite trajectories. This enables Hybrid Monte Carlo to efficiently explore the space without the need to traversing regions of low probability that separate the homotopy classes. In practice, we initialize a Markov chain for each considered homotopy class at its most likely composite trajectory, which we can compute efficiently using gradient-based optimization.

To learn the discrete navigation behavior from demonstrations, our goal is to find the probability distribution $P(\psi)$ that induces the agents' discrete decisions in terms of features

$$\mathbf{f}^{\text{hom}} : \Psi \mapsto \mathbb{R} \tag{23}$$

of the homotopy classes. Hence, we want

$$\mathbb{E}_P[\mathbf{f}^{\text{hom}}] = \mathbf{f}^{\text{hom}}_{\mathcal{D}} = \sum_{\mathbf{x} \in \mathcal{D}} \frac{\mathbf{f}^{\text{hom}}(\psi_{\mathbf{x}})}{|\mathcal{D}|}, \tag{24}$$

where $\mathbf{f}_{\mathcal{D}}^{\text{hom}}$ is the empirical feature value of the observed trajectories $\mathcal{D}$, and $\psi_{\mathbf{x}}$ is the homotopy class of $\mathbf{x}$. According to the principle of maximum entropy, which we discussed in Section 3.1, the distribution that best describes the observed behavior without implying any further assumptions is given by

$$P_{\boldsymbol{\theta}^{\text{hom}}}(\psi) = \frac{1}{Z(\boldsymbol{\theta}^{\text{hom}})} \exp(-\boldsymbol{\theta}^{\text{hom}^T} \mathbf{f}^{\text{hom}}(\psi)), \tag{25}$$

where $Z(\boldsymbol{\theta}^{\text{hom}})$ is a normalization factor. Likewise, we can apply gradient-based optimization to compute the feature weights $\boldsymbol{\theta}^{\text{hom}}$ based on the observed trajectories $\mathcal{D}$. According to the mixture distribution $p(\mathbf{x})$, the navigation behavior is captured by the probability distribution over the composite trajectories given by

$$p(\mathbf{x}) = P(\psi_{\mathbf{x}}) p_{\psi_{\mathbf{x}}}(\mathbf{x}). \tag{26}$$

As illustrated in Figure 3, the resulting generative model allows us to sample composite trajectories from the mixture distribution given in Equation (26) by means of ancestral sampling (Bishop, 2006).

### 3.3.2 Features

To represent important properties of natural navigation behavior in terms of homotopy classes, we use feature $\mathbf{f}^{\text{hom}}(\psi)$, that map homotopy classes $\psi$ to real values. To capture the sides on which each pair of agents passes each other, we compute the rotation of the vector $x^b(t) - x^a(t)$ for a pair of agents $a$ and $b$ along their trajectories, which we denote as $\omega_a^b$. For example, two agents that pass each other on the right yield a positive $\omega_a^b$, in contrast to a negative value for passing on the left. It is important to note that $\omega_a^b$ is an invariant for all composite trajectories of one homotopy class with fixed start- and goal positions. We describe the computation of $\omega_a^b$ in more detail in Section 4.2.1. In the following, we present the features we utilize to describe relevant properties of natural decision making during navigation.

**Passing left vs. passing right** To capture the decisions to avoid other agents on the left or on the right, we consider the feature

$$f_{\text{lr}}^{\text{hom}}(\psi) = \sum_{a \in A} \sum_{b \neq a} \omega_a^b. \tag{27}$$

**Group Behavior** Similarly, in case we are able to recognize group memberships of agents, the following feature indicates if an agent moves through such a group.

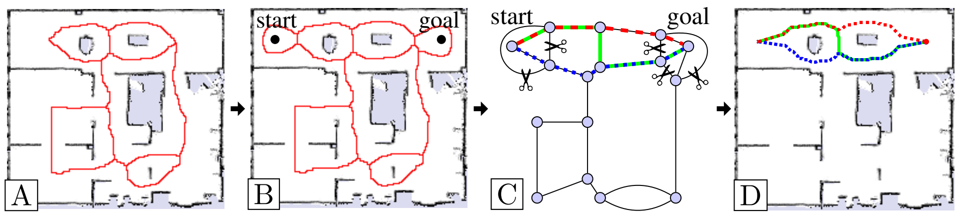


Figure 4: Overview of the proposed method to compute static homotopy classes. (A) Voronoi diagram (red) of an office environment. (B) We add Voronoi cells around the start and the goal location for a robust connection to the diagram. (C) Graph representation of the Voronoi diagram with connected vertices for start and goal location. Each path in this graph corresponds to a homotopically distinct path from start to goal. (D) Trajectories generated from the three shortest paths.

An agent that passes two members of a group on different sides moves through the corresponding group. Therefore, we have

$$f^{\text{hom}}_{\text{group}}(\psi) = \sum_{a \in A} |\{G \in \mathcal{G} \mid \exists b, c \in G : b, c \neq a \wedge \omega_a^b \omega_a^c < 0\}|,$$

where $\mathcal{G}$ is the set of groups of agents, which allows our approach to learn to which extent the agents avoid to move through groups.

**Most likely composite trajectory** Furthermore, we allow the features $f^{\text{hom}} : \Psi \mapsto \mathbb{R}$ to depend on the distribution over composite trajectories of the corresponding homotopy class. For example, the feature

$$f^{\text{hom}}_{\text{ml\_cost}}(\psi) = \min_{\mathbf{x} \in \psi} \boldsymbol{\theta}^T \mathbf{f}(\mathbf{x}) \tag{28}$$

captures the cost of the most likely composite trajectory $\mathbf{x}$ of homotopy class $\psi$, which allows the model to reason about the homotopy class the agents choose in terms of the cost of the composite trajectory that is most likely according to the learned distribution $p_\psi(\mathbf{x})$. Based on the results of our previous experiments (Kuderer et al., 2012), we assume that we can compute the most likely composite trajectory using gradient-based optimization techniques.

### 3.3.3 Computing Feature Expectations

Learning the model parameters $\boldsymbol{\theta}^{\text{hom}}$ to capture the agents' discrete navigation decisions requires computing the expected feature values

$$\mathbb{E}_{P(\psi)}[\mathbf{f}^{\text{hom}}] = \sum_{\psi \in \Psi} P(\psi) \mathbf{f}^{\text{hom}}(\psi) \tag{29}$$

with respect to the discrete probability distribution $P(\psi)$. However, summing over the entire space $\Psi$ of homotopy classes can be computationally prohibitory since the number of homotopy classes scales exponentially with the number of agents and static obstacles.

Fortunately, in most situations, only a few homotopy classes constitute the majority of the probability mass of the distribution $P(\psi)$. For example, homotopy classes that entail a long detour around obstacles might be highly unlikely and for this reason have only limited effect on the feature expectations. We therefore propose to approximate the expected feature values $\mathbb{E}_{P(\psi)}[\mathbf{f}^{\text{hom}}]$ based on a subset $\Psi'$ of the homotopy classes $\psi \in \Psi$.

Evaluating the feature values $\mathbf{f}^{\text{hom}}(\psi)$ of a given homotopy class $\psi$ requires computing a composite trajectory $\mathbf{x} \in \psi$ of that homotopy class $\psi$. In the following, we present a method to efficiently compute a set of homotopically distinct composite trajectories that are likely to constitute the majority of the probability.

In particular, to compute an initial guess for the distinct homotopy classes that arise from static obstacles, we utilize the concept of Voronoi diagrams. In general, the Voronoi diagram is defined as the set of points in free space to which the two closest obstacles have the same distance (Choset and Burdick, 2000). We compute a discretized form on an obstacle grid map bounded by occupied cells and represent it as a binary grid map $VD$ in which a cell

$$(x, y) \in \mathbb{N}^2 \tag{30}$$

either belongs to the Voronoi diagram or not, giving

$$VD(x, y) \in \{\text{true}, \text{false}\}. \tag{31}$$

Figure 4 (A) shows such a discretized Voronoi diagram over the obstacle map and depicts the cells for which $VD(x, y) = \text{true}$ in red. We also add dummy obstacles at the start and the goal position of the agents such that these positions become effectively enclosed by “bubbles” which are obstacle free by construction of the Voronoi diagram. This provides a robust way to connect these positions to the Voronoi diagram, see Figure 4 (B) vs. (A).

