Probabilistic Map-based Pedestrian Motion Prediction Taking Traffic Participants into Consideration

Jingyuan Wu[1], Johannes Ruenz[1], and Matthias Althoff[2]

Abstract— **As pedestrians are one of the most vulnerable traffic participants, their motion prediction is of utmost importance for intelligent transportation systems. Predicting motions of pedestrians is especially hard since they move in less structured environments and have less inertia compared to road vehicles. To account for this uncertainty, we present an approach for probabilistic prediction of pedestrian motion using Markov chains. In contrast to previous work, we not only consider motion models, constraints from a semantic map, and various goals, but also explicitly adapt the prediction based on crash probabilities with other traffic participants. Also, our approach works in any situation; this is typically challenging for pure machine learning techniques that learn behaviors for a particular road section and which might consequently struggle with a different road section. The usefulness of combining the aforementioned aspects in a single approach is demonstrated by an evaluation using recordings of real pedestrians.**

I. Introduction

A. Motivation

Prediction of other traffic participants is an integral part in motion planning of autonomous vehicles or for threat assessment in driver assistant systems [1]. The main challenge in predicting the behavior of other traffic participants is the inherent uncertainty about their future actions. Several strategies have been developed: a) predicting a single likely behavior [2]–[6], b) predicting several behaviors possibly weighted by probabilities [7]–[9], c) computing probability distributions of future behaviors [10]–[12], and d) bounding behaviors by sets [13], [14]. All of the above techniques are particularly useful for certain applications. Single behaviors are easy to compute and are thus often used to warn drivers in simple driving assistance applications. Predicting several behaviors is useful for more advanced threat assessment, and probability distributions are suitable for trajectory planning; a comparison between both techniques can be found in [15]. Set-based prediction is useful for formal verification of motion plans and can be combined with probabilistic approaches [13]: probabilistic techniques help strategic planning, while set-based techniques verify fail-safe maneuvers over shorter time horizons.

B. Literature Review

Much work has been done on predicting road vehicles, while vulnerable road users have received less attention [1]. This work focuses on probabilistic prediction of pedestrians, whose actions are particularly hard to predict. Due to their smaller inertia compared to road vehicles, they can change their behavior quite unpredictably, and especially in open space, their future direction is quite unclear. We categorize existing work into short-term prediction, prediction to cross a road, goal-oriented prediction, interaction-aware prediction, and prediction in known environments.

a) Short-term prediction: We consider a prediction to be short-term when it is not longer than around 2 seconds. For those prediction horizons, classical filtering techniques, such as Kalman filters or interacting multiple models, provide good results [16], [17]. A slightly different approach using an interacting multiple model filter in combination with a latent-dynamic conditional random field model is presented in [18]. In [19] Bayesian networks and Gaussian mixture models are used to predict the future behavior using discretized velocities (standing, walking, jogging, and running) for short-term prediction. Models for short-term prediction are learned in [20] using the method of moments and least-squares optimization. Prediction using Markov chains is beneficial when the distributions are non-Gaussian and when constraints have to be considered [21]. In particular, the acceleration phase after standing is investigated in [22].

b) Crossing prediction: Due to the importance of predicting whether a pedestrian will cross a road—automated vehicles have to decide whether to brake or not—much work exists on this particular prediction problem. A dynamic Bayesian network is combined with a switching linear dynamical system in [23], where the switching is governed by several factors (e.g., pedestrian head orientation as evidence for situational awareness). In [24] the features mostly determining whether a person will cross a street are extracted from recorded data using a support vector machine, but no precise spatial prediction is performed. The binary decision for crossing a street or not is also investigated in [25] using a gap-acceptance approach with a special focus on whether the vehicle or the pedestrian will yield at a crosswalk. To incorporate the specific setting of a zebra crossing in a more general inner-city model, a context model tree is developed in [26]. Several techniques (Gaussian process dynamical models, probabilistic hierarchical trajectory matching, Kalman filter, and interacting multiple models) are compared in [27] and evaluated based on real experiments.

c) Goal-oriented prediction: Goal-oriented planning using Markov decision processes is used in [28] to predict pedestrian trajectories in indoor environments. To obtain a variety of motions, the value iteration procedure for Markov

[1] Jingyuan Wu and Joahnnes Ruenz are with Robert Bosch GmbH, D-74232 Abstatt, Germany, `jingyuan.wu@de.bosch.com` and `johannes.ruenz@de.bosch.com`

[2] Matthias Althoff is with the Department of Computer Science, Technical University of Munich, D-85748 Garching, Germany, `althoff@in.tum.de`

decision processes is relaxed. This approach has been further improved in [29] using inverse optimal control and considering observation uncertainty in a hidden variable Markov decision process. Similarly, in [30], goals are used to guide the prediction, but jump-Markov processes are used instead of solving relaxed Markov decision processes. A framework in [31] combines a velocity-based model with a cost-based model (deterministic planning). Gaussian-distributed inputs to a continuous model in combination with backwards prediction from uncertain goals is proposed in [32].

d) Interaction-aware prediction: Many prediction techniques in crowded environments rely on the social force model [33]. Prediction in densely populated environments is performed with learned Gaussian process regression models in [34]. A particle filter is combined with an agent-based crowd model that infers collision-free velocities to combine methods from short- and long-term prediction [35].

e) Prediction in known environments: In known and structured environments, learning typical behavior patterns results in very accurate predictions. Clustering techniques are applied in [36]. Long short-term memory neural nets (subclass of recurrent neural nets) are used in [37] to predict human movement. Another technique is to use kriging-based models estimating pedestrian movement flows from previously observed pedestrian trajectories [38]. Local filtering techniques and global paths learned from observation are combined in [39]. Further, polynomial least-squares approximation and multilayer perceptron artificial neural networks are used to predict pedestrian trajectories at a fixed intersection [40]. Finally, matching with classified motion patterns is used [41].

C. Contributions

In this paper, we address the problem of predicting motions of pedestrians. This work is based on [12], [21], [42] using Markov chains for motion prediction due to their ability to deal with the typical non-Gaussian distribution and several constraints in generic city traffic scenarios. We extend these works by incorporating policies to predict the moving direction [28]–[31] and by taking other traffic participants into account; the latter is realized by adapting the collision checking approach for vehicles in [42] to risk checking based on gap-acceptance [25], [43]. In particular, this extension enables us to predict the multimodal distribution of pedestrians' motion depending on dynamic environments. Moreover, we propose a heuristic method to automatically infer the positions of several potential goals on a generic semantic map. This procedure can remove the necessity in [30], [31] that goals are assumed to be given a priori. Finally, using real-world data, we compare the average position deviation between Markov chains with different input transition matrices.

