Inferring Pedestrian Motions at Urban Crosswalks

Benjamin Völz[1], Holger Mielenz[2], Igor Gilitschenski[3], Roland Siegwart[4], and Juan Nieto[4]

***Abstract*—Robust prediction of pedestrian behavior is one of the most challenging problems for autonomous driving. Particularly, predicting pedestrian crossings at crosswalks is of considerable importance for avoiding accidents on the one hand and not unnecessarily slowing down traffic on the other hand. Traditional model-based motion tracking and prediction approaches have difficulties in capturing abrupt changes in motions, as humans can perform. In this paper, an approach for predicting pedestrian motions that combines established motion tracking algorithms with data-driven methods is presented. The approach is built upon a hierarchical structure, where, first, the intent of each pedestrian is classified. Then, the approach computes several qualitative metrics, such as time-to-cross, for the pedestrians classified as crossing. The approach is evaluated on a challenging urban dataset collected for different types of crosswalks such as roundabouts and straight roads. The evaluation also provides a thorough analysis of the generalization performance of the proposed approach.**

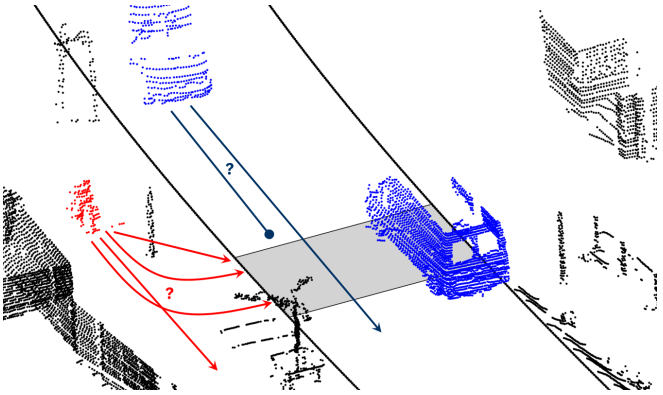


Fig. 1. Point cloud depicting a typical urban scenario: a car (blue) and a pedestrian (red) are approaching a crosswalk (grey box), where the pedestrian has priority. In this scenario we want to infer the pedestrians future motion. Some possible motions are depicted as red arrows. This information has the potential to enable automated vehicles to perform smooth manoeuvres in complicated traffic situations involving pedestrians.

I. Introduction

One important task for Advanced Driver Assistance Systems (ADAS) and autonomous vehicles is prediction of other participants' future actions. The accuracy and robustness of this will condition the certainty and quality of the decision making module. Interaction among vehicles has been intensively studied [1], [2]. On the other hand, interaction between vehicles and pedestrians requires other types of solutions and still remains as a major challenge. The main problem here arises due to the very different dynamics of the actors involved. While cars can drive very fast, they are, due to physical constraints, quite limited in terms of changing the movement direction. This simplifies their prediction significantly. Pedestrians on the other hand move relatively slow but very agile. They are able to do sharp (e.g. 90°) turns without a loss of speed. Due to this high agility current state-of-the-art pedestrian prediction systems focus on safety-related predictions. These predictions aim at time horizons of only few hundred milliseconds (e.g. [3]) and are usually used for pedestrian protection systems.

In this paper we want to address the problem of pedestrian intention prediction. Such systems are of paramount importance for safety, and also a key to enable natural and smooth maneuvers on the vehicle side. Let us illustrate the problem with a typical traffic scenario as depicted in Figure 1. An automated car and a pedestrian are approaching an urban crosswalk, where the pedestrian has the right-of-way. The car is obliged to stop if the pedestrian intends to cross the street. If we reflect about the behavior that the vehicle should have, we can derive a small set of requirements based on two important principles: safety and comfort. From the safety perspective we want to avoid both the passing by pedestrian with a small safety distance and the necessity for large accelerations, e.g. due to emergency braking maneuvers. The avoidance of large accelerations is also highly desirable for a comfortable driving, an essential feature for people to adopt the technology. Based on this point of view we can also generalize and state, that accelerations in general, and particularly full stops, should be avoided whenever possible. Accordingly the third important requirement can be defined: we only want to stop, if it is inevitable. Hence, if a pedestrian does not intend to cross the road, we do not want to stop. To be able to fulfill all aforementioned requirements it is necessary to both infer the pedestrians' intention and predict their motion as early as possible. Additionally to the necessity to provide timely predictions, we also have to minimize the amount of false predictions. Regardless of whether we mistakenly marked a crossing pedestrian as non-crossing or vice versa. Motivated by these problems, our work aims to develop a system that (i) minimizes false detections and (ii) maximizes the time-frame of the prediction to facilitate smooth and safe maneuvers. Building on our previous work [4], [5] we will introduce a new hierarchical prediction system, that provides pedestrians' future movement in traffic scenarios.

Due to the complexity in modeling context to perform model-based predictions, we opted for a data-driven approach. Our proposed system provides inference at two different levels. First it provides the pedestrians' intention, specifically the intention to cross the street. We define this task as a

[1]Benjamin Völz is with Robert Bosch GmbH, Corporate Research, Renningen, Germany benjamin.voelz@de.bosch.com

[2] Holger Mielenz is with Robert Bosch GmbH, Chassis Systems Control, Stuttgart-Vaihingen, Germany

[3]Igor Gilitschenski is with the Computer Science and Artificial Intelligence Lab, Massachusetts Institute of Technology, Cambridge, MA, USA

[4]Roland Siegwart, and Juan Nieto are with the Autonomous Systems Lab, ETH Zurich, Zurich, Switzerland

classification problem which is solved utilizing a Support Vector Machine (SVM). The second level provides metrics that serve as qualitative descriptors of the crossing behavior. Due to the high agility of pedestrians, predicting spatial trajectories becomes quickly very uncertain. Therefore, we propose instead to predict important discrete events on these trajectories rather than the trajectory itself. In our example shown in Fig. 1 the system will predict both: the time-to-cross and the distance-to-cross. Here the second value basically represents a simplification that can be used to calculate the crossing point, which is defined as the intersection of the pedestrians' trajectory and the road boundary. We define these predictions as regression problems, which we solve with a special type of regression known as *Quantile Regression* [6]. The motivation to use this technique is that it is able to learn arbitrary conditional quantiles instead of just the conditional mean provided by standard regression algorithms. With these quantiles we are able to enrich our prediction both with minimal/maximal values and a probability density for different possible predictions.