Based on this discretized representation of the Voronoi diagram, we build a graph that effectively captures the connectivity of the free space. In particular, each vertex in the graph corresponds to a “branching point” in the original Voronoi diagram, as illustrated in Figure 4 (C). In addition, we add vertices for the start and goal positions and connect them to the graph. Finally, we need to remove the edges that were introduced by the dummy obstacles, as illustrated by scissor symbols in Figure 4 (C). This process ensures that each simple path in the graph corresponds to one unique homotopy class. The resulting abstract graph substantially reduces the number of states compared to the original grid map representation of the Voronoi diagram, as shown in Figure 4 (C). Finally, we set the weights of the edges according to the length of the lines in the Voronoi diagram.

The graph grows linearly with the number of obstacles in the environment (Aurenhammer, 1991). This follows from viewing the Voronoi graph as a planar graph where the number of faces $l$ corresponds to the number of obstacles. Since each vertex in the Voronoi graph has a minimum degree of three, the sum over the degrees of all vertices $\sum_{v \in V} \deg(v)$ is at least three times the number of vertices $|V|$. Furthermore, any undirected graph satisfies

$$\sum_{v \in V} \deg(v) = 2|E|. \tag{32}$$

Hence, we have

$$2|E| \geq 3|V|. \tag{33}$$

Combining this with the Euler relation for planar graphs given by

$$|V| - |E| + l = 2 \tag{34}$$

leads to

$$|E| \leq 3l \tag{35}$$

and

$$|V| \leq 2l. \tag{36}$$

As a result, the number of edges and vertices is linear in the number of obstacles.

In the abstract graph, different paths correspond to different homotopy classes in the environment. Therefore, searching for the $k$ best homotopically different simple paths in this graph is equivalent to searching for the $k$ best simple paths. According to Katoh et al. (1982), the best known algorithm for this problem has a runtime complexity in

$$O(k(|E| + |V| \log |V|)), \tag{37}$$

which follows from the complexity of

$$O(|E| + |V| \log |V|) \tag{38}$$

of Dijkstra's algorithm (Dijkstra, 1959). As the number of vertices and edges in our graph depends linearly on the number $o$ of obstacles, it follows that our algorithms to extract the $k$ best homotopically different simple paths for one agent has a complexity

$$O(k(o \log o)). \tag{39}$$

See Kuderer et al. (2014) for further details on the construction of homotopically distinct navigation paths.

Figure 4 (C) shows the three best paths from the start to the goal position for one agent in the abstract graph. Figure 4 (D) shows the corresponding three best paths in the Voronoi diagram. We conduct these computations for each agent and convert the resulting paths to the spline-based trajectory representation. Thus, we can generate composite trajectories that are homotopically distinct. In practice, we only consider the homotopy classes that correspond to the $k$ shortest trajectories for each agent.

So far, we computed a set of homotopic distinct composite trajectories result from static obstacles. In the following, we show how to include also homotopy classes that result from the interaction of agents. We propose to consider different outcomes of an interaction between two agents only when agents are likely to interact in the first place, i.e., when they are close to each other at some point along their trajectories. In this way, interactions between agents that are highly unlikely are ignored.

For each composite trajectory $\mathbf{x}_{\text{init}}$, our algorithm identifies a potential evasive maneuver when two agents $a$ and $b$ come close to each other at some point along $\mathbf{x}_{\text{init}}$. For such a potential evasive maneuver, we want to reason about both possible outcomes, i.e., passing left or passing right. To this end, we compute a composite trajectory of $\mathbf{x}'$ in which the agents $a$ and $b$ pass on the other side compared to the original composite trajectory. Our algorithm repeatedly looks for such potential evasive maneuvers as described above until there are no unresolved collisions.

## 4 Socially Compliant Robot Navigation

A socially compliant robot that navigates in an environment populated by humans has to reason about future paths the humans are likely to take. The humans, however, will react to the actions of the robot themselves, which is why the robot has to adopt its behavior, which in turn will affect the humans. To break up this infinite loop, our approach jointly reasons about the trajectories that are likely to be followed by all the agents, including the robot itself. To do so, the robot incorporates the current poses of the humans and itself into the prediction process and implicitly "predicts" its own trajectory, which it then follows. The robot uses the learned policy to maintain a set of possible interactions from which it can select the trajectory that fits the current situation best. In this section, we integrate the model of natural pedestrian behavior, which we described in Section 3, into a consistent framework for mobile robot navigation. In particular, we present how we use learned behavior models for robot navigation and give details on how to maintain a set of homotopically different composite trajectories for the robot and nearby pedestrians. In addition, this section presents how to infer the goals of pedestrians and how we cope with online planning in larger environments.

## 4.1 Mobile Robot Navigation Using the Learned Behavior Model

The framework presented in Section 3 predicts the actions of the pedestrians in the vicinity of the robot in terms of a joint probability distribution over the composite trajectories of the robot and the pedestrians, taking into account the environment of the robot. In this work, we predict the future behavior of the pedestrians based on the current positions and velocities of the pedestrians. Taking into account the parts of the pedestrians' trajectories that the robot has observed so far, similar to Ziebart et al. (2009), may further improve the predictions. In this section, we will discuss how the robot can exploit this probability distribution to plan paths through populated environments.

A possible way to use the learned model for mobile robot navigation is to have the robot behave according to a composite trajectory sampled from this probability distribution. In this way, the robot imitates the learned behavior, including the observed stochasticity of the trajectories. Using ancestral sampling, the robot first samples a homotopy class $\psi$ from $P(\psi)$. The robot subsequently samples a composite trajectory $\mathbf{x}$ from the corresponding continuous probability distribution $p_{\psi}(\mathbf{x})$. Such a sample drawn from the joint probability distribution predicts the trajectories of nearby pedestrians. During navigation, we need to constantly update this prediction with the current situation, which means that we need to repeatedly sample new composite trajectories. However, sampling from the high-dimensional distributions is not feasible online during navigation.

In our experiments, we found a different way to integrate the learned model into the navigation framework to perform best. The robot executes at each time step its planned trajectory, which is captured in the most likely composite trajectory from the most likely homotopy class. In practice, we can efficiently optimize the parameters of an initial composite trajectory $\mathbf{x}$ with respect to its probability density

$$p_{\psi_{\mathbf{x}}}(\mathbf{x}) \propto \exp(-\boldsymbol{\theta}^T \mathbf{f}(\mathbf{x})), \tag{40}$$

using gradient-based optimization techniques. To this end, we compute the derivative of the feature vector $\mathbf{f}(\mathbf{x})$ with respect to the spline control points of $\mathbf{x}$ using a series of analytical and numerical differentiations. We found the optimization algorithm RPROP (Riedmiller and Braun, 1993) to perform best at optimizing the control points of composite trajectories. Previous experiments (Kuderer et al., 2012) suggest that gradient-based optimization indeed leads to the most likely state of the homotopy class $\psi(\mathbf{x})$. In this way, we compute the most likely trajectory in each relevant homotopy class. Thereby, we can compute the most likely homotopy class according to the discrete distribution

$$P(\psi) \propto \exp(-\boldsymbol{\theta}^{\mathrm{hom}^T} \mathbf{f}^{\mathrm{hom}}(\psi)). \tag{41}$$

## 4.2 Maintaining a Set of Non-Homotopic Composite Trajectories

The convergence time of optimization-based techniques typically decreases when the initial guess is already close to the optimum. Thus, during navigation, it is desirable to re-use previously optimized trajectories. In particular, from one planning cycle to the next the current position of the robot as well as the environment do not change substantially in most situations. Therefore, we propose to maintain a set $T$ of optimized, homotopically different trajectories during navigation.

However, whenever a new homotopy class emerges due to changes in the environment, we need to add the corresponding trajectory to the set $T$. When an obstacle vanishes, two homotopy classes merge. As a consequence, $T$ contains two homotopic composite trajectories, and we can discard one of them. In the following, we will describe how to uniquely identify the homotopy class of a given composite trajectory, which is needed for inserting trajectories corresponding to new homotopy classes as well as for discarding homotopic duplicates during navigation.

### 4.2.1 Identifying Homotopy Classes

As Vernaza et al. (2012) point out, the homotopy class of a trajectory can be represented as the vector of winding angles around the regions of interest, i.e., the obstacles in the environment. As illustrated in Figure 5, the winding angle $\omega_a^b(\mathbf{x})$ of a composite trajectory $\mathbf{x}$ is defined as the sum of infinitesimal angle differences $\Delta\omega$ of the trajectory of an agent $a$ to a representative point $b$ along the trajectory.

In the case of static obstacles, we take an arbitrary point inside an obstacle $b$ as the representative point to compute the winding angle. We consider the trajectory at discrete time steps and sum up the angle differences $\Delta\omega$. The step size of this discretization does not change the computed winding angle as long as it still captures the “winding direction”. We exploit this by adapting the step size dynamically as we walk along the trajectory to efficiently compute the winding angles.