D. Overview

As a result of using Markov chains for motion prediction, one can obtain probabilistic occupancy grids directly, see the top part of Fig. 1. The abstraction of continuous movement of pedestrians to Markov chains is described in Sec. II. We generate different Markov chains for various intervals of the inputs *orientation* (moving direction) and *velocity*. After predicting probability distributions of inputs (lower part of Fig. 1), our prediction combines the various Markov chains for different input values with their respective probability distributions as presented in Sec. II-C. Sec. III involves the extended prediction of probability distributions of *orientation* by taking semantic maps into consideration. In Sec. IV, the influence of dynamic environments on the probability distributions of inputs is modeled. For example, a multimodal distribution of *velocity* as depicted in the lower middle part of Fig. 1 implies that a pedestrian, in the presence of other traffic participants, might slow down to stop or accelerate to cross rather than cross at its current velocity (lower left part of Fig. 1).

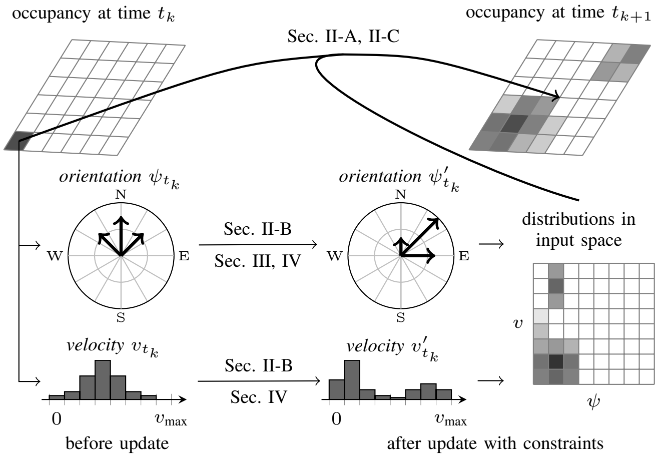


Fig. 1: Overview of the approach. In the top part we present the propagation of one cell of the occupancy grid at time t_k (the darker the color, the higher the probability value). The lower left part shows the separate conditional probability distributions of the inputs *orientation* and *velocity* of the only occupied cell before update, while the updated input probability distributions with constraints are visualized separately in the lower middle part and jointly in the lower right part, respectively.

II. Markov chain of pedestrian dynamics

To probabilistically predict motions of pedestrians in open space, discrete Markov chains accounting for dynamic constraints of human beings are generated. This section is inspired by [42].

A. Abstraction of Motion Model

Given the orientation ψ and the velocity v as inputs $\mathbf{u} = (\psi, v)^T$, the state $\mathbf{x} = (x_E, x_N)^T$ containing the 2D position can be propagated by the following dynamic model:

$$\dot{x}_E = v \cos\psi, \qquad \dot{x}_N = v \sin\psi. \tag{1}$$

In order to obtain probabilistic occupancy grids, the above continuous motion model is abstracted to a discrete time Markov chain. There are three main reasons for this: first, the generation of Markov chains can be done offline and their online execution is computationally inexpensive; second,

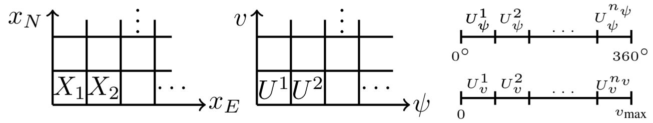


Fig. 2: Discrete state (left) and input (middle) space [12]. x_E and x_N represent the 2D position as states, while orientation ψ and velocity v are inputs. Separate indexes for intervals of ψ and v (right).

integrating a continuous probability density function can be reduced to a summation of probabilities; third, it is also advantageous to treat the state and input space as grids for better compatibility with discrete semantic maps.

The generation of a Markov chain can be divided into two steps: a) discretizing the state and input space, see Fig. 2, and b) determining transition probabilities. Each state cell is denoted by $X_i \subset \mathbb{R}^2$ with subscript index i (Latin letter) and each input cell by $U^\alpha \subset \mathbb{R}^2$ with superscript index α (Greek letter). Computing the transition probability from a cell X_j to another X_i can be done with Monte Carlo simulations. Samples are uniformly generated within each state and input cell. After introducing the number of samples generated jointly from the state cell X_j and the input cell U^α as n_j^α and those ending up in X_i after time T as $n_{i,j}^\alpha$, transition probabilities are computed as

$$\Phi_{ij}^\alpha := P(\mathbf{x}_{t_{k+1}} \in X_i | \mathbf{x}_{t_k} \in X_j, \mathbf{u}_{t_k} \in U^\alpha) = \frac{n_{i,j}^\alpha}{n_j^\alpha}, \quad (2)$$

where $\Phi^\alpha \in \mathbb{R}^{d \times d}$ is the transition matrix for d state cells subject to the set of inputs U^α and Φ_{ij}^α represents the element in the i-th row and j-th column of Φ^α. An alternative is to obtain Markov chains from a continuous model by reachability analysis [42].

B. Conditional Probability of Inputs

This subsection addresses the problem of obtaining input distributions for a given state. The conditional probability of an input cell U^α for a given state cell X_i is denoted by $q_i^\alpha := P\left(\mathbf{u} \in U^\alpha | \mathbf{x} \in X_i\right)$. Let $\Gamma_i(t_k) \in \mathbb{R}^{c \times c}$ be a state-dependent and possibly time-varying input transition matrix for c input cells. Then the conditional probabilities $q_i^\beta(t_k)$ can be updated according to the input transition values $\Gamma_i^{\alpha\beta}(t_k)$ instantaneously at times t_k:

$$q_i^\alpha(t_k)' = \sum_\beta \Gamma_i^{\alpha\beta}(t_k) q_i^\beta(t_k), \quad (3)$$

where $\Gamma_i^{\alpha\beta}$ represents the element in the α-th row and β-th column of Γ_i. The input transition probabilities $\Gamma_i^{\alpha\beta}$ consist of two components: a) the intrinsic transition probability $\Psi^{\alpha\beta}$, which models the behavior of pedestrians by taking their physical dynamic constraints into account, and b) the priority variable $0 \leq \lambda_i^\alpha \leq 1$ which gives an input cell a certain priority depending on the state. After introducing the normalization operator $\mathtt{norm}()$, the input transition probabilities are computed as

$$\Gamma_i^{\alpha\beta} = \mathtt{norm}(\hat{\Gamma}_i^{\alpha\beta}) := \frac{\hat{\Gamma}_i^{\alpha\beta}}{\sum_\alpha \hat{\Gamma}_i^{\alpha\beta}}, \quad \hat{\Gamma}_i^{\alpha\beta} = \lambda_i^\alpha \Psi^{\alpha\beta}. \quad (4)$$

Later, we will split each priority value λ_i^α into two parts: a) a static priority value for *orientation* $\lambda_{i,\text{stat}}^\alpha$ in Sec. III and b) a dynamic priority value $\lambda_{i,\text{dyn}}^\alpha$ in Sec. IV, such that

$$\lambda_i^\alpha = \lambda_{i,\text{stat}}^\alpha \lambda_{i,\text{dyn}}^\alpha. \quad (5)$$

Referring to different λ_i^α, the following input transition matrices for the prediction of pedestrian motion are created:

- *Basic Input Transition Matrix*:
  $\lambda_i^\alpha = 1, \forall i, \alpha$, thus only the state-independent and time-invariant intrinsic matrix Ψ plays a role.
- *Goal-oriented Input Transition Matrix*:
  By setting $\lambda_{i,\text{dyn}}^\alpha = 1, \forall i, \alpha$, only $\lambda_{i,\text{stat}}^\alpha$ is considered.
- *Extended Input Transition Matrix*:
  Both parts of priority values $\lambda_{i,\text{stat}}^\alpha$ and $\lambda_{i,\text{dyn}}^\alpha$ are explicitly computed.