Our evaluation will be carried out with empirical data collected at several different crosswalks in a German city. A major contribution of this work is the evaluation of the algorithms presented and a thorough analysis of their generalization performance. In particular this work aims to elucidate whether, for the particular case of pedestrian intentions at crosswalks, models learned at particular crosswalks generalize well to new ones with different configurations or in different locations.

Altogether, this work provides the following contributions:

- a hierarchical pedestrian motion prediction model,
- a new extended feature set,
- prediction of the pedestrians distance-to-cross,
- an extended evaluation which will focus on the generalization performance of the proposed algorithms.

The remainder of the paper is structured as follows. Section II shows the current state-of-the-art in the field of predicting trajectories, behavior and intentions of road users in urban traffic. Section III introduces the hierarchical prediction system and the extended features set. An overview on the pedestrian intention recognition algorithms will be presented in Section IV. Section V comprises the theoretical foundation of the *Quantile Regression* and the corresponding prediction of the time-to-cross and distance-to-cross. Section VI provides an overview of our dataset and the evaluation. Conclusions are presented in Section VI.

II. Related Work

In this section, we focus on the related work for both pedestrian path prediction and intention recognition. Recent research is primarily concerned with short-time vision-based pedestrian path predictions. These predictions are typically used for pedestrian protection systems, where the pedestrian approaches the curb orthogonally. In this scenario they predict whether the pedestrian will stop at the curb or not and therefore whether they have to perfrom an emergency brake [3], [7], [8]).

Most of the vision-based algorithms combine both the detection and prediction of pedestrians.

A seminal work that identifies the cues that human drivers use to decide whether a pedestrian will stop at the curb or not, is presented in [9]. They have shown that at least one part of the human body, either the head, the upper-body, or the legs, must be visible for a human driver to make correct predictions for the pedestrians' future movements. Consequently there has been a large number of work employing human body features. The most relevant work is reviewed in the next paragraphs.

The contour of the pedestrians' motion is used in [10] to infer their intention to cross the street. This contour includes implicitly the modeling of specific body language traits. In this case the main contributing features are the body bending and the spread of the legs. Similar approaches are presented in [7]. They show methods based both on the dense optical flow, and a low-dimensional flow-based histogram. They calculate the so called motion features, which again capture both the legs and upper-body movement. These features are then linked with the pedestrians' position to create a special trajectory representation. These enriched trajectories are then used for trajectory matching. A larger variety of body parts is used in [11], such as including arm movements, together with a sparse geometrical representation, where every body-part is depicted with a single line. A common limitation of all discussed algorithms is that they consider a very short prediction horizon of up to several hundred milliseconds. Additionally, the shown scenarios review pedestrians who are approaching the street orthogonally.

One very important feature is missing from the previously shown approaches, the pedestrians' head orientation. A sophisticated approach is presented in [8]. Here the head orientation is used to determine the pedestrians situational awareness, i.e. if the pedestrians is aware of the approaching car. The paper incorporates this measure into a Dynamic Bayesian Network (DBN) [12] and shows the additional benefit of using head tracking for improving existing prediction algorithms. While this approach is able to outperform more complex state-of-the-art algorithms, the considered time horizon is still very limited.

Apart from these vision-based systems there are other relevant approaches which utilize the pedestrians' trajectory, for example by incorporating the cartesian coordinates of the tracks. A simple approach is to use the prediction of standard tracking filters like e.g. Kalman filters for a specific dynamical model or Interacting Multiple Model (IMM) filters for the combination of different dynamics [13]. We will use such a IMM filter based prediction as basis of comparison for our evaluation in Section VI. Again in the context of collision avoidance systems, [3] models the trajectory of the pedestrian together with the approaching car to analyze their remaining time to collision (TTC) with a Bayesian Network (BN). Additionally, concerning pedestrians in an arbitrary given environment, Gaussian process regression has been used to model pedestrian trajectory patterns [14]. These patterns represent the most common paths in this specific environment. In [15] a mixture of Switching Linear Dynamics (SLD) based approach is used to identifying both low-level actions and high-level behavior patterns of object tracks. Another pattern based approach is presented in [16]. Here, both global, and

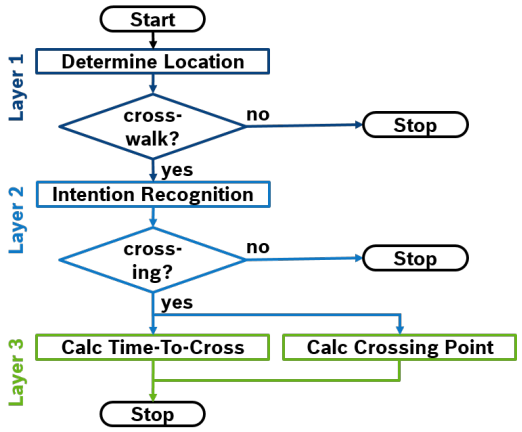


Fig. 2. Flowchart of the proposed hierarchical system architecture. First a geographic area, e.g. a crosswalk, is chosen. The second layer identifies the pedestrians intention to cross the street in the given area. Afterwards the third layer calculates relevant detailed predictions, e.g. the remaining time to cross.

local movement patterns are learned from 2D trajectories and used to predict pedestrian movements in crowds. Another approach that predicts such pedestrian movements in crowds is [17]. They utilize a Long-Short-Term-Memory (LSTM) model to learn general human movements based on hand-crafted functions that model "social forces". A long-term prediction approach is presented in [18]. In a given urban environment hand-labeled goals for pedestrian movements are defined and used together with a jump-Markov process to model their behavior.