In the case of pairs of agents, we apply the same approach. In contrast to static obstacles, however, the representative point moves along the trajectory of the second agent. In this case, we effectively compute the rotation of the vector $x^b(t) - x^a(t)$ for a pair of agents $a$ and $b$.

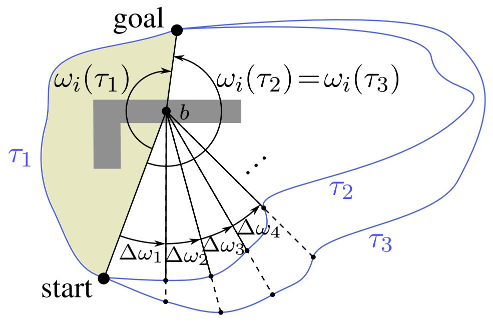


Figure 5: Computation of the winding angles with respect to obstacles. The figure shows three trajectories $\tau_1, \tau_2$ and $\tau_3$ that bypass the obstacle $b$. The infinitesimal angles $\Delta\omega$ sum up to the winding angle $\omega_i$ around the representative point of the obstacle. The two paths on the right yield the same winding angle $\omega_i(\tau_2) = \omega_i(\tau_3)$ in contrast to the path on the left.

Most importantly, composite trajectories that are homotopy equivalent yield the same winding angles for all obstacles and for all pairs of agents. Therefore, computing $\omega_a^b$ for all agents $a$ and for all other agents and obstacles $b$ yields a fingerprint

$$F(\mathbf{x}) = \langle \ldots, \omega_{a_i}^{b_i}(\mathbf{x}), \ldots \rangle_{a_i \neq b_i} \tag{42}$$

that allows us to describe and recognize the homotopy classes of composite trajectories. Note that we need to compute the angles each time the environment changes, since homotopy classes can emerge or vanish.

## 4.3 Inferring Target Positions

When using our model to predict the trajectories of the agents in new situations, the target positions of the agents might be unknown. Following (Ziebart et al., 2008), applying Bayes' theorem allows our model to reason about their target positions. After having observed the agents traveling from the composite positions $A$ to $B$ along the composite trajectory $\mathbf{x}_{A\to B}$, the probability that the agents proceed to composite target $C$ is given by

$$\begin{aligned} P_{\boldsymbol{\theta}}(C \mid \mathbf{x}_{A\to B}) &\propto p_\theta(\mathbf{x}_{A\to B} \mid C) P_{\boldsymbol{\theta}}(C) \\ &\propto \frac{\exp(-\boldsymbol{\theta}^T \mathbf{f}(\mathbf{x}_{A\to B})) \int_{\mathbf{x}\in\mathcal{X}_{B\to C}} \exp(-\boldsymbol{\theta}^T \mathbf{f}(\mathbf{x})) d\mathbf{x}}{\int_{\mathbf{x}\in\mathcal{X}_{A\to C}} \exp(-\boldsymbol{\theta}^T \mathbf{f}(\mathbf{x})) d\mathbf{x}} P_{\boldsymbol{\theta}}(C), \end{aligned} \tag{43}$$

where $\mathcal{X}_{A\to C}$ and $\mathcal{X}_{B\to C}$ refer to the set of all composite trajectories that lead the agents from $A$ to $C$, and from $B$ to $C$, respectively.

### 4.4 Path Planning in Large Environments

The time to compute the probability distribution over composite trajectories in a certain situation scales linearly with the travel time of the agents. Furthermore, the uncertainties in predicting cooperative behavior grow with the prediction horizon. Therefore, we employ our learned policy to represent the immediate, joint behavior of all the agents in the near future and represent more distant behavior by globally planned paths independently for each agent. This also enables socially compliant robot navigation in large environments.

Specifically, at time $t_0$, we evaluate the learned policy in the time interval $[t_0, t_0 + t_\mathrm{p}]$ and represent the behavior in the time interval $(t_0 + t_\mathrm{p}, t_\mathrm{end}]$ by a globally planned trajectory for each agent. More specifically, for a pedestrian $a$ that is detected by the robot for the first time, we estimate its target position and generate an initial trajectory $x^a_\mathrm{global}$ using A$^*$ search. The cost function for global path planning accounts for the time to reach the target and the distance to obstacles, which is a subset of the features used for the learned policy. Based on this global path, we set intermediate target positions given by $x^a_\mathrm{global}(t_0 + t_\mathrm{p})$ and use the learned policy to compute a probability distribution of the composite trajectory of all agents from their current positions to these intermediate target positions.

In each planning cycle, the robot updates the intermediate targets along this global path and executes the trajectory that corresponds to the most likely composite trajectory. Our current implementation allows the robot to replan at a frequency of 5 Hz, which enables the robot to quickly adapt its plan to changes in the environment. In our experiments, we typically set the planning horizon to $t_\mathrm{p} = 10\,\mathrm{s}$, which seems to enable natural predictive planning in most situations.

## 5 Experimental Evaluation

In this section, we present a thorough experimental evaluation of our approach. In Section 5.1, we evaluate the performance of our method to predict the trajectories of pedestrians based on a model that is learned using observed trajectories of the pedestrians. In Section 5.2, we present experiments of a mobile robot that uses our pedestrian model to navigate in populated environments.

### 5.1 Learning Pedestrian Navigation Behavior

We applied our approach to the problem of learning a model of pedestrian behavior. We considered two datasets of trajectories of interacting pedestrians. The first dataset, depicted in Figure 6, comprises one hour of interactions of four persons that we recorded using a motion capture system, leading to 81 individual composite

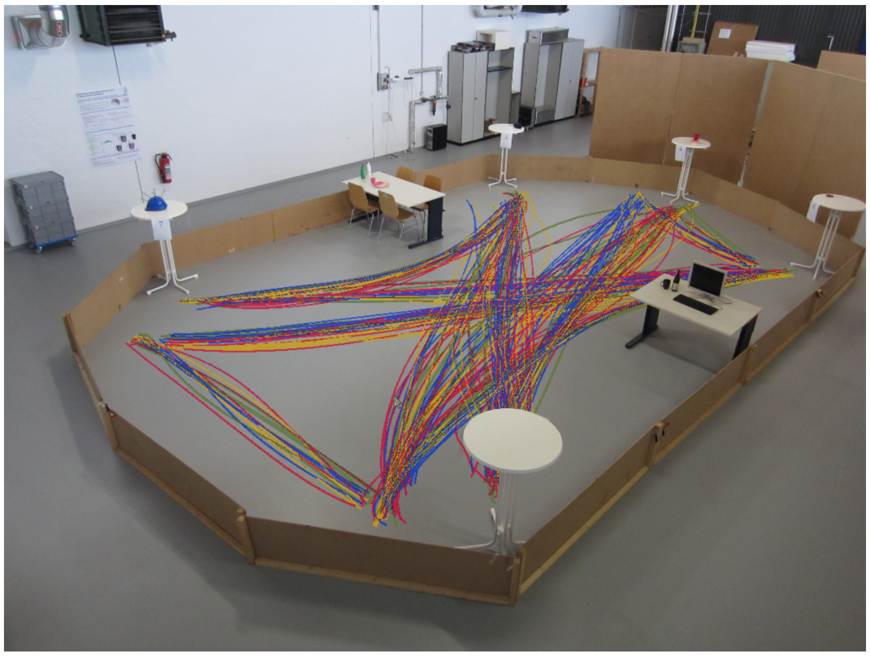


Figure 6: Trajectories observed during one hour of interactions of four persons in our test environment. The depicted area has a size of approximately 11 m $\times$ 8 m.

trajectories. To distract the persons from the navigation task, we made them read and memorize newspaper articles at different locations that were consecutively numbered. At a signal, they simultaneously walked to the subsequent positions, which repeatedly gave rise to situations where the participants had to evade each other. The second dataset (Pellegrini et al., 2009) comprises 12 minutes of trajectories of pedestrians interacting in a hotel entrance, leading to 71 composite trajectories with three to five pedestrians each.