Next, we present the computation of the intrinsic transition matrix Ψ. To obtain Ψ, the prediction of the inputs *orientation* and *velocity* is separately treated with respect to modeling physical dynamic constraints of pedestrians. For reasons of clarity, we introduce separate indexes $\alpha_\psi \in \{1, \ldots, n_\psi\}$ for n_ψ intervals of *orientation* $U_\psi^{\alpha_\psi} \subset \mathbb{R}$ and $\alpha_v \in \{1, \ldots, n_v\}$ for n_v intervals of *velocity* $U_v^{\alpha_v} \subset \mathbb{R}$, respectively. According to the index sequence in Fig. 2, it is clear that $U^\alpha = \{(\psi, v)^T | \psi \in U_\psi^{\alpha_\psi}, v \in U_v^{\alpha_v}\}$ with $\alpha = n_\psi \cdot (\alpha_v - 1) + \alpha_\psi$.

1) Orientation: We assume that the one-step transition of *orientation* from an interval $U_\psi^{\beta_\psi}$ to another $U_\psi^{\alpha_\psi}$ is subject to physical dynamic constraints of human beings and additionally depends on the current velocity v_{t_k}. This is motivated by the fact that the faster a pedestrian walks, the more difficult he or she can turn around. After introducing the operator $\mathtt{center}()$ which returns the volumetric center of a set and the operator $\mathtt{difference}()$ which limits the difference of two orientation values between 0 and 180 °, the transition probability $q^{\alpha_\psi \beta_{\psi,v}} := P\left(\psi'_{t_k} \in U_\psi^{\alpha_\psi} | \psi_{t_k} \in U_\psi^{\beta_\psi}, v_{t_k} \in U_v^{\beta_v}\right)$ is computed heuristically as

$$\begin{aligned} q^{\alpha_\psi \beta_{\psi,v}} \propto \exp\Big(&-k_1 \cdot \mathtt{center}(U_v^{\beta_v}) \\ &\cdot \mathtt{difference}\big(\mathtt{center}(U_\psi^{\alpha_\psi}) - \mathtt{center}(U_\psi^{\beta_\psi})\big)\Big) \end{aligned} \quad (6)$$

with the proportionality operator $\propto$ and a parameter $k_1 > 0$.

2) Velocity: We assume that the one-step transition of *velocity* from an interval $U_v^{\beta_v}$ to another interval $U_v^{\alpha_v}$ is also subject to physical dynamic constraints, since pedestrians have limited acceleration and deceleration abilities. Furthermore, we assume that a pedestrian, in the absence of other traffic participants, might keep a desired velocity v^* which can be adjusted according to online measurements. After mapping v^* into intervals of *velocity* to obtain a corresponding index α_v^*, the conditional probability $q^{\alpha_v \beta_v} :=$

$P\left(v'_{t_k} \in U_v^{\alpha_v} | v_{t_k} \in U_v^{\beta_v}\right)$ can be computed heuristically as

$$q^{\alpha_v\beta_v} \propto \frac{1}{(\alpha_v - \beta_v)^2 + k_2 \cdot (\alpha_v - \alpha_v^*)^2 + k_3}, \qquad (7)$$

where the term $(\alpha_v - \beta_v)^2$ implies that the bigger the difference of velocity values[3], the more unlikely this change is; the second term in the denominator with a parameter $k_2 \geq 0$ attempts to keep the distribution concentrated around a desired velocity; a high value of the parameter $k_3 > 0$ lets probability distributions converge to stable values quickly.

3) Intrinsic Transition Matrix: We simplify the computation of the components of the intrinsic transition matrix $\Psi^{\alpha\beta}$ using the conditional transition probabilities in (6) and (7):

$$\Psi^{\alpha\beta} = \texttt{norm}(\hat{\Psi}^{\alpha\beta}), \quad \hat{\Psi}^{\alpha\beta} = q^{\alpha_\psi\beta_{\psi,v}} q^{\alpha_v\beta_v}, \qquad (8)$$

with $\alpha = n_\psi \cdot (\alpha_v - 1) + \alpha_\psi$ and $\beta = n_\psi \cdot (\beta_v - 1) + \beta_\psi$.

C. Combined Propagation of States and Inputs

Fig. 1 shows the propagation procedure for one cell in the state space. To describe the propagation of all cells, we first define the joint probability of the state and input $p_i^\alpha := P(\mathbf{x} \in X_i, \mathbf{u} \in U^\alpha)$. Let the total probability of a state be denoted by $\hat{p}_i := P(\mathbf{x} \in X_i)$. Then it is clear that $\hat{p}_i = \sum_\alpha p_i^\alpha$ and $p_i^\alpha = q_i^\alpha \hat{p}_i$. After defining a joint probability vector $p^\alpha \in \mathbb{R}^d$ including all values of p_i^α for a fixed value of α, p^α can be propagated by using Φ^α in (2) as

$$p^\alpha(t_{k+1}) = \Phi^\alpha p^\alpha(t_k). \qquad (9)$$

As pointed out in [42], instead of updating the conditional probabilities q_i^α via $\Gamma_i^{\alpha\beta}(t_k)$ in (3), one can also update the joint probabilities p_i^α because the state probability $\hat{p}_i$ does not change instantaneously:

$$\text{Eq. (3)} \xrightarrow{p_i^\alpha = q_i^\alpha \hat{p}_i} p_i^\alpha(t_k)' = \sum_\beta \Gamma_i^{\alpha\beta}(t_k) p_i^\beta(t_k). \qquad (10)$$

In order to elegantly perform (9) and (10) simultaneously, we create a new probability vector $\tilde{p} \in \mathbb{R}^{c \cdot d \times 1}$ containing all p_i^α for c input cells and d state cells:

$$\tilde{p} = \left(p_1^1\, p_1^2\, \dots\, p_1^c\, p_2^1\, p_2^2\, \dots\, p_d^c\right)^T. \qquad (11)$$