The common factor in all the related literature is the focus on short (hundreds of milliseconds) timeframe predictions. While this is sufficient for safety systems such as collision avoidance, we aim at achieving longer prediction horizons in order to enable use of this information within comfort systems. This also enables safer interaction between pedestrians and vehicles and is a basic requirement for fully automated driving systems.

III. System Description

Predicting pedestrian movements is a highly complex task. As stated in Section I pedestrians are moving relatively slow, but very agile, i.e. they can easily change both their speed and walking direction. To address this agile movement we propose to split the problem into hierarchically orders sub-tasks. This hierarchical prediction system, as we call it, will be described in Section III-A. Additionally we will describe the feature set used within our entire inference processes in Section III-B.

A. Hierarchical Prediction System

For predicting the movement of vulnerable road users we propose a hierarchical system as depicted in Figure 2. The system contains three main layers. Within the first layer the geographical context of the given situation is selected. Possible context classes could be e.g. *crosswalk* or *intersection*. As this paper considers pedestrian motions at crosswalks, we assume the first layer to be given *a priori* and have detected a crosswalk. An example for such a detection algorithm can be found, e.g., in [19] which is based on utilizing a Dynamic Bayesian Network as described.

The second layer contains the so called intention recognition. The main task of this layer is to distinguish between crossing and non-crossing pedestrians (Section IV). The third and last main layer contains all the inferences of continuous variables which are approached with regression methods (Section V). Therefore all continuous predictions for crossing pedestrians are computed in this layer. There are two main continuous metrics that we aim to evaluate. We want to infer *when* the pedestrian will enter the street, or in other words the *time-to-cross*. And the second important metric to identify is *the location* where a pedestrian will enter the street. The combination of these two continuous values will facilitate smooth and safe manoeuvres during the interactions between vehicles and pedestrians.

B. Features

For the machine learning algorithms in the following sections a meaningful set of features is necessary. Based on our previous work [4] we will introduce a new, extended feature set.

The feature set consists of two main parts. The first part contains pedestrian features, which describe both the state estimates of the motion itself and the movement relative to the street. The other part describes the interaction with a car, namely the relative movement of the car and the pedestrian additionally to the cars state estimates and the movement along the street.

The feature set contains some additional variables which in this work are inferred using an Interactive Multiple Model (IMM) tracking filter. This tracking filter is much better suited for the tracking of agile pedestrian movements than a simple Kalman filter, which only represents one motion model.

IMM tracking filter

An IMM filter is basically a combination of several Kalman filters running in parallel [20]. Each filter represents a different motion model, typical models can be found in [21].

The IMM estimator calculates the probabilities that the observed object is moving according to each of the single Kalman filter models. These probabilities are then used to calculate a weighted sum of the state estimate of all filters. Through the combination of different movement models from the single Kalman filters it is possible to compute a more precise state estimate for any object. The utilization of different filters allows both the tracking of standard straight constant movement and any uncommon movements like sharp turns. Since the quality of the tracked state estimation, especially over these uncommon sharp turns, is of significant importance for the prediction quality (compare Section VI-E) we choose the IMM over a single Kalman filter.

For our implementation we model the pedestrians motion as a combination of constant velocity (CV) and constant acceleration (CA) with an estimate for standing pedestrians. The car tracking features a slightly different combination

of models, including a constant turn rate and acceleration (CTRA) model[1]. The IMM state estimates are directly used to calculate the following features for both the pedestrian and relevant vehicles:

- the velocity and the acceleration both in 2d coordinates and as absolute value,
- the heading,
- the distance traveled between the last and the current time step,
- the model state probabilities.

Pedestrians' movement relative to the map

The IMM position estimate of the pedestrian is used together with a map to calculate three distance measures, which describe the pedestrians' movement relative to the crosswalk. The three distances are defined as follows: dx describes the signed longitudinal distance to the center of the crosswalk. The lateral distance is conveniently named and calculated as the distance to the curb $dtcurb$. $dtcurb$ is therefore the minimal orthogonal distance to the closest curb.

$$dtcurb \begin{cases} \geq 0 & \text{if the pedestrian is on the sidewalk} \\ < 0 & \text{otherwise} \end{cases}$$

The third distance measure is the absolute, minimal distance to the crosswalk $dtcross$. This distance is always calculated relative to the closest edge of the crosswalk.

$$dtcross \begin{cases} \geq 0 & \text{if the pedestrian is in the sidewalk} \\ = 0 & \text{otherwise} \end{cases}$$

Please note, that in most of the following cases the pedestrians movement is only analyzed and predicted while she walks on the sidewalk. As soon as she enters the street it is, for obvious reasons, no longer necessary to calculate a crossing intention or e.g. a time-to-cross. Figure 3 depicts all the described features.

Car to pedestrian interaction

A vehicle position and speed can influence the movement of a pedestrian. This section introduces features to model that interaction.

Additionally to the aforementioned solely state dependent features, the position estimate of the car is again used together with the map to calculate a distance to the crosswalk:

$$dtcross_{car} \begin{cases} \geq 0 & \text{if the car has not reached the crosswalk} \\ = 0 & \text{if the car is on the crosswalk} \\ \leq 0 & \text{otherwise} \end{cases}$$

Please note that the last case should in general not be used as a feature, because the car has passed the relevant crossing area and is therefore no longer relevant. In this case either a new most-relevant or no car is selected. However the 'no relevant car in the scene' case is important for the evaluation, we it will be shown in section VI-B.

[1]Constant turn rate models are only used for cars since they describe circular (or clothoid) movements which rarely occur for pedestrians.