### 5.1.1 Cross Validation

We conducted a five-fold cross validation on the aforementioned datasets to evaluate how well the models learned by our approach generalize to new situations. We compared our approach to the approach of Kuderer et al. (2012), the social forces algorithm by Helbing and Molnar (1995), and the reciprocal velocity obstacles (RVO) introduced by van den Berg et al. (2009). Figure 9 shows example trajectories of the different methods. We evaluated the evolution of the discrepancy between the feature expectations and the empirical feature values on the training sets while learning the model parameters of our approach. Figure 7 shows that our method is

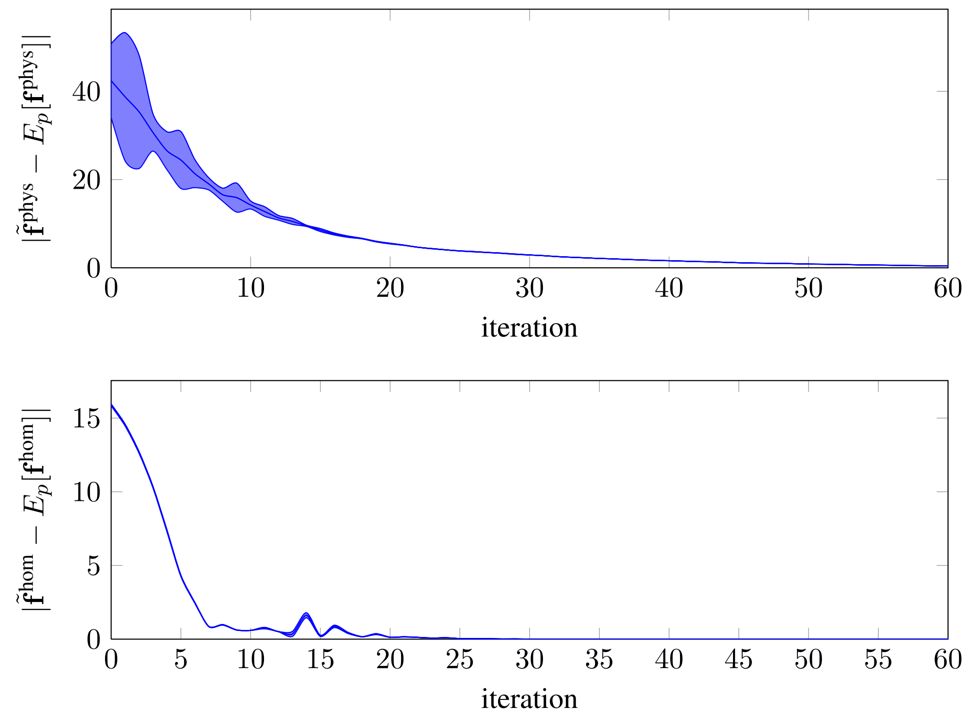
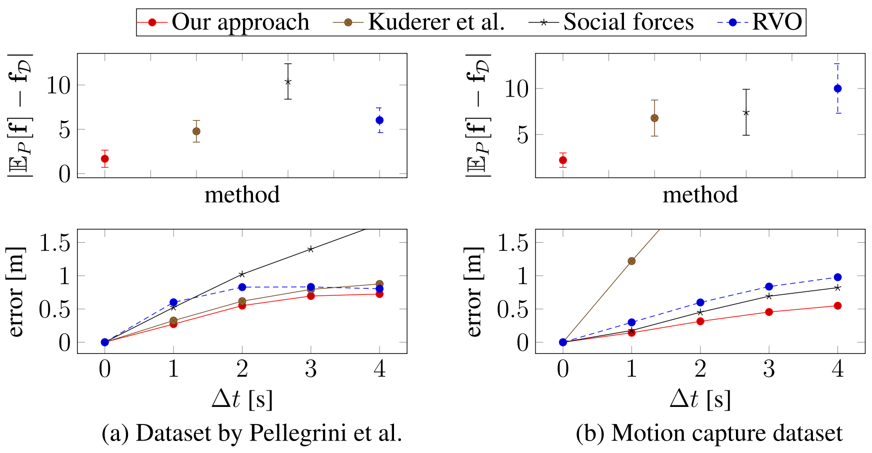


Figure 7: Evolution of the norm and variance over the five folds of the discrepancy between the feature expectations of the model and the empirical feature values while learning pedestrian navigation behavior. Top: Learning the physical properties of the trajectories. Bottom: Learning the discrete decisions that determine the homotopy classes of the composite trajectories. The gradient-based optimization method RPROP does not force a monotonic decrease in the value of the objective function, hence the bumps in the graph.

able to replicate the observed behavior in terms of the features. To allow for a fair comparison, we used the same set of features for all the methods. To optimize the parameters of the social forces method and RVO, we minimized the norm of the discrepancy between the feature values as induced by the methods and the empirical feature values using stochastic gradient descend. We additionally evaluated the parameters provided by Helbing and Molnar (1995) and Guy et al. (2012), which turned out to not perform better than the learned parameters. For all methods, we assumed that the target positions of the agents were the positions last observed in the datasets. The results of the cross validation, depicted in Figure 8, suggest that our method is able to capture human behavior more accurately than the other methods in terms of features and in terms of the prediction error in Euclidean distances. Note



(a) Dataset by Pellegrini et al.

(b) Motion capture dataset

Figure 8: Cross validation. The results suggest that our approach better captures pedestrian navigation behavior in terms of the features (top) and the prediction error in meters (bottom) compared to the approach of Kuderer et al. (2012), the social forces method (Helbing and Molnar, 1995), and reciprocal velocity obstacles (van den Berg et al., 2009). Left: Results on the dataset provided by Pellegrini et al. (2009). Right: Results on the dataset recorded using our motion capture system.

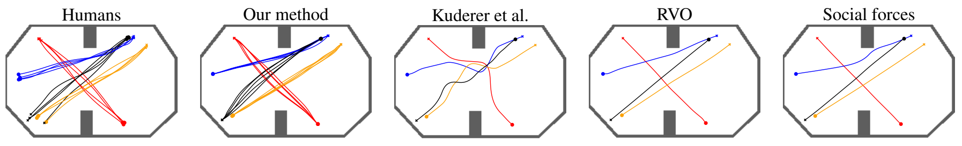


Figure 9: Trajectories of four pedestrians predicted by four different methods in the same situation. Humans: Observed trajectories recorded in the test environment shown in Figure 6. Our method: Samples drawn from the policy learned by our method replicate the stochasticity of the observed trajectories. Kuderer et al. (2012): The Dirac approximation favors samples from highly unlikely homotopy classes. RVO and social forces: Deterministic predictions.

that the comparison in terms of feature differences favors our particular method, since the other techniques do not explicitly optimize this value. The Euclidean distance metric, however, serves as a fair comparison.

### 5.1.2 Turing Test

A robot that navigates in a socially compliant way is required to plan trajectories in a human-like way. Our method is also applicable to create human-like navigation behavior for example to simulate agents in the context of computer graphics. Here, it is crucial that the generated trajectories are perceived as human-like. We carried out a Turing test to evaluate how human-like the behavior generated by our approach compares to other methods. We asked ten human subjects to distinguish recorded human behavior from behavior generated by one of the algorithms. We evaluated how well the subjects performed on a set of runs that was randomly drawn from recorded human demonstrations. We showed them animations of trajectories that were either recorded from the human demonstrations or from the prediction of one of the algorithms. In particular, we presented 40 runs to each of the human subjects, where the trajectories were equally drawn from the human demonstrations, from the predictions computed by our approach, by the approach of Kuderer et al. (2012), and Helbing and Molnar (1995). Figure 10 summarizes the results. The human subjects correctly identified $79\,\%$ of all the human demonstrations, but they mistook $68\,\%$ of the predictions of our approach, $40\,\%$ of the predictions of the approach of Kuderer et al. (2012), and $35\,\%$ of the predictions of the social forces algorithm for human behavior. The results of this Turing test indicate that the behavior induced by our approach is perceived to be significantly more human-like than the behavior induced by the other two methods according to a one-sided paired sample t-test at a 95% confidence level, where the measure is the percentage of the human subjects who perceived a particular trajectory as human-like.

## 5.2 Socially Compliant Mobile Robot Navigation

The goal of this section is to demonstrate that our approach allows a mobile robot to autonomously navigate in a socially compliant way. We conducted experiments where an autonomous robotic wheelchair interacted with real humans. The robot was equipped with laser range finders to perceive the environment including pedestrians. We localize the robot in a static map using laser-based Monte Carlo Localization (Thrun et al., 2000). To estimate the position and velocity of nearby pedestrians, we extract objects in the laser image that show typical characteristics of pedestrians. We then assign these observations to existing tracks of pedestrians, or add a new track if the observation cannot be assigned to any of the existing tracks. To prevent false positive detections, we disregard all laser observations in occupied space according to the static map. Our implementation allows the robot to observe its environment during tele-operation as well as during autonomous navigation only relying on on-board sensors. In our experiments, we assumed known target positions of the

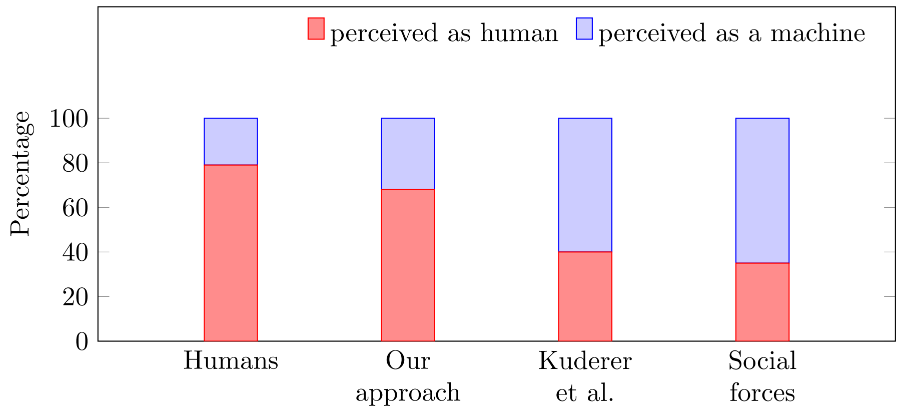


Figure 10: Turing test to evaluate whether the behaviors induced by our new approach, the approach of Kuderer et al. (2012), and the social forces model by Helbing and Molnar (1995) qualify as human. The results suggest that the behavior induced by our approach most resembles human behavior.

pedestrians.