Accordingly, a state transition matrix $\tilde{\Phi} \in \mathbb{R}^{c \cdot d \times c \cdot d}$ and an input transition matrix $\tilde{\Gamma} \in \mathbb{R}^{c \cdot d \times c \cdot d}$ need to be constructed. $\tilde{\Phi}$ is obtained by rearranging values of different Φ^α:

$$\tilde{\Phi} = \begin{pmatrix} \Phi_{11}^1 & 0 & 0 & \dots & 0 & \Phi_{12}^1 & 0 & \dots & 0 \\ 0 & \Phi_{11}^2 & 0 & \dots & 0 & 0 & \Phi_{12}^2 & \dots & 0 \\ \vdots & & & & & & & & \vdots \\ 0 & 0 & 0 & \dots & \Phi_{d1}^c & 0 & 0 & \dots & \Phi_{dd}^c \end{pmatrix}, \qquad (12)$$

and $\tilde{\Gamma}$ is computed as

$$\tilde{\Gamma} = \begin{pmatrix} \Gamma_1 & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{0} & \Gamma_2 & \dots & \mathbf{0} \\ \vdots & & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \dots & \Gamma_d \end{pmatrix}, \Gamma_i = \begin{pmatrix} \Gamma_i^{11} & \Gamma_i^{12} & \dots & \Gamma_i^{1c} \\ \Gamma_i^{21} & \Gamma_i^{22} & \dots & \Gamma_i^{2c} \\ \vdots & & \ddots & \vdots \\ \Gamma_i^{c1} & \Gamma_i^{c2} & \dots & \Gamma_i^{cc} \end{pmatrix}, \qquad (13)$$

where $\mathbf{0}$ is a matrix of zeros. This rewriting allows combined propagation of states and inputs:

$$\tilde{p}(t_{k+1}) = \tilde{\Gamma}(t_k)\tilde{\Phi}\tilde{p}(t_k). \qquad (14)$$

For a more detailed explanation, we refer to [42].

III. Goal-oriented Prediction

As pedestrians may prefer some paths to achieve their goals, we derive policies from semantic maps as static priority values $\lambda_{i,\text{stat}}^\alpha$ in (5). The aim of using $\lambda_{i,\text{stat}}^\alpha$ is to control the transition of the input *orientation* in a more rational direction regarding a semantic map.

A. Determining Positions of Goals

In this subsection, a heuristic method is proposed to automatically determine positions of potential goals of a pedestrian on a 2D semantic map. In contrast, in [30], [31] a finite set of goals is predefined and in [29] goals are generated from points along the perimeter of cars in a parking lot. The procedure of our method is described below:

1) Beginning with the measured position of a pedestrian "∗", see Fig. 3 (left), we draw a set of rays within 360° on a semantic map. These rays are absorbed at positions "▽" (*visible points*) by the most remote cells (related to "∗") representing the safe areas for staying, such as sidewalks, before intersecting with obstacles (black areas). The pseudo code for finding *visible points* "▽" is presented in Alg. 1.
2) Beginning with the initial cell, to which the position "∗" belongs, with the distance value (D-value) 0, we run Dijkstra's algorithm [44] until every *visible point* "▽" possesses a D-value related to the initial cell.
3) Absorbing process: beginning with an arbitrary *visible point* "▽", see magnified region in Fig. 3 (middle), we assign its D-value to the initial step value (S-value) of the process. At each declining step (S-value minus 1 iteratively), we detect the *pass by cells* "○" which are defined as the cells fulfilling the conditions shown in Line 3 of Alg. 2. Detection of "○" is interrupted, if the unsafe areas for staying such as roads or crosswalks come up, see all detected "○" from a beginning *visible point* in Fig. 3 (middle). Afterward, we delete those "▽" absorbed by "○" except for the beginning *visible point* with which this absorbing process begins (Lines 8, 13, and 14 of Alg. 2).
4) We repeat the absorbing process in 3), but begin with another *visible point* among the remaining ones, until no more *visible points* can be absorbed. Ultimately, the remaining *visible points* are regarded as potential goals "□", as shown in Fig. 3 (right).

B. Stochastic Policy

We estimate the policy for the input *orientation* based on a planning problem. The policy specifies the moving direction a pedestrian should move at a given position, so that the cost to the goal is minimal. Assuming that each state cell has a cost function $C(X_i, U_\psi^{\alpha_\psi})$ regarding its topographic

[3] As the indexes for discrete velocities are numbered in increasing sequence, their difference is a measure for the difference of values.

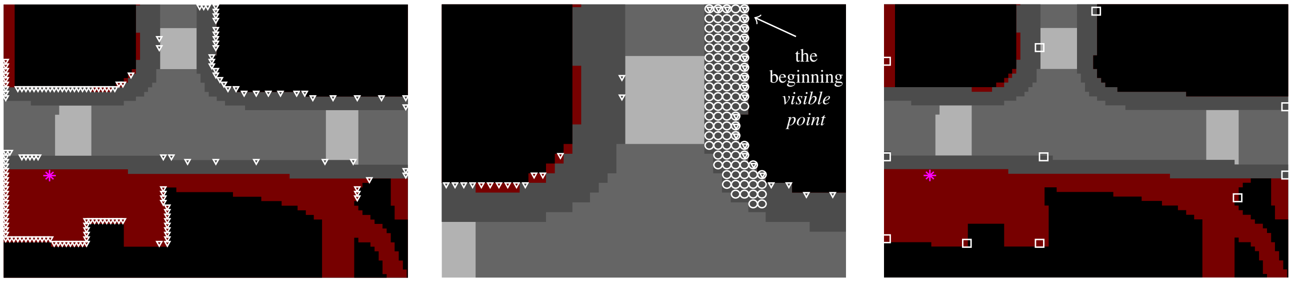


Fig. 3: Determining positions of potential goals on a semantic map with building (black), sidewalk type 1 (red), sidewalk type 2 (dark gray), road (gray), and zebra crossing (light gray). *Visible points* "▽" for a pedestrian "*" with 360° view (left); detected *pass by cells* "○" from the beginning *visible point* on the upper right corner towards a pedestrian with the tolerance setting of 3 m (middle); inferred goals "□" after removing absorbed *visible points* (right).

Algorithm 1 Finding *visible points*

```
Require: Position of the pedestrian $(x_E, x_N)$, set $\Delta angle$, set $\Delta radius$ according
    to the grid size, set $radius_{\max}$ according to the map size.
for $angle = 0 : \Delta angle : 2\pi$ do
    $radius = \Delta radius$
    while $radius \leq radius_{\max}$ do
        $x_E^r = x_E + radius \cdot \cos(angle), \quad x_N^r = x_N + radius \cdot \sin(angle)$
        if $(x_E^r, x_N^r)$ is out of the map region then
            break the while loop
        else
            find the cell $X_i$, such that $(x_E^r, x_N^r) \in X_i$
            if $\texttt{roadType}(X_i) \in \{\text{sidewalk, other safe areas}\}$ then
                $visiblePoint = X_i$
            else if $\texttt{roadType}(\texttt{X}_\texttt{i}) \in \{\text{buiding, other obstacles}\}$ then
                break the while loop
            end if
        end if
        $radius = radius + \Delta radius$
    end while
    $\{visiblePoints\} = \{visiblePoints\} \cup visiblePoint$
end for
```