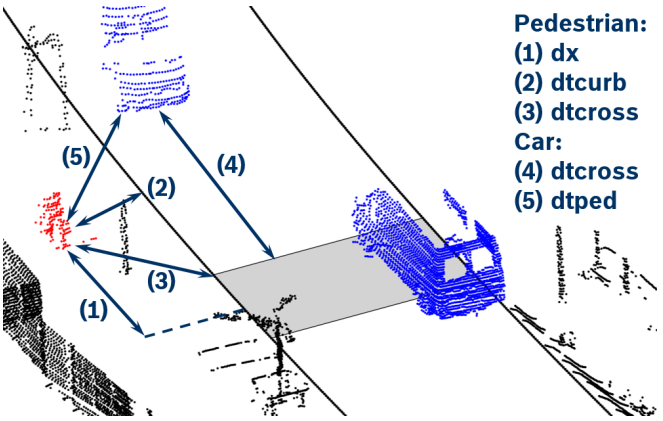


Fig. 3. All relevant distance measures for the interaction of all relevant dynamic objects both with the map and each other are shown. The underlying image shows a Velodyne Point Cloud with an sketch of the street. The two black lines mark the curbs and the grey box symbolizes the position of the crosswalk. The image contains the following Objects: cars (blue), pedestrians (red) and background (black).

Track history

Within our previous work [4] we have shown, that it is important to include the history of the features into our feature space. This improves the performance significantly, because the machine learning algorithms are now able to learn from time sequences. Therefore we include 5 time steps for every feature, i.e. instead of just $dx(t)$ we include the values: $dx(t)$, $dx(t-1)$, $dx(t-2)$, $dx(t-3)$ and $dx(t-4)$.

IV. Pedestrian Intention Recognition

As a first step our system needs to recognize the intent of pedestrians by classifying them into those who plan to cross a road at a crosswalk and those who do not intend to cross. For this we revisit the methods from our previous work [4]. Separating the intention recognition from the filtering step is on the one hand justified by the fact that most characteristics that are estimated within further processing (such as the predicted time at which the crossing starts) are not applicable or relevant for pedestrians who do not plan to cross the street. Furthermore, this classification stage serves as a data reduction procedure removing irrelevant pedestrians in the scene and therefore reducing the number of targets to be tracked. For this we will employ a nonlinear[2] Support Vector Machine (SVM). SVM's belong to the class of supervised machine learning algorithms. They have been developed for binary classification, separating a linear separable input with a maximum-margin line. By using the so called kernel trick it is also possible to perform nonlinear classification. Utilizing the kernel the nonlinear input is mapped into a high-dimensional feature space, where the input appears linear. Here, a maximum-margin hyperplane is fitted to separate the data as best as possible.

In our previous work [4] we analyzed the most relevant features for identifying pedestrians' intention to cross the street. We use an algorithm called Recursive Feature Elimination

[2]A simple test with a linear SVM produced inferior results.

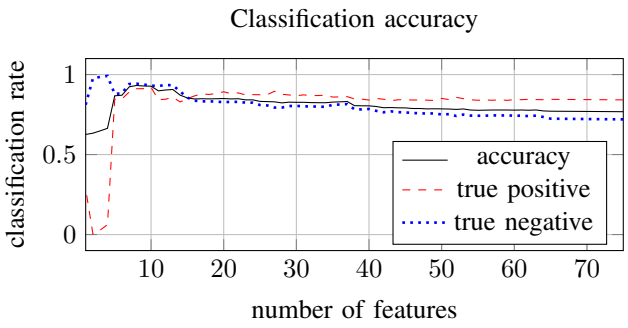


Fig. 4. Result of the single feature elimination. The classification accuracies are plotted over the number of used features. The Evaluation is carried out on a time step basis, therefore the results show the perecentage of correct classified time steps of all trajectories. Please note that this is not an evaluation on the trajectory level.

(RFE) [22]. Starting with the full feature set the RFE is an iterative algorithm which contains the following steps:

(1) Train the SVM with the current feature set.
(2) Compute the accuracy on a separate test dataset.
(3) Compute a ranking criterion for all features. One implementation could for example utilize the raw SVM weights as ranks.
(4) Remove the feature with the smallest ranking criterion. In the above mentioned example this would be the feature with the smallest weight.
(5) Repeat from (1), until all features are eliminated (no early termination).
(6) Evaluate the accuracy over all iterations.

In the last step, the relevant features are identified. The main task here is to find the best accuracy for the smallest possible feature set. Figure 4 shows the results from [4]. In addition to the accuracy we also calculate the true positive and the true negative rate, which represent the percentage of correctly classified crossing and non-crossing pedestrians.

Another possible implementation of the feature relevance estimation is the so called group elimination [22]. For this the features are combined into arbitrary groups. The algorithm is changed as follows: the ranking criterion in step (3) now computes a ranking for all groups instead of the single features, this could e.g. be the average SVM weight of all features that are part of the specific group. Accordingly in step (4) the group with the smallest ranking is removed. We used this implementation to group all time steps of our features and therefore analyze the importance of the features with their history.

Our analysis has shown, that only a small subset of the feature space is necessary to achieve satisfactory results. Altogether we only needed 10 out of the 15 features in the following groups:

- Distance to the crosswalk $dtcross$.
- Distance to the crub $dtcurb$.
- One component of the pedestrians velocity, e.g. $v_{ped,x}$.

All these features can be computed from the pedestrians track. An important finding of this analysis is the limited influence of the car to pedestrian features in the classification. However, please note that this property may vary at different countries and even different cities due to cultural differences. For instance in some countries the vehicle drivers may respect more or less the pedestrians cross-walks, and therefore people has to be less or more alert of the vehicles intentions.

V. Continuous Predictions

In this section we will introduce the general concept and our implementation for the lowest layer of our hierarchical system architecture. This layer provides detailed motion predictions for very specific situations. In our context of urban automated driving we will focus on the situations containing pedestrians about to cross the street. These pedestrians are identified with our intention recognition algorithms as described in the previous section. Therefore we will now focus on detailed, continuous motion prediction. Such continuous predictions are typically approached as trajectory or path prediction problem, where the exact trajectory is predicted for a few time steps. We claim that this procedure is not very well suited for large time horizons, since the pedestrians motion may change drastically.