### 5.2.1 Robot Navigation in the Presence of Static Obstacles

In a first experiment, the robotic wheelchair controlled by our method navigates through a static environment. Figure 11 visualizes the belief of the robot during the navigation task, i.e., the most likely trajectories for each homotopy class according to the learned policy. The most likely trajectory that is selected for navigation is shown in thick red in four images that correspond to consecutive time steps. The first image shows the trajectory from the start position of the robot to its target position in the static map of the environment. In the second image, after having traveled around the corner, the robot perceives a static obstacle in the middle of the corridor, which is not part of the static map. As a result of that, the robot starts reasoning about the resulting homotopy classes, i.e., trajectories that pass this obstacle on the top (left side) and about trajectories that pass this obstacle on the bottom (right side). The robot prefers to pass the obstacle on the bottom since this trajectory has higher likelihood according to the learned policy. The third and the fourth figure show the robot pursuing the selected trajectory moving to the target position. The gray lines show for each time step the trajectory driven by the robot so far.

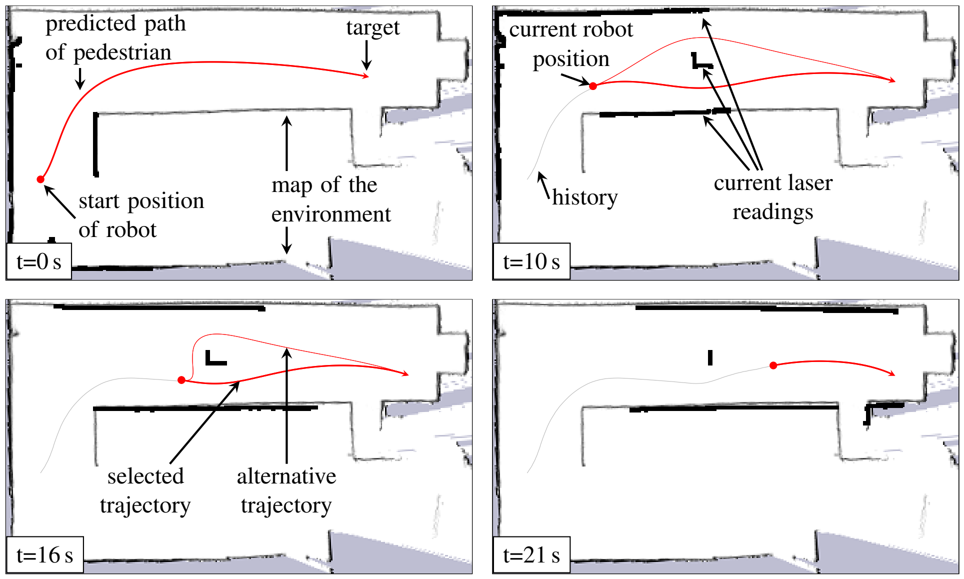


Figure 11: A robot controlled by our method navigates from its current position to its target position and avoids a static obstacle in the process. The red dot illustrates the robot's position at four individual time steps, and the gray line visualizes the trajectory that the robot has already driven. When the robot detects the static obstacle, it computes both possible homotopy classes and behaves according to the most likely trajectory, which is depicted as a thick red line.

### 5.2.2 Robot Navigation in the Presence of Static Obstacles and Humans

Figure 12 visualizes a second experiment, where the robot navigates through the same environment. In the third image, however, the robot suddenly encounters a pedestrian moving in the opposite direction. Assuming cooperative behavior, the robot starts reasoning about composite trajectories comprising itself and the pedestrian. In other words, the robot jointly reasons about its own trajectory and the trajectory of the pedestrian. As can be seen in the third and fourth image, the robot concludes that the pedestrian most likely passes the static obstacle on the top, since this joint behavior has the highest likelihood according to the learned policy. As a result, the robot chooses to pursue its original plan and passes the obstacle on the bottom. In the fourth image the robot has lost track of the pedestrian since it was occluded by the static obstacle. The gray line corresponds to the pedestrian's trajectory according to the perception of the robot.

Figure 13 visualizes a third experiment in the same environment. The robot also encounters the static obstacle and a pedestrian that moves in the opposite

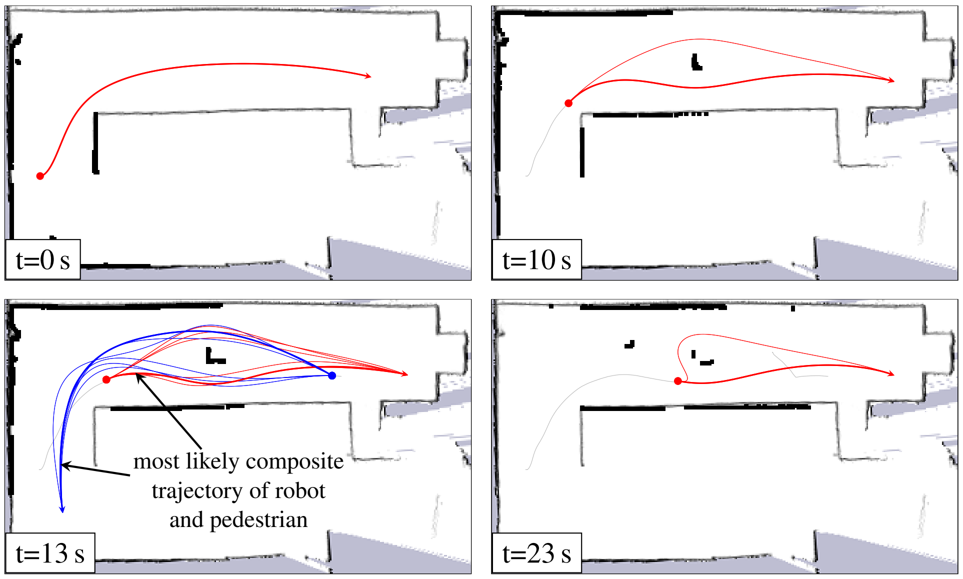


Figure 12: A robot controlled by our method avoids a static obstacle and a pedestrian. As soon as the robot detects the pedestrian, it computes the possible joint behavior of the pedestrian (blue) and the robot (red). The robot predicts for the pedestrian to evade the obstacle on the right. The pedestrian behaves according to the prediction of the robot and thus the robot proceeds to follow its plan.

direction. In the second image, the robot assigns highest likelihood to the homotopy class in which the pedestrian passes the obstacle on the top, similar to the previous experiment. However, in this experiment the pedestrian insists on passing the static obstacle on the bottom, which does not match the robot's prediction, as illustrated in the third figure. Since the robot constantly updates the probability distribution to the current state of the environment, it is able to adapt the prediction of the joint behavior. As a consequence, the robot changes its plan, decides to give way to the pedestrian and passes the obstacle on the top.