Algorithm 2 Detecting *pass by cells* and deleting absorbed *visible points* by *pass by cells*

```
Require: Precomputation of distance values related to the position of the pedestrian
    $D(X_j), \forall X_j \in \{visiblePoints\}$, select one $X_i \in \{visiblePoints\}$ as the
    beginning visible point, set tolerance.
Initialize $S = D(X_i)$, $\{passByCells\} = X_i$, $\{passByCellsLastStep\} = X_i$
while $S > 0$ do
    $\{passByCellsThisStep\} = \{X_j | \texttt{abs}(D(X_j) - S) \leq tolerance,$
    $\texttt{roadType}(X_j) \in \{\text{sidewalk, other safe areas}\},$
    $X_j$ belong to the neighbor cells of $\{passByCellsLastStep\},$
    $X_j \notin \{passByCells\}\}$
    if $\{passByCellsThisStep\} \neq \emptyset$ then
        $\{passByCells\} = \{passByCells\} \cup \{passByCellsThisStep\}$
        $\{passByCellsLastStep\} = \{passByCellsThisStep\}$
    else
        $\{passByCells\} = \{passByCells\} \setminus X_i$
        break the while loop
    end if
    $S = S - 1$
end while
$\{absorbedVisiblePoints\} = \{visiblePoints\} \cap \{passByCells\}$
$\{visiblePoints\} = \{visiblePoints\} \setminus \{absorbedVisiblePoints\}$
```

Algorithm 3 A posteriori goal estimate for one goal [30]

```
Require: Measured position series $\mathbf{y}^{t_0}$ from $t_{m'}$ up to $t_0$.
1: Initialize $P\left(\mathbf{x}_{t_{m'}} | \mathbf{y}^{t_{m'}}, g_z\right)$, set $P\left(g_z | \mathbf{y}^{t_{m'}}\right) = 1/n_g$
2: for $m = m' + 1 : 1 : 0$ do
3:     $P\left(\mathbf{x}_{t_m} | \mathbf{y}^{t_{m-1}}, g_z\right) = \sum_{\mathbf{x}_{t_{m-1}}} \underbrace{P\left(\mathbf{x}_{t_m} | \mathbf{x}_{t_{m-1}}, g_z\right)}_{\text{MC-}Goal_{g_z}} P\left(\mathbf{x}_{t_{m-1}} | \mathbf{y}^{t_{m-1}}, g_z\right)$
4:     $P\left(\mathbf{x}_{t_m} | \mathbf{y}^{t_m}, g_z\right) \propto P\left(\mathbf{y}_{t_m} | \mathbf{x}_{t_m}, g_z\right) P\left(\mathbf{x}_{t_m} | \mathbf{y}^{t_{m-1}}, g_z\right)$
5:     $P\left(\mathbf{y}_{t_m} | \mathbf{y}^{t_{m-1}}, g_z\right) = \sum_{\mathbf{x}_{t_m}} P\left(\mathbf{y}_{t_m} | \mathbf{x}_{t_m}, g_z\right) P\left(\mathbf{x}_{t_m} | \mathbf{y}^{t_{m-1}}, g_z\right)$
6:     $P\left(g_z | \mathbf{y}^{t_m}\right) \propto P\left(\mathbf{y}_{t_m} | \mathbf{y}^{t_{m-1}}, g_z\right) P\left(g_z | \mathbf{y}^{t_{m-1}}\right)$
7: end for
```

feature (e.g., the positive cost value of a cell on a road is bigger than that on a sidewalk) and moving direction, one can obtain the cost to the goal $V(X_i)$ for every state cell using Dijkstra's algorithm [44]. However, pedestrians would not obey only one certain policy, thus stochastic policies are more appropriate, i.e., each action can have a certain probability. The stochastic policies for achieving a goal are regarded as the static priority values $\lambda_{i,\text{stat}}^{\alpha}$ in (5). Each stochastic policy for a certain *orientation* $U_{\psi}^{\alpha_\psi}$ can be computed according to the Boltzmann policy [30] as

$$\lambda_{i,\text{stat}}^{\alpha} \propto \exp\left(-k_4 \cdot \left(V(X_{\text{next}}) + C(X_i, U_{\psi}^{\alpha_\psi}) - V(X_i)\right)\right) \tag{15}$$

with $\alpha = n_\psi \cdot (\tilde{\alpha}_v - 1) + \alpha_\psi$ for $\tilde{\alpha}_v \in \{1, \ldots, n_v\}$, where X_{next} indicates the next cell to arrive by taking action $U_{\psi}^{\alpha_\psi}$ from X_i and the stochastic degree increases as the parameter $k_4 \to 0$ and decreases as $k_4 \to \infty$.

C. Probability of Goals

While Sec. III-A gives the positions of potential goals, this subsection computes the probabilities of n_g inferred goals. We leverage the past trajectory of a tracked pedestrian to update the probability of each goal $g_z, z \in \{1, \ldots, n_g\}$ for each pedestrian using Bayes' rule as in [30]. We first introduce the following notations: the past points in time are denoted by t_m, $m \in \{m'+1, \ldots, -1, 0\}$ with $m' \in \mathbb{Z}^-$; $t_{m'}$ and t_0 represent the time of the beginning of the tracking and the time of the last measurement (the beginning of the prediction), respectively. The measured position at t_m is denoted by $\mathbf{y}_{t_m}$ and the measurement series from $t_{m'}$ up to t_m by $\mathbf{y}^{t_m}$.

The pseudo code for estimating the probability of one goal g_z is presented in Alg. 3. In Line 1 of Alg. 3, the a posteriori probabilities of states $P\left(\mathbf{x}_{t_{m'}} | \mathbf{y}^{t_{m'}}, g_z\right)$ are initialized according to the measurement $\mathbf{y}_{t_{m'}}$, while the initial probability of the goal g_z is assigned an average value. At each time step t_m, a posteriori probabilities of states at the previous time step are propagated (Line 3) by Markov chains with the *Goal-oriented Input Transition Matrix* (MC-*Goal*$_{g_z}$). After multiplying a priori probabilities of states with the likelihood $P\left(\mathbf{y}_{t_m} | \mathbf{x}_{t_m}, g_z\right)$ of observing

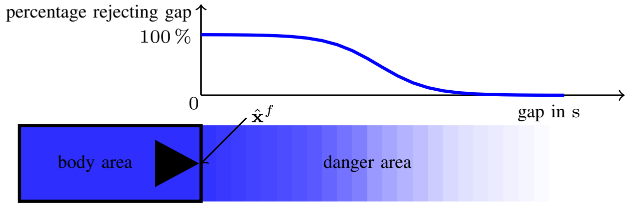


Fig. 4: Computing the weights for cells in the danger area caused by a vehicle based on gap-acceptance. The darker the color, the higher the weight.

the position $\mathbf{y}_{t_m}$, one obtains the a posteriori probabilities of states (Line 4) which are used for propagation at the next time step. Since we are interested in the a posteriori probabilities of goals rather than that of states, their marginal likelihood (Line 5) is combined with the a priori knowledge of the goal for the a posteriori goal estimate (Line 6). After running the above procedure up to t_0 and for each goal, we obtain their a posteriori probabilities $P(g_z|\mathbf{y}^{t_0})$.

IV. Interaction with Traffic Participants

It is obvious that pedestrians change their motion in the presence of other traffic participants. An algorithm to adapt the inputs *velocity* and *orientation* based on collision probabilities is described in this section.