Instead of this typical approach we propose to predict predefined important events with a selection of meaningful variables, that describe either the time or distance until the event starts. We want to predict when and where the pedestrian will enter the street. For this we use two main variables:

- time-to-cross: the time it will take to the pedestrian from her current position to set the first foot on the street,
- distance to cross: distance between the current position and the point where the pedestrian enters the street.

Since these variables are continuous (they change over time, when the pedestrian approaches the crosswalk) they are best approached with regression algorithms. State-of-the-art regression algorithms, like e.g. random forests, typically predict a conditional mean for the target variable. As a result of this process, other information from the probability distribution, which could provide additional helpful insights, may be lost. Therefore we decided to use a *Quantile Regression* algorithm which is able to learn the whole probability distribution and predict arbitrary conditional quantiles. The quantiles can for example be used to calculate minimal and maximal values. With additional quantiles, e.g. the median, it is possible to provide a more informative description of the likelihood of the event. Additionally the gap between the min/max values can indicate the complexity of the current situation and the action probabilities of the observed pedestrian. In our previous work [5] we compared different *Quantile Regression* algorithms and decided to use *Quantile Regression Forests*, which will be introduced in the next section.

A. Quantile Regression Forests (QRF)

QRF [23] is an extension of *Random Forests* [24]. Random Forests are an ensemble learning method that grows a large number of decision trees during training time. They can both be used for classification or regression tasks. The prediction for unseen examples can be made by majority vote (for classification) or averaging the prediction of all trees (for regression). The best results are obtained when single trees are not correlated, because then averaging reduces individual tree

uncertainty. This is because the prediction of a single tree is highly sensitive to noise, but the average of many trees is not, as long as the trees are not correlated. To achieve this, *Random Forests* utilize two techniques. First, tree bagging is used to select a random sample of the training set for every tree. Additionally, for every tree a random subset of the features is used. This method is known as *random subspace method* or *feature bagging* [24]. Both the size of the random subset $mtry$ and the number of trees to grow $ntree$ are tunable parameters of the *Random Forests*.

A typical regression *Random Forest* calculates and stores the average observation for every leaf of every tree. The main difference for QRF is that in every leaf of every tree all relevant observations are stored, not just their average. With this information the full conditional distribution can be assessed [23]. Altogether the training of a QRF is straight forward: grow $ntree$ trees just like in *Random Forests*, but instead of storing the average observations in a leaf, store all observations.

To compute the prediction of a QRF and therefore compute an arbitrary conditional quantile for a new data point $X = x$ first the average weights $w_i(x)$ of every observation i over all trees of the random forests has to be calculated as described in [23]. These weights can be used to compute the estimate of the cumulative distribution function $\hat{F}$, which can be defined as:

$$\hat{F}\left(y|X=x\right) = \sum_{i=1}^{n} w_i(x) 1_{\{Y_i \leq y\}}.$$

Now we can calculate the estimate of the conditional quantile $Q_\alpha(x)$ for any α, with $0 < \alpha < 1$,

$$Q_\alpha(x) = \inf\left\{y : \hat{F}\left(y|X=x\right) \geq \alpha\right\}.$$

B. Time-To-Cross

One of the two variables we want to predict is the time-to-cross. It can be defined as the time which the pedestrian needs to move from his current position along his trajectory to the point where he enters the street. Our database, which will be introduced in Section VI-B, contains full trajectories. Therefore this time can be calculated for every point of every trajectory and accordingly used for both training and testing. In our previous work [5], we predicted this time measure with a carefully tuned QRF. In this paper we will extend the evaluation by analyzing the generalization performance of the algorithm for larger datasets and additional unique crosswalk geometries (Section VI).

C. Distance-To-Cross

Additionally to the time-to-cross we want to infer the location where the pedestrian is most likely to step on the street. This point is a position in our 2D global coordinate frame. The prediction of two dependent coordinates requires to explicitly model that dependency, which adds complexity. To simplify the inference process we project the trajectories onto a 1D representation (Figure 5). We want to predict the point where the pedestrian will cross the curb and enter the street. So

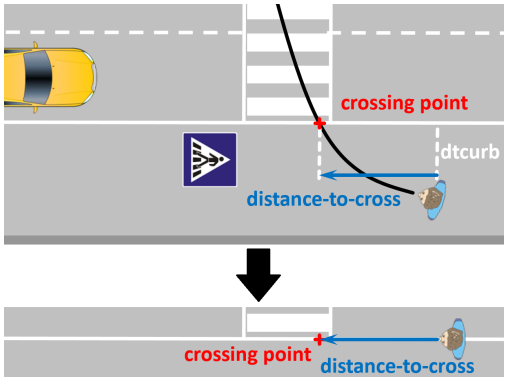


Fig. 5. Definition of the crossing distance label. The 2D problem in the global coordinate frame can be projected into a 1D representation, because the distance to the curb $dtcurb$ is known. If the distance-to-cross is known, it is easily possible to calculate the corresponding *crossing point* in the global coordinate frame.

basically we want to predict the intersection of the pedestrians trajectory with the roadside. Due to our digital map and the previously calculated features, we already know both a 2D line which represents the roadside and the pedestrians' distance to the curb $dtcurb$. Since, by definition, $dtcurb$ was calculated as the "minimal orthogonal distance to the closest curb", we also know the position of the pedestrian projected onto the 2D borderline of the road. With all these information we can project our problem into the 1D representation. Our prediction problem gets reduced to a regression where we try to predict a distance-to-cross, defined as the 1D distance between the current position and crossing point. Accordingly, it is now possible to calculate the crossing point, if both the current position and the distance-to-cross are known.

VI. Evaluation

Our evaluation is composed by two main parts. Before we start with the evaluation itself, the metrics employed will be discussed in Section VI-A, followed by a description of the datasets in Section VI-B.

In the first part of the evaluation will analyze the performance of our algorithms with our largest dataset, which was recorded at one specific crosswalk. For this we will perform cross validation.