### 5.2.3 Cooperative Navigation in an Office Environment

We furthermore evaluated the ability of our approach to cooperatively navigate in an office environment in the presence of humans. The autonomous robotic wheelchair passes two pedestrians in a hallway. Figure 14 depicts the composite trajectory that the robotic wheelchair predicted to be most likely at four different timesteps during the encounter. First, the pedestrians walk side by side, blocking the corridor. In contrast, our method expects the humans to cooperatively engage in joint collision

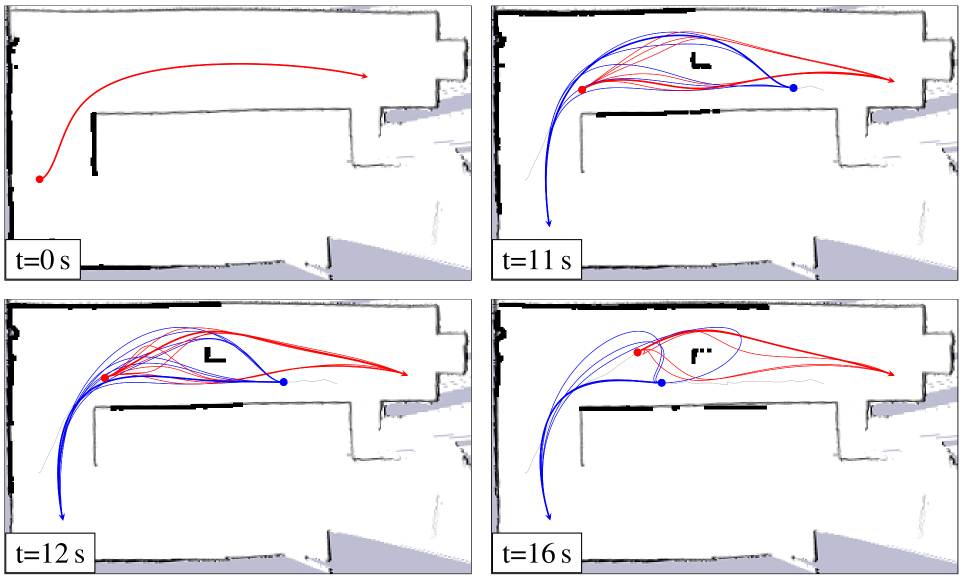


Figure 13: Example interaction of the robot (red) with a pedestrian (blue) in the presence of a static obstacle in the corridor. In the top left image, the robot predicts for the pedestrian to evade above the obstacle (thick blue line). However, the pedestrian insists on passing on the other side. Thus, the robot reevaluates the situation, as illustrated in the lower right image.

Table 1: Comparison of our approach to an A* planner.

| algorithm | min dist | avg velocity | % blocked |
|---|---|---|---|
| our approach | 0.34 | 0.56 | 0 |
| $A^*$, $r_{\text{min}} = 0.2$ | 0.16 | 0.49 | 1 |
| $A^*$ , $r_{\text{min}} = 0.8$ | 0.38 | 0.27 | 60 |

avoidance. During the encounter, the robot repeatedly computes the most likely cooperative interaction with the pedestrians, which allows the wheelchair to engage in natural joint collision avoidance. A traditional path planner would not be able to find a path to the target position in such a situation, since there is no valid path if the humans are considered as static obstacles, as suggested in the following set of experiments.

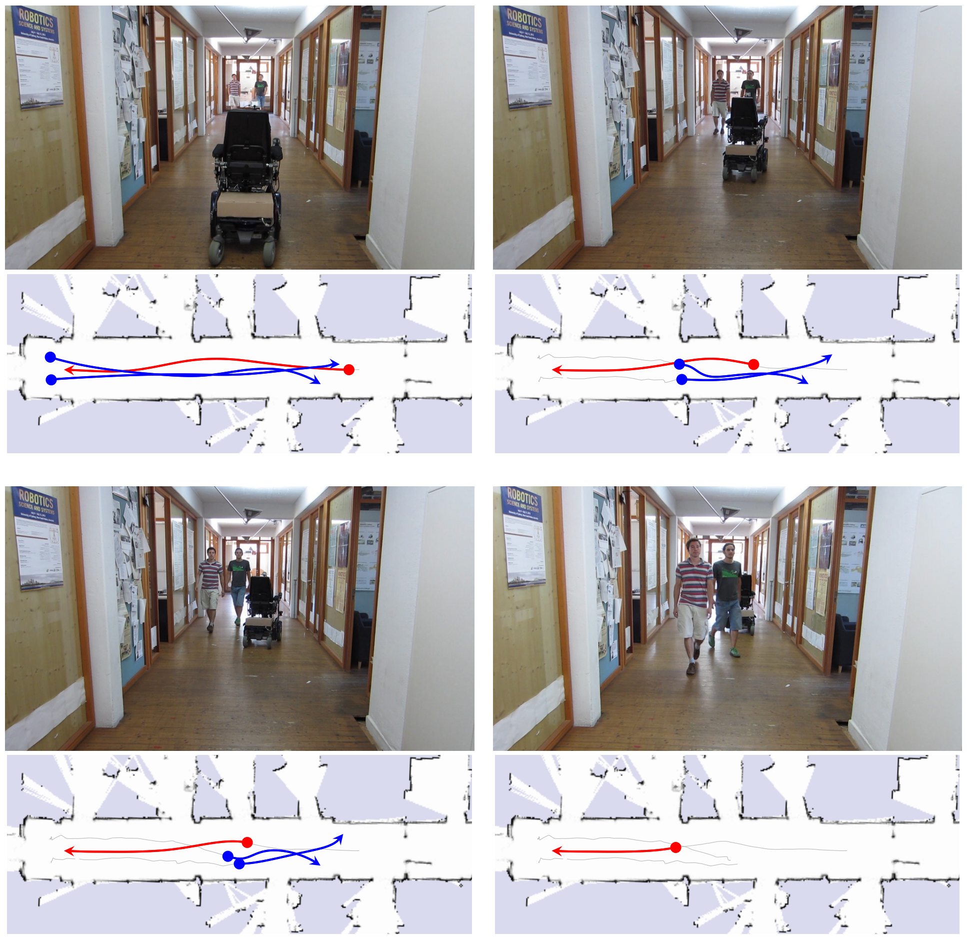


Figure 14: Autonomous mobile robot navigation in a realistic scenario in which a robotic wheelchair passes two pedestrians in a hallway using the approach presented in this paper. The bottom images depict the driven trajectories (gray) and the interaction of the robot (red) with the pedestrians (blue) that is considered to be most likely by the robot at different time steps. At first, the pedestrians block the hallway such that a traditional path planning algorithm would be unable to find a path to the target position. In contrast, our method expects the pedestrians to cooperatively engage in joint collision avoidance with the wheelchair, and therefore is able to find a path to the target position. Left: The robot assigns highest probability to evading on the right before the pedestrians begin to evade to either side. Middle left and middle right: The robotic wheelchair and the pedestrians evade each other in a natural manner. Right: After the situation has been resolved, the wheelchair proceeds to its target position.

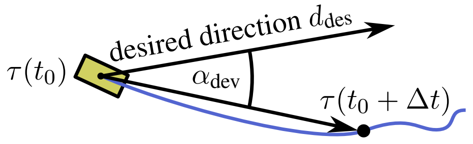


Figure 15: The angle $\alpha_{\text{dev}}$ used in the feature modeling preferred user direction.

### 5.2.4 Comparison to Traditional Path Planning

To compare the performance of our method with the performance of a global path planning algorithm that is designed for dynamic environments, we furthermore conducted a set of experiments in simulation. We therefore implemented an A$^*$ planner in configuration-time space that predicts the motion of the pedestrians with a constant velocity model. We set the maximum velocity to $0.5\,\frac{m}{s}$, which is similar to the velocity learned by the policy used for our approach. To acquire realistic test data, we used the abovementioned laser-based people tracker to record natural evasive movements of two pedestrians that evaded a third person in a hallway. To allow for a fair comparison, we fixed the trajectories of the two pedestrians and let the method proposed in this paper and the A$^*$ planner control the third person, respectively.

Tab. 1 summarizes the results of the two methods averaged over 10 different scenarios. The results suggest that our method learned a policy that safely evaded the other pedestrians at a minimum clearance of 0.34 m and reached its target at an average velocity of $0.5\,\frac{\text{m}}{\text{s}}$. In contrast, it turned out that it is difficult to tune the A$^*$-planner to obtain satisfactory results. Setting the minimal planned distance between the robot and dynamic obstacles to a low value such as 0.2 m, the planner did not sufficiently evade the pedestrians. To achieve an acceptable clearance, the minimal planned distance to dynamic obstacles needed to be set to 0.8 m since the pedestrians do not always comply with the constant velocity assumption. However, a value this large prevents the A$^*$-planner from finding a path to the target in many situations. In our experiments, the A$^*$-planner failed and needed to stop the robot in 60 % of the time steps, resulting in a rather low average velocity of $0.27\,\frac{\text{m}}{\text{s}}$. This problem is referred to as the “freezing robot problem” (Trautman and Krause, 2010). Our experiment demonstrates the shortcomings of A$^*$-like path planners when navigating in the presence of humans. The approach presented in this paper is able to predict cooperative behavior of nearby pedestrians and is therefore able to navigate a robot efficiently and in a socially compliant way.