A. Conditional Risk Probability

The idea is to constantly adapt the motion prediction of pedestrians by evaluating their collision probability with other traffic participants. To compute a risk measure at times $\tau \in \{t_k + T,\, t_k + 2T,\, \dots,\, t_k + T_{\text{check}}\}$ with a user-defined horizon T_{check}, we first need to predict collision probabilities for different inputs of pedestrians.

a) Prediction of pedestrians for given inputs and states: Since the aim is to compute the dynamic priority value $\lambda^{\alpha}_{i,\text{dyn}}(t_k)$ for each input cell U^{α} depending on a state cell X_i at time t_k, a vector $p^{i,\alpha}_{\text{check}}(t_k) \in \mathbb{R}^d$ is introduced whose i-th element equals 1 and others equal 0. For a constant input cell U^{α} during checking, one can obtain the evolution of $p^{i,\alpha}_{\text{check}}(t_k)$ at times τ according to (2) by

$$p^{i,\alpha}_{\text{check}}(\tau) = (\Phi^{\alpha})^{\frac{\tau - t_k}{T}}\, p^{i,\alpha}_{\text{check}}(t_k). \tag{16}$$

These distributions for given inputs and states are subsequently used to relate inputs with collision probabilities.

b) Prediction of vehicles: Due to the bigger inertia of vehicles compared to that of a pedestrian, their trajectories at times τ are computed using constant velocity and orientation. The position of the middle point of an object's front edge is denoted by $\hat{\mathbf{x}}^f$ and the velocity of this object by $\hat{v}$.

c) Body- and danger area: In order to integrate the pedestrians' traffic gap-acceptance into the collision checking approach for vehicles in [42], we use the dimensions of road vehicles propagated in their moving directions, which form a passed area which we refer to as a danger area (lower part of Fig. 4). We use a gap-rejection curve [25], [43] (top part of Fig. 4) to compute the probability for taking a certain action from a given position. Thus, the state cells in the danger area are assigned different weights $0 \leq w_{l,\tau} \leq 1$ with respect to the percentage of rejecting a gap, i.e., the time gap between the pedestrian and a moving object arrives at a position X_l successively. These $w_{l,\tau}$ are approximately computed based on a logistic function in [43]:

$$w_{l,\tau} \approx \frac{1}{1 + \exp\left(-6.96 + 1.19 \frac{\|\hat{\mathbf{x}}^f_{\tau} - \texttt{center}(X_l)\|_2}{\hat{v}_{t_0}}\right)}. \tag{17}$$

In contrast, the cells occupied by the vehicle body have a weight $w_{l,\tau} = 1$ accounting for collisions. The static obstacles are treated as occupied cells with a weight of one.

After introducing an indicator function $\texttt{ind}(X_j, X_l)$ which returns 1 if an object with reference point in X_j intersects with another object with a reference point in X_l and 0 otherwise, the conditional collision probability in [42] can be adapted to the conditional risk probability $p^{i,\alpha}_{\text{risk},\tau}$ approximately as

$$p^{i,\alpha}_{\text{risk},\tau} \approx \sum_{j,l,\hat{\mathbf{x}}^f_{\tau},\hat{v}_{t_0}} w_{l,\tau} P\left(\hat{\mathbf{x}}^f_{\tau}, \hat{v}_{t_0}\right) P\left(\mathbf{x}_{\tau} \in X_j\right) \texttt{ind}(X_j, X_l) \tag{18}$$

with the occupancy probability $P\left(\mathbf{x}_{\tau} \in X_j\right)$ as the j-th element of the probability vector $p^{i,\alpha}_{\text{check}}(\tau)$ in (16).

B. Dynamic Priority Values

To obtain the risk measure over a horizon $p^{i,\alpha}_{\text{risk},(t_k,t_k+T_{\text{check}}]}$, we pick the maximal $p^{i,\alpha}_{\text{risk},\tau}$ for a specific time in (18) [45]:

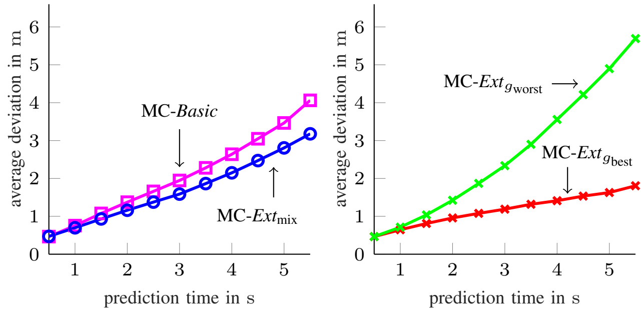
$$p^{i,\alpha}_{\text{risk},(t_k,t_k+T_{\text{check}}]} \approx \max_{\tau} p^{i,\alpha}_{\text{risk},\tau}. \tag{19}$$

As previously discussed, we regard $\lambda^{\alpha}_{i,\text{dyn}}(t_k)$ in (5) as the percentage of pedestrians accepting a gap, so that

$$\lambda^{\alpha}_{i,\text{dyn}}(t_k) = 1 - p^{i,\alpha}_{\text{risk},(t_k,t_k+T_{\text{check}}]}. \tag{20}$$

V. Evaluation

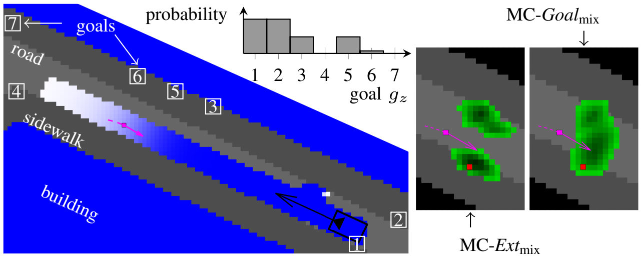
To evaluate the performance of our approach, we use 72 recordings of real pedestrians with a duration of at least 4.5 seconds each: 45 trajectories are taken from the KITTI dataset [46] (City 2011_09_29_drive_0071) and the other 27 are taken from our internal dataset measured in Rutesheim, Germany. For uniformity, we divide all recordings into two parts: the first part of recordings (0 to 1.5 s) are used for initialization and the rest for evaluation. Within the first part of each recording, the beginning of the prediction is chosen as $t_0 = 1.5\,\text{s}$; probability vectors are initialized by measurements at t_0; the positions of goals are inferred according to the measured position of pedestrian at $t_{m'} = 0\,\text{s}$; the a posteriori probabilities of goals are updated up to t_0 and are unchanged during prediction. The second part of each recording is used as reference data $\mathbf{y}_{t_k}$; the metric used for evaluating a single trajectory η is the weighted arithmetic average of position deviation $\delta x'_{t_k,\eta} = \sum_{i=1}^{d} \|\texttt{center}(X_i) - \mathbf{y}_{t_k}\|_2 P\left(\mathbf{x}_{t_k} \in X_i\right)$ [47]. Then $\delta x_{t_k} = \frac{1}{n_{\text{tra}}} \sum_{\eta=1}^{n_{\text{tra}}} \delta x'_{t_k,\eta}$ represents the average position deviations of n_{tra} trajectories at prediction times t_k.