Afterwards, we will analyze the generalization performance by testing the resulting model at different crosswalks. The differences arise mainly from the geometry of the crosswalk and the surroundings (Section VI-D). This section will especially analyze the level to which a model generalization might be possible.

Finally in Section VI-E we discuss the overall remaining challenges, which limit the performance in general.

A. Baseline and Evaluation Metric

We aimed to design an algorithm that is capable of doing long-term predictions. Our pipeline contains both a classification and an regression part. For classification problems the time horizon is usually evaluated based on the time-to-collision, time-to-curb, or comparable. Because of the large

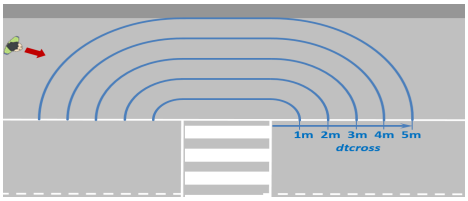
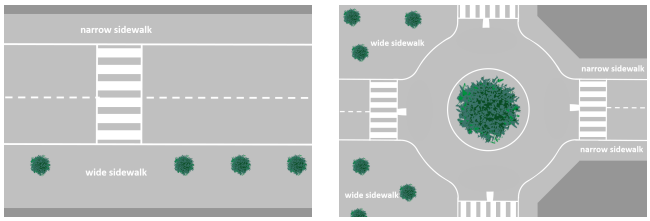


Fig. 6. The distance-based evaluation principle is shown. All further evaluations will provide performance measures relative to the pedestrians distance to the crosswalk $dtcross$ as a measure for the prediction horizon.



(a) Straight road. (b) Roundabout.

Fig. 7. Visualization of different road and crosswalk geometries. The road shape is either (a) straight or (b) a roundabout. The images also depict the different possible sidewalk sizes (narrow or wide).

amount of *non-crossing* pedestrians that neither cross the street nor the path of a relevant car, it is difficult to calculate a sophisticated time measure without biasing the results by the own beliefs. I.e. we could always calculate the time-to-cross for the worst case scenario by taking the minimum distance to the crosswalk together with a high velocity. This calculation would result in a highly conservative time measure which is not suited to represent the real world scenarios, since it only represents the minority of high-risk situations. Therefore we decided to evaluate the prediction horizon for our classification problems differently. We evaluate our performance relative to the pedestrians distance to the crosswalk $dtcross$. The general idea is presented in Figure 6.

For our classification problems we use the prediction of our IMM tracking filter from Section III-B as a baseline for comparison. To provide a functionality equal to our SVM we create a new IMM for every track and frame based on the same 5 time steps used by the SVM. To avoid any problems or inaccuracies caused by the transient we use the available frames to calculate proper state estimates and initialize the IMM's and their models accordingly. The IMM's are then used to predict the state of every single frame for up to 10 seconds. The prediction time of the IMM is chosen deliberately high to assure that the prediction horizon is definitely longer than the actual time-to-cross. The resulting predicted trajectories are then checked for "collision" with the crosswalk and the predicted class (*cross* or *non-cross*) is inferred accordingly.

B. Dataset

Our database contains car and pedestrian tracks recorded with a Velodyne laser scanner. The raw point cloud is processed according to [25]. This includes: the segmentation of the point cloud into arbitrary objects, the tracking of these objects over time and a classifier that issues one of four class labels: car, pedestrian, bicyclist or background. The classifier consists of a nonlinear multiclass SVM trained and validated on the Stanford Track Collection (STC). Figure 3 shows a preprocessed point cloud.

Every track is associated with a precise digital map, which describes the static, urban environment, i.e. road boundaries, crosswalk positions and more.

Our database consists of several datasets recorded as different crosswalks as depicted in Figure 7. The two main attributes that distinguish these geometries are the road shape and the size of the sidewalk. The road shape can be either a straight with a crosswalk or a roundabout. Usually, if there is a crosswalk at a roundabout, there are many. For our evaluation we discretized the sidewalk size into qualitative groups (narrow, wide). By combining these attributes combinatorially we get 4 (2 by 2) different geometries which are used in Section VI-D for the generalization tests.

All data driven modules utilized in our pipeline are supervised learning methods. Therefore, both track and frame labels are needed. This is easily done, since the whole track is known. First we infer a label for crossing and non-crossing pedestrians. Additionally, we want to make detailed predictions for all crossing pedestrians, therefore we also infer time-to-cross and distance-to-cross values for all relevant frames. This automatic labeling procedure has some disadvantages which will be analyzed and evaluated in Section VI-E.

C. Cross Validation

The first part of our evaluation focuses on the overall algorithm performance under nearly ideal conditions. We will show the performance for the case, where both train and test data are recorded at the same crosswalk. The single datasets are still independent, because they were recorded at different days and times. For a real world implementation this resembles the most expensive but also most reliable case, where a model is learned for every single crosswalk. The high costs arise primarily for two reasons: A large dataset has to be recorded and labeled for every single crosswalk and a model has to be stored and, if applicable, transmitted to a vehicle, whenever it visits a new location.

We will perform a 5-fold cross validation on our largest single dataset with roughly 2000 pedestrian trajectories with 100000 time frames. Concerning the regression problems we will only analyze the results of the distance-to-cross predictions. Qualitatively the time-to-cross prediction works similar and can be found in [5].

Intention Recognition

Compared with our previous work [4], the performance reported here is slightly better. This is mainly possible due to a more precise labeling process and the elimination of confusing trajectories from our training data. One example for such a confusing trajectory is shown in Figure 8 where a pedestrian moves in several circles before crossing the road. These trajectories will be analyzed further as part of the remaining challenges in Section VI-E.