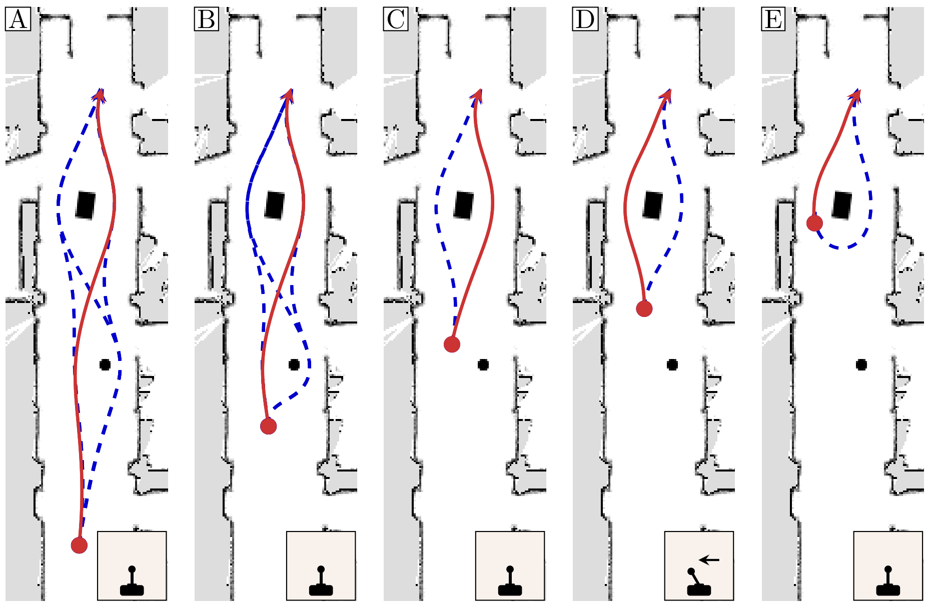


Figure 16: Shared autonomy control of a wheelchair in a corridor with two static obstacles. The figure shows the best trajectory according to the cost function, which the wheelchair executes (red) and an additional set of optimized trajectories in different homotopy classes (blue, dashed). The joystick in the lower right indicates the current direction preference of the user.

### 5.2.5 Shared Autonomy Navigation

Our method enables shared autonomy navigation, i.e., navigation in which a user and the system both contribute to the resulting navigation behavior. To this end, our technique maintains a set of trajectories each of which is locally optimized with respect to a user-defined cost function. We use features that incorporate user preferences online during navigation. The robot follows the trajectory that has lowest cost according to a tradeoff between these user preferences and the parts of the cost function that penalize properties such as high velocities or closeness to obstacles. The feature weights determine how strongly the robot follows the user preferences.

Let us assume a wheelchair scenario in which the handicapped user is only capable of issuing high-level commands rather than low-level controls. For example, such a user might want to express navigation preferences by joystick deflection, by head posture (Mandel and Frese, 2007) or even through brain-machine interfaces (Carlson and Milln, 2013). To achieve this with our approach, we introduce a feature $f_{\text{dir}}$ that penalizes the deviation $\alpha_{\text{dev}}$ of a trajectory from the preferred user direction:

$$f_{\text{dir}}(\tau) = \alpha_{\text{dev}}^2 = \arccos^2 \frac{d(\tau) \cdot d_{\text{desired}}}{\|d(\tau)\| \|d_{\text{desired}}\|}, \tag{44}$$

where $d(\tau)$ is the direction of the trajectory and $d_{\text{desired}}$ is the direction selected by the user, as illustrated in Figure 15. To compute $d(\tau)$ we use the location of trajectory $\tau$ at the time $\Delta t$ in the future, i.e., $d(\tau) = \tau(t_0 + \Delta t) - \tau(t_0)$. Note that we use this feature only for evaluating the costs of the optimized trajectories in the selection process, not for the optimization itself.

We conducted an experiment, in which we applied this method to shared autonomy control of our robotic wheelchair. We allowed the user to bias the system by adding $f_{\text{dir}}$ to the cost function, which reflects the deviation of the joystick from the planned trajectory. Figure 16 shows how the wheelchair navigated the corridor in which we placed additional static obstacles: (A) Our system has computed a set of optimized trajectories in different homotopy classes; (B) Without user preferences, the system selects and follows the green trajectory since it has lowest costs; (C) The wheelchair discarded one of the trajectory alternatives that require turning around since its costs have exceeded a threshold; (D) In front of the second obstacle, the user turns the joystick to the left, which biases the costs of each trajectory, and the wheelchair selects the left trajectory; (E) The wheelchair follows the left trajectory since it now has the lowest costs.

## 6 Conclusion

We presented a novel approach that allows a mobile robot to learn a model of the navigation behavior of cooperatively navigating agents such as pedestrians. Based on observations of their continuous trajectories, our method infers a model of the underlying decision-making process. To cope with the discrete and continuous aspects of this process, our model uses a joint mixture distribution that captures the discrete decisions regarding the homotopy classes of the composite trajectories as well as continuous properties of the trajectories such as higher-order dynamics. To compute the feature expectations with respect to the continuous, high-dimensional probability distributions, our method uses Hamiltonian Markov chain Monte Carlo sampling. To efficiently explore the space of trajectories, we use a Voronoi graph of the environment. The learned model enables socially compliant mobile robot navigation since it allows the robot to predict the navigation behavior of pedestrians in terms of a probability distribution over their trajectories. A Turing test suggests that the pedestrian trajectories induced by our approach appear highly human-like. Furthermore, a cross validation demonstrates that our method generalizes to new situations and outperforms three state-of-the-art techniques. A set of experiments with a real robot illustrates the applicability of our approach to socially compliant mobile robot navigation.

## Acknowledgment

This work has partly been supported by the German Research Foundation (DFG) under contract number SFB/TR-8, and by the EC under contract number FP7-610603.

## References

P. Abbeel and A. Ng. Apprenticeship learning via inverse reinforcement learning. In *Proc. of the Int. Conf. on Machine Learning (ICML)*, 2004.

G. Arechavaleta, J.-P. Laumond, H. Hicheur, and A. Berthoz. An optimality principle governing human walking. *IEEE Transactions on Robotics (T-RO)*, 24(1):5–14, 2008.

B. D. Argall, S. Chernova, M. Veloso, and B. Browning. A survey of robot learning from demonstration. *Robotics and Autonomous Systems*, 57(5):469–483, 2009.

C. Atkeson and S. Schaal. Robot learning from demonstration. In *Proc. of the Fourteenth Int. Conf. on Machine Learning (ICML)*, 1997.

F. Aurenhammer. Voronoi diagrams – a survey of a fundamental geometric data structure. *ACM Computing Surveys (CSUR)*, 23(3):345–405, 1991.

S. Bhattacharya, M. Likhachev, and V. Kumar. Topological constraints in search-based robot path planning. *Autonomous Robots*, 33(3):273–290, 2012.

C. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag New York, Inc., 2006.

S. Bitgood and S. Dukes. Not another step! economy of movement and pedestrian choice point behavior in shopping malls. *Environment and Behavior*, 38(3):394–405, 2006.

A. Boularias, J. Kober, and J. R. Peters. Relative entropy inverse reinforcement learning. In *Int. Conf. on Artificial Intelligence and Statistics*, pages 182–189, 2011.

T. E. Carlson and J. d. R. Milln. Brain-Controlled Wheelchairs: A Robotic Architecture. *IEEE Robotics and Automation Magazine*, 20(1), 2013.

H. Choset and J. Burdick. Sensor-based exploration: The hierarchical generalized voronoi graph. *Int. Journal of Robotics Research (IJRR)*, 19(2):96–125, February 2000.

H. Christensen and E. Pacchierotti. Embodied social interaction for robots. *AISB-05*, pages 40–45, 2005.

D. Demyen and M. Buro. Efficient triangulation-based pathfinding. In *Proc. of the 21st Nat. Conf. on Artificial Intelligence (AAAI)*, 2006.

E. W. Dijkstra. A note on two problems in connexion with graphs. *Numerische Mathematik*, 1(1):269–271, 1959.

S. Duane, A. Kennedy, B. Pendleton, and D. Roweth. Hybrid monte carlo. *Physics Letters B*, 195(2):216–222, 1987.

P. Fiorini and Z. Shillert. Motion planning in dynamic environments using velocity obstacles. *Int. Journal of Robotics Research (IJRR)*, 17:760–772, 1998.

C. Fox. *An Introduction to the Calculus of Variations*. Courier Dover Publications, 1987.

D. Fox, W. Burgard, and S. Thrun. The dynamic window approach to collision avoidance. *IEEE Robotics & Automation Magazine (RAM)*, 4(1):23–33, 1997.