(a) Overall performance of proposed MC regarding all goals.

(b) Individual analysis of proposed MC for certain goals.

Fig. 5: Average position deviations δx_{t_k} of all 72 trajectories at prediction times t_k from using Markov chains (MC) with different input transition matrices. In part (a), the weighted results regarding all goals from using *Extended Input Transition Matrices* (MC-*Ext*$_{\text{mix}}$) are better than the results from using the *Basic Input Transition Matrix* (MC-*Basic*). Since each *Extended Input Transition Matrix* is related to a certain goal, the results of the best g_{best} and worst goal g_{worst} among those goals with above-average a posteriori probabilities are depicted in part (b). The values for g_{best} and g_{worst} are obtained by computing their average position deviations.



(a) Oncoming traffic situation in a low speed area at $t_0 = 1.5$ s.

(b) Prediction at $t_k = 4.5$ s.

Fig. 6: A comparison between Markov chains with *Extended Input Transition Matrices* (MC-*Ext*$_{\text{mix}}$) and Markov chains with *Goal-oriented Input Transition Matrices* (MC-*Goal*$_{\text{mix}}$) regarding all goals. The part (a) shows an oncoming traffic situation where not only danger areas caused by vehicles, but also static obstacles are marked with "🟦" in the semantic map. The vehicle body is denoted by "□" and its moving direction by "→". In part (b), the predicted positions of the pedestrian are shown by "🟩". The darker the color, the higher the weight of "🟦" or probability of "🟩" is. The initial position of the pedestrian is shown by "■" with the moving direction "→" at t_0, the future position in the recording at t_k is shown by "🟥". Based on the past trajectory "....", the positions of inferred goals with estimated probabilities are depicted in part (a).

Fig. 5 shows the average position deviations δx_{t_k} of all trajectories from using Markov chains (MC) with different input transition matrices. As Markov chains with the *Extended Input Transition Matrix* (MC-*Ext*$_{g_z}$) only make predictions for a single goal and in order to describe the overall performance regarding all goals, the predicted occupancy grids for different goals are mixed with respect to their estimated a posteriori probabilities (MC-*Ext*$_{\text{mix}}$). In Fig. 5a, the comparison between MC-*Ext*$_{\text{mix}}$ and Markov chains with the *Basic Input Transition Matrix* (MC-*Basic*) shows the benefit of taking environment constraints into account additionally rather than pedestrian dynamics only.

An oncoming traffic situation in Fig. 6 illustrates the difference between using Markov chains with the *Goal-oriented Transition Matrix* (MC-*Goal*) and MC-*Ext*; the former is without risk checking with dynamic environments. Due to our risk measure, some actions are assigned lower priority values. Thus, our method MC-*Ext* predicts that the pedestrian will possibly turn right or turn left to avoid the risk area, see Fig. 6b.

VI. Discussion

Considering several constraints in Markov chains enables us to predict different behaviors of pedestrians. In particular, an adapted collision checking approach involving conflicts between pedestrians and other traffic participants is proposed. This is beneficial for interaction planning strategies of automated vehicles. However, this approach can be improved even more by considering enhanced environment knowledge, e.g., road constraints for vehicles and the situation awareness of pedestrians.

Our approach generates several possible paths to underlying goals. Despite some inappropriate goals, overall the prediction improves when considering goals in our tested scenarios. This also points out the potential for improving our approach by using more information to estimate the probabilities of goals.

Acknowledgment

We gratefully acknowledge financial support by the project *interACT* within the EU Horizon 2020 programme under grant agreement No 723395.

References

[1] S. Lefèvre, D. Vasquez, and C. Laugier, "A survey on motion prediction and risk assessment for intelligent vehicles," *ROBOMECH journal*, vol. 1, no. 1, pp. 1–14, 2014.

[2] A. Barth and U. Franke, "Where will the oncoming vehicle be the next second?" in *Proc. of the IEEE Intelligent Vehicles Symposium*, 2008, pp. 1068–1073.

[3] A. Eidehall, "Multi-target threat assessment for automotive applications," in *Proc. of the 14th Int. IEEE Conference on Intelligent Transportation Systems*, 2011, pp. 433–438.

[4] M. Brännström, E. Coelingh, and J. Sjöberg, "Model-based threat assessment for avoiding arbitrary vehicle collisions," *IEEE Transactions on Intelligent Transportation Systems*, vol. 11, no. 3, pp. 658–669, 2010.

[5] J.-H. Kim and D.-S. Kum, "Threat prediction algorithm based on local path candidates and surrounding vehicle trajectory predictions for automated driving vehicles," in *Proc. of the IEEE Intelligent Vehicles Symposium*, 2015, pp. 1220 – 1225.

[6] J. Wei, J. M. Snider, T. Gu, J. M. Dolan, and B. Litkouhi, "A behavioral planning framework for autonomous driving," in *Proc. of the IEEE Intelligent Vehicles Symposium*, 2014.

[7] A. Eidehall and L. Petersson, "Statistical threat assessment for general road scenes using Monte Carlo sampling," *IEEE Transactions on Intelligent Transportation Systems*, vol. 9, pp. 137–147, 2008.

[8] S. Danielsson, L. Petersson, and A. Eidehall, "Monte Carlo based threat assessment: Analysis and improvements," in *Proc. of the IEEE Intelligent Vehicles Symposium*, 2007, pp. 233–238.

[9] A. E. Broadhurst, S. Baker, and T. Kanade, "Monte Carlo road safety reasoning," in *Proc. of the IEEE Intelligent Vehicles Symposium*, 2005, pp. 319–324.

[10] T. Gindele, S. Brechtel, and R. Dillmann, "Learning driver behavior models from traffic observations for decision making and planning," *IEEE Intelligent Transportation Systems Magazine*, vol. 7, no. 1, pp. 69–79, 2015.

[11] A. Lambert, D. Gruyer, G. S. Pierre, and A. N. Ndjeng, "Collision probability assessment for speed control," in *Proc. of the 11th International IEEE Conference on Intelligent Transportation Systems*, 2008, pp. 1043–1048.

[12] M. Althoff, O. Stursberg, and M. Buss, "Model-based probabilistic collision detection in autonomous driving," *IEEE Transactions on Intelligent Transportation Systems*, vol. 10, no. 2, pp. 299 – 310, 2009.

[13] M. Althoff and S. Magdici, "Set-based prediction of traffic participants on arbitrary road networks," *IEEE Transactions on Intelligent Vehicles*, vol. 1, no. 2, pp. 187–202, 2016.

[14] M. Koschi and M. Althoff, "SPOT: A tool for set-based prediction of traffic participants," in *Proc. of the IEEE Intelligent Vehicles Symposium*, 2017.

[15] M. Althoff and A. Mergel, "Comparison of Markov chain abstraction and Monte Carlo simulation for the safety assessment of autonomous cars," *IEEE Transactions on Intelligent Transportation Systems*, vol. 12, no. 4, pp. 1237–1247, 2011.

[16] N. Schneider and D. M. Gavrila, "Pedestrian path prediction with recursive Bayesian filters: A comparative study," in *Proc. of the German Conference on Pattern Recognition*, 2013, pp. 174–183.