Figure 9 shows the classification results for the SVM and our IMM baseline relative to the pedestrians' distance to the

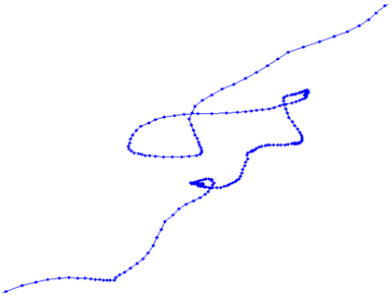
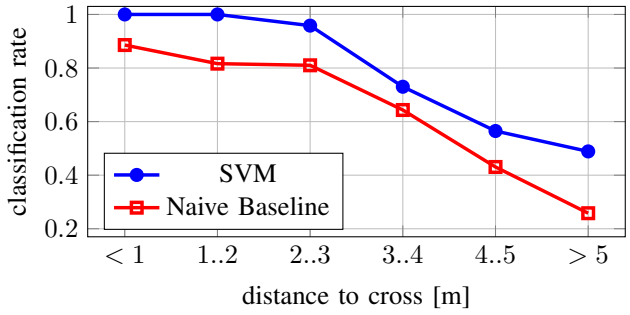
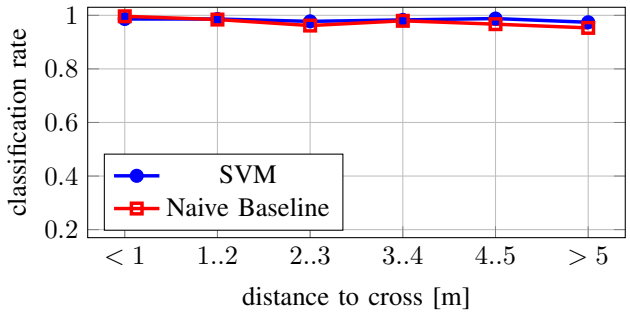


Fig. 8. Trajectory of a pedestrian moving in several circles before moving towards the road (road not shown). The trajectory starts in the lower left corner and visualizes each measurement as one blue dot. Such trajectories show confusing behavior that is almost impossible to label properly and could deteriorate the training performance significantly. Therefore they are removed from the training set and only used for the evaluation and the analysis of remaining challenges in Section VI-E.



(a) Crossing pedestrians (True Positive)



(b) Non-crossing pedestrians (True Negative)

Fig. 9. SVM classification results compared to a simple decision based on the IMM tracking filter prediction. The accuracy is shown both for (a) crossing and (b) non-crossing pedestrians. The results are shown relative to the pedestrians distance to the crosswalk to provide an impression for the prediction horizon.

crosswalk. Both algorithms show an overall good performance for all non-crossing scenarios. However SVM outperforms the IMM prediction by 10-20% in correcting classifying crossing pedestrians.

The performance's decline for large $dtcross$ values can be understood by analyzing the typical pedestrian movements in these area (Figure 10). For this crosswalk a large amount of the non-crossing pedestrians move parallel to the street with a $dtcrub$ of at least 3m. This results in the observed high accuracy for non-crossing pedestrians. Additionally, we can observe that a large amount of crossing pedestrians will walk parallel to the street before doing a late turn towards the

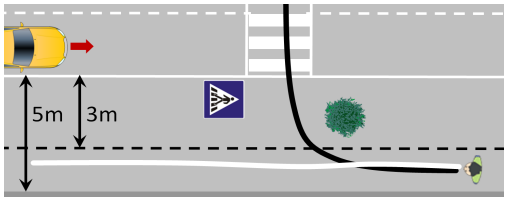


Fig. 10. Typical trajectories at a crosswalk. White: Trajectory of a pedestrian, who passes the crosswalk with a constant $dtcurb$ of $3-5$m. Black: Crossing pedestrian, who is walking parallel to the street for a long time before turning towards the crosswalk. Since both trajectories are more or less parallel at their beginning, they are almost indistinguishable and result in most of the false classifications in this area.

TABLE I
AVERAGE CROSS VALIDATION RESULTS FOR THE QRF BASED DISTANCE-TO-CROSS PREDICTION. BOTH THE PERCENTAGE OF CORRECTLY PREDICTED TIME STEPS AND THEIR CORRESPONDING INTERVAL SITE ARE SHOWN RELATIVE TO $dtcross$.

$x = dtcross$ [m]	Regression Accuracy	Interval Size
all	84.74%	
$0 < x \leq 1$	75.86%	0.26m
$1 < x \leq 2$	90.73%	0.72m
$2 < x \leq 3$	80.65%	0.81m
$3 < x \leq 4$	88.54%	1.90m
$4 < x \leq 5$	92.80%	3.61m
$x \geq 5$	79.90%	3.20m

crosswalk. These two behaviors are inseparable for most large distance measures, which results in a poorer performance accuracy.

Distance-to-cross

As mentioned before we will show the performance of the regression algorithms exemplary with the distance-to-cross prediction. Drawing on the theory presented in Section V-C, Table I provides a quantitative representation of both the percentage of correctly predicted time steps and the size of the corresponding intervals relative to the pedestrians' $dtcross$. In this case a prediction is marked as *correct*, if the observed value (ground truth) lies within the predicted interval. This is also the reason, why it is important to additionally analyze the corresponding interval size. The shown result is an average of the cross validation results. In general the accuracy is very stable with values between 80 and 90%. The main difference is given by the interval size that is necessary to achieve this accuracy. For distances of up to 3 meters the predicted crossing point has an associated interval size of ≤ 81 cm. The interval size' variance increases for larger distances which represents the difference between a pedestrian who cuts the street to get to the crosswalk faster and a pedestrian who does a late turn after moving parallel to street. A small dip in the accuracy occurs for the pedestrians walking very close to the crosswalk. The cause of this is a small amount of overly careful pedestrians, who stop at the sidewalk until all cars are either gone or have stopped. While waiting they often move sideways which for our models is an unexpected behavior and causes false predictions due to the very tight interval.

D. Generalization Test

One of the main contributions of this paper is the analysis of the generalization performance of our algorithms for a number of different crosswalks. The crosswalks differ mainly in the road geometry (see Figure 7). We analyze the influence of both the shape of the street itself (straight or roundabout) and the sidewalk width on our prediction performance. For this we recorded data at four different crosswalks, with the following characteristics. Our main crosswalk, known from the previous sections, is characterized by a straight street with a quite wide sidewalk, with a width of up to 5m. This crosswalk is used to train the prediction model. The performance measures which we will provide for this crosswalk are taken from Section VI-C and define the *baseline* for comparisons.

The second crosswalk has the same geometry only with a much *narrower* sidewalk. Depending on the specific location the width of this sidewalk is between 2m and 3m. The remaining two datasets belong both to crosswalks at roundabouts. One roundabout (*round1*) has an adjacent large square and the other (*round2*) a mid-size sidewalk.

Table II shows the true positive and true negative prediction accuracy for an intention prediction at these crosswalks. For the *narrow* crosswalk one can easily see, that the performance is quite poor. Especially the prediction performance for all crossing pedestrians (43.6% for all combined frames). This was not unexpected, since the results show that the width of the sidewalk has indeed a large influence on the prediction performance, especially for crossing pedestrians. If we on the other hand take a look on the non-crossing pedestrians, we can see that the performance improves. The reason for the large amount of correctly classified non-crossing pedestrians can be identified, when taking a closer look on the single trajectories. During the evaluation of these trajectories we have seen, that the majority of the non-crossing pedestrians show an identical behavior for both crosswalks, which can be characterized by one simple rule: the pedestrians who are not crossing and moving parallel to the street try, if possible, to always keep a safe distance to the curb. In this context, a safe distance can be seen as the largest possible distance, that allows a comfortable walk. Such behavior can also be observed for many crossing pedestrians. These pedestrians are then also walking parallel to the crosswalk before doing a late turn towards it. This results in almost the same problem we discussed earlier in the cross-validation. We only have one important difference. Due to the narrower sidewalk the described late turns appear much closer to the crosswalk (see Figure 11), which results in a poor performance over all distances.

If we now take again a look at Table II, we can analyze the influence of the street layout itself. Namely the difference between a straight and a roundabout. For the first roundabout *round1* we see, that the overall performance is comparable to the *baseline* for all values in the area $0 < dtcross \leq 4$m. The main reason for this good performance can be found in the similarities between the large square at the roundabout and the large sidewalk in the model. The behavior of pedestrians in both cases is similar. One important question remains: why does the performance for crossing pedestrians drop for $dtcross > 4m$. The roundabout replaces an intersection with crosswalks on all connected lanes (4 in total). These other crosswalks are not present in the training data. The results show that they must possess un-modelled effects in the pedestrian trajectories.

TABLE II

INTENTION RECOGNITION GENERALIZATION TEST FOR DIFFERENT CROSSWALKS GEOMETRIES. THE RESULTS FROM SECTION VI-C ARE USED AS A *basis* FOR COMPARISON. THE OTHER EXAMINED CROSSWALKS ARE: A CROSSWALK WITH A VERY *narrow* SIDEWALK ($\leq 2m$), A CROSSWALK AT A ROUNDABOUT WITH AN ADJACENT LARGE SQUARE (*round1*) AND A SECOND ROUNDABOUT WITH A MID SIZE SIDEWALKS (*round2*).

$x = dtcross$	True Positive			
[m]	*base*	*narrow*	*round1*	*round2*
all	82.02%	43.60%	73.58%	62.16%
$0 < x \leq 1$	99.99%	52.49%	99.99%	96.47%
$1 < x \leq 2$	99.99%	60.46%	99.99%	95.97%
$2 < x \leq 3$	95.83%	50.51%	98.85%	71.11%
$3 < x \leq 4$	73.01%	28.87%	90.34%	37.95%
$4 < x \leq 5$	56.49%	23.38%	67.46%	17.39%
$x \geq 5$	48.87%	24.97%	10.06%	15.23%
	True Negative			
all	98.15%	85.06%	88.47%	94.25%
$0 < x \leq 1$	98.63%	81.51%	99.99%	70.36%
$1 < x \leq 2$	98.62%	78.78%	89.32%	71.08%
$2 < x \leq 3$	97.74%	80.44%	84.05%	86.88%
$3 < x \leq 4$	98.29%	80.42%	78.28%	95.27%
$4 < x \leq 5$	98.79%	87.15%	74.46%	97.11%
$x \geq 5$	97.40%	95.10%	95.37%	96.63%

The last column of Table II shows the results for a crosswalk at a roundabout with a mid size sidewalk. The performance for large distances suffers also from the presence of other crosswalks. Because of the special geometry of this roundabout which features 6 connecting lanes instead of 4, the effect occurs earlier on (for $dtcross \geq 3$m). For all other cases we can see, that although the performance is inferior compared to the first roundabout, it is still acceptable. In general the performance suffers from the same problem as in the *narrow* scenario, but the impact is significantly lower.

Altogether we can summarize the following findings. Regarding the influence of the road shape, we were not able to identify a difference between a straight and a roundabout for most cases. The main difference arises due to the other nearby crosswalks. The presence of these crosswalks is generally given by definition, if a roundabout features one crosswalks. Secondly, the results show that the main problem that limits the generalization performance of our approach is the sidewalk width. We have seen at several examples that the prediction accuracy degrades with decreasing size, but we have also seen that it is possible to make better predictions when the sidewalk widths are comparable.

E. Remaining Challenges

Additionally to the previously described findings, we want to provide some insights on the more general problems we found, which are limiting the prediction performance. Al-

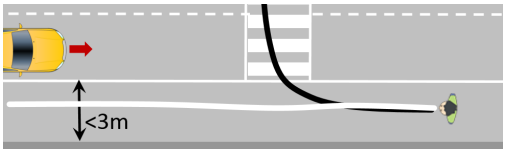


Fig. 11. Typical trajectories at a *narrow* crosswalk. White: Trajectory of a pedestrian, who passes the crosswalk with a constant $dtcurb$ of $1-2$m. Black: Crossing pedestrian, who is walking parallel to the street for a long time before turning towards the