S. Gulati, C. Jhurani, B. Kuipers, and R. Longoria. A framework for planning comfortable and customizable motion of an assistive mobile robot. In *Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, 2009.

S. Guy, M. Lin, and D. Manocha. Modeling collision avoidance behavior for virtual humans. In *Proc. of the Ninth Int. Conf. on Autonomous Agents and Multi-Agent Systems (AAMAS)*, pages 575–582, 2010.

S. Guy, J. van den Berg, W. Liu, R. Lau, M. Lin, and D. Manocha. A statistical similarity measure for aggregate crowd dynamics. *ACM Transactions on Graphics (TOG)*, 31(6): 190, 2012.

E. Hall. *The Hidden Dimension*. Doubleday, New York, 1966.

W. Hastings. Monte carlo sampling methods using markov chains and their applications. *Biometrika*, 57(1):97–109, 1970.

D. Helbing and A. Johansson. Pedestrian, crowd and evacuation dynamics. In *Encyclopedia of Complexity and Systems Science*, pages 6476–6495. Springer New York, 2009.

D. Helbing and P. Molnar. Social force model for pedestrian dynamics. *Physical Review E (PRE)*, 51:4282–4286, 1995.

S. Hoogendoorn and P. Bovy. Simulation of pedestrian flows by optimal control and differential games. *Optimal Control Applications and Methods*, 24(3):153–172, 2003.

E. Jaynes. Where do we stand on maximum entropy. *Maximum Entropy Formalism*, pages 15–118, 1978.

A. Johansson, D. Helbing, and P. Shukla. Specification of the social force pedestrian model by evolutionary adjustment to video tracking data. *Advances in Complex Systems (ACS)*, 10:271–288, 2007.

M. Kalakrishnan, P. Pastor, L. Righetti, and S. Schaal. Learning objective functions for manipulation. In *Proc. of the IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2013.

M. Kalakrishnan, S. Chitta, E. Theodorou, P. Pastor, and S. Schaal. Stomp: Stochastic trajectory optimization for motion planning. In *Proc. of the IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2011.

N. Katoh, T. Ibaraki, and H. Mine. An efficient algorithm for k shortest simple paths. *Networks*, 12(4):411–427, 1982.

B. Kim and J. Pineau. Socially adaptive path planning in human environments using inverse reinforcement learning. *International Journal of Social Robotics*, pages 1–16, 2015.

R. Kirby, R. Simmons, and J. Forlizzi. Companion: A constraint optimizing method for person-acceptable navigation. In *IEEE Int. Symp. on Robot and Human Interactive Communication (RO-MAN)*, pages 607–612, 2009.

K. Kitani, B. Ziebart, D. Bagnell, and M. Hebert. Activity forecasting. In *Proc. of the European Conf. on Computer Vision (ECCV)*, 2012.

H. Kretzschmar, M. Kuderer, and W. Burgard. Learning to predict trajectories of cooperatively navigating agents. In *Proc. of the IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2014.

M. Kuderer, H. Kretzschmar, C. Sprunk, and W. Burgard. Feature-based prediction of trajectories for socially compliant navigation. In *Proc. of Robotics: Science and Systems (RSS)*, 2012.

M. Kuderer, H. Kretzschmar, and W. Burgard. Teaching mobile robots to cooperatively navigate in populated environments. In *Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, 2013.

M. Kuderer, C. Sprunk, H. Kretzschmar, and W. Burgard. Online generation of homotopically distinct navigation paths. In *Proc. of the IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2014.

A. Lerner, Y. Chrysanthou, and D. Lischinski. Crowds by example. *Computer Graphics Forum*, 26(3):655–664, 2007.

C. Mandel and U. Frese. Comparison of wheelchair user interfaces for the paralysed: Head-joystick vs. verbal path selection from an offered route-set. In *European Conf. on Mobile Robots (ECMR)*, 2007.

K. Mombaur, A. Truong, and J.-P. Laumond. From human to humanoid locomotion – an inverse optimal control approach. *Autonomous Robots*, 28:369–383, 2010.

J. Müller, C. Stachniss, K. Arras, and W. Burgard. Socially inspired motion planning for mobile robots in populated environments. In *Proc. of the Int. Conf. on Cognitive Systems (COGSYS)*, pages 85–90, 2008.

A. Ng and S. Russell. Algorithms for inverse reinforcement learning. In *Proc. of the Int. Conf. on Machine Learning (ICML)*, 2000.

A. Pandey and R. Alami. A framework for adapting social conventions in a mobile robot motion in human-centered environment. In *Int. Conf. on Advanced Robotics (ICAR)*, pages 1–8, 2009.

S. Pellegrini, A. Ess, K. Schindler, and L. Van Gool. You'll never walk alone: Modeling social behavior for multi-target tracking. In *Proc. of the IEEE Int. Conf. on Computer Vision (ICCV)*, 2009.

Q.-C. Pham, H. Hicheur, G. Arechavaleta, J.-P. Laumond, and A. Berthoz. The formation of trajectories during goal-oriented locomotion in humans. II. a maximum smoothness model. *European Journal of Neuroscience*, 26:2391–2403, 2007.

N. Ratliff, M. Zucker, J. A. Bagnell, and S. Srinivasa. CHOMP: Gradient optimization techniques for efficient motion planning. In *Proc. of the IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2009a.

N. Ratliff, M. Zucker, J. A. Bagnell, and S. Srinivasa. Chomp: Gradient optimization techniques for efficient motion planning. In *Proc. of the IEEE Int. Conf. on Robotics and Automation (ICRA)*, pages 489–494, 2009b.

N. Ratliff, J. Bagnell, and M. Zinkevich. Maximum margin planning. In *Proc. of the Int. Conf. on Machine Learning (ICML)*, 2006.

M. Riedmiller and H. Braun. A direct adaptive method for faster backpropagation learning: The RPROP algorithm. In *Proc. of the IEEE Int. Conf. on Neural Networks (ICNN)*, 1993.

C. Sprunk, B. Lau, P. Pfaff, and W. Burgard. Online generation of kinodynamic trajectories for non-circular omnidirectional robots. In *Proc. of the IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2011.

S. Thrun, D. Fox, W. Burgard, and F. Dellaert. Robust monte carlo localization for mobile robots. *Artificial Intelligence*, 128(1-2):99–141, 2000.

P. Trautman and A. Krause. Unfreezing the robot: Navigation in dense, interacting crowds. In *Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, 2010.

P. Trautman, J. Ma, R. M. Murray, and A. Krause. Robot navigation in dense human crowds: the case for cooperation. In *Proc. of the IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2013.

J. van den Berg, S. Guy, M. Lin, and D. Manocha. Reciprocal n-body collision avoidance. In *Proc. of the Int. Symp. of Robotics Research (ISRR)*, 2009.

P. Vernaza and D. Bagnell. Efficient high dimensional maximum entropy modeling via symmetric partition functions. In *Advances in Neural Information Processing Systems (NIPS)*, volume 25, pages 584–592. 2012.

P. Vernaza, V. Narayanan, and M. Likhachev. Efficiently finding optimal winding-constrained loops in the plane. In *Proc. of Robotics: Science and Systems (RSS)*, 2012.

W. Warren. The dynamics of perception and action. *Psychological Review*, 113:358–389, 2006.

M. Yoda and Y. Shiota. Analysis of human avoidance motion for application to robot. In *Proc. of the 5th IEEE Int. Workshop on Robot and Human Communication*, pages 65–70. IEEE, 1996.

M. Yoda and Y. Shiota. The mobile robot which passes a man. In *Proc. of the 6th IEEE Int. Workshop on Robot and Human Communication*, pages 112–117. IEEE, 1997.

D. Zambrano, D. Bernardin, D. Bennequin, C. Laschi, and A. Berthoz. A comparison of human trajectory planning models for implementation on humanoid robot. In *In Proc. of the 4th IEEE RAS and EMBS Int. Conf. on Biomedical Robotics and Biomechatronics (BioRob)*, pages 663–668, 2012.

B. Ziebart, A. Maas, J. Bagnell, and A. Dey. Maximum entropy inverse reinforcement learning. In *Proc. of the AAAI Conf. on Artificial Intelligence (AAAI)*, 2008.

B. Ziebart, N. Ratliff, G. Gallagher, C. Mertz, K. Peterson, J. Bagnell, M. Hebert, A. Dey, and S. Srinivasa. Planning-based prediction for pedestrians. In *Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, 2009.

B. Ziebart, A. Dey, and J. Bagnell. Probabilistic pointing target prediction via inverse optimal control. In *Proc. of the ACM Int. conference on Intelligent User Interfaces*. ACM, 2012.