[17] J. Hariyono, A. Shahbaz, and K. H. Jo, "Estimation of walking direction for pedestrian path prediction from moving vehicle," in *Proc. of the IEEE/SICE International Symposium on System Integration*, 2015, pp. 750–753.

[18] A. T. Schulz and R. Stiefelhagen, "A controlled interactive multiple model filter for combined pedestrian intention recognition and path prediction," in *Proc. of the IEEE 18th International Conference on Intelligent Transportation Systems*, 2015, pp. 173–178.

[19] C. Braeuchle, J. Ruenz, F. Flehmig, W. Rosenstiel, and T. Kropf, "Situation analysis and decision making for active pedestrian protection using Bayesian networks," *6. Tagung Fahrerassistenzsysteme*, 2013.

[20] C. Zhou, B. Balle, and J. Pineau, "Learning time series models for pedestrian motion prediction," in *Proc. of the IEEE International Conference on Robotics and Automation*, 2016, pp. 3323–3330.

[21] F. Rohrmüller, M. Althoff, D. Wollherr, and M. Buss, "Probabilistic mapping of dynamic obstacles using Markov chains for replanning in dynamic environments," in *Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2008, pp. 2504–2510.

[22] M. Goldhammer, M. Gerhard, S. Zernetsch, K. Doll, and U. Brunsmann, "Early prediction of a pedestrian's trajectory at intersections," in *Proc. of the 16th International IEEE Conference on Intelligent Transportation Systems*, 2013, pp. 237–242.

[23] J. F. P. Kooij, N. Schneider, F. Flohr, and D. M. Gavrila, "Context-based pedestrian path prediction," in *European Conference on Computer Vision*. Springer, 2014, pp. 618–633.

[24] B. Völz, H. Mielenz, G. Agamennoni, and R. Siegwart, "Feature relevance estimation for learning pedestrian behavior at crosswalks," in *2015 IEEE 18th International Conference on Intelligent Transportation Systems*, 2015, pp. 854–860.

[25] B. J. Schroeder, "A behavior-based methodology for evaluating pedestrian-vehicle interaction at crosswalks," Ph.D. dissertation, North Carolina State University, 2008.

[26] S. Bonnin, T. H. Weisswange, F. Kummert, and J. Schmuedderich, "Pedestrian crossing prediction using multiple context-based models," in *Proc. of the 17th International IEEE Conference on Intelligent Transportation Systems*, 2014, pp. 378–385.

[27] C. G. Keller and D. M. Gavrila, "Will the pedestrian cross? A study on pedestrian path prediction," *IEEE Transactions on Intelligent Transportation Systems*, vol. 15, no. 2, pp. 494–506, 2014.

[28] B. D. Ziebart, N. Ratliff, G. Gallagher, C. Mertz, K. Peterson, J. A. Bagnell, M. Hebert, A. K. Dey, and S. Srinivasa, "Planning-based prediction for pedestrians," in *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2009, pp. 3931–3936.

[29] K. M. Kitani, B. D. Ziebart, J. A. Bagnell, and M. Hebert, "Activity forecasting," in *European Conference on Computer Vision*. Springer, 2012, pp. 201–214.

[30] V. Karasev, A. Ayvaci, B. Heisele, and S. Soatto, "Intent-aware long-term prediction of pedestrian motion," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*, 2016, pp. 2543–2549.

[31] D. Vasquez, "Novel planning-based algorithms for human motion prediction," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*, 2016, pp. 3317–3322.

[32] E. Rehder and H. Kloeden, "Goal-directed pedestrian prediction," in *Proc. of the IEEE International Conference on Computer Vision Workshop*, 2015, pp. 139–147.

[33] D. Helbing and P. Molnár, "Social force model for pedestrian dynamics," *Physical Review E: Statistical, Nonlinear, and Soft Matter Physics*, vol. 51, no. 5, pp. 4282–4286, 1995.

[34] Y. Li, M. L. Mekhalfi, M. M. Al Rahhal, E. Othman, and H. Dhahri, "Encoding motion cues for pedestrian path prediction in dense crowd scenarios," *IEEE Access*, vol. 5, pp. 24 368–24 375, 2017.

[35] W. Liu, A. B. Chan, R. W. H. Lau, and D. Manocha, "Leveraging long-term predictions and online learning in agent-based multiple person tracking," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 25, no. 3, pp. 399–410, 2015.

[36] Z. Chen, D. C. K. Ngai, and N. H. C. Yung, "Pedestrian behavior prediction based on motion patterns for vehicle-to-pedestrian collision avoidance," in *Proc. of the 11th International IEEE Conference on Intelligent Transportation Systems*, 2008, pp. 316–321.

[37] H. Xue, D. Q. Huynh, and M. Reynolds, "Bi-prediction: Pedestrian trajectory prediction based on bidirectional LSTM classification," in *Proc. of the International Conference on Digital Image Computing: Techniques and Applications*, 2017, pp. 1–8.

[38] K. Kawamoto, Y. Tomura, and K. Okamoto, "Learning pedestrian dynamics with kriging," in *Proc. of the IEEE/ACIS 15th International Conference on Computer and Information Science*, 2016, pp. 1–4.

[39] A. Bera, S. Kim, T. Randhavane, S. Pratapa, and D. Manocha, "GLMP-realtime pedestrian path prediction using global and local movement patterns," in *Proc. of the IEEE International Conference on Robotics and Automation*, 2016, pp. 5528–5535.

[40] M. Goldhammer, S. Köhler, K. Doll, and B. Sick, "Camera based pedestrian path prediction by means of polynomial least-squares approximation and multilayer perceptron neural networks," in *Proc. of the SAI Intelligent Systems Conference*, 2015, pp. 390–399.

[41] Z. Chen and N. H. C. Yung, "Improved multi-level pedestrian behavior prediction based on matching with classified motion patterns," in *Proc. of the 12th International IEEE Conference on Intelligent Transportation Systems*, 2009, pp. 1–6.

[42] M. Althoff, "Reachability analysis and its application to the safety assessment of autonomous cars," Ph.D. dissertation, Technische Universität München, 2010.

[43] M. Brewer, K. Fitzpatrick, J. Whitacre, and D. Lord, "Exploration of pedestrian gap-acceptance behavior at selected locations," *Transportation Research Record: Journal of the Transportation Research Board*, no. 1982, pp. 132–140, 2006.

[44] E. W. Dijkstra, "A note on two problems in connexion with graphs," *Numerische Mathematik*, vol. 1, no. 1, pp. 269–271, 1959.

[45] R. Altendorfer and C. Wilkmann, "What is the collision probability and how to compute it," *CoRR*, vol. abs/1711.07060, 2017. [Online]. Available: http://arxiv.org/abs/1711.07060

[46] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? The KITTI vision benchmark suite," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2012, pp. 3354–3361.

[47] J. Grossman, M. Grossman, and R. Katz, *The First Systems of Weighted Differential and Integral Calculus*. Non-Newtonian Calculus, 2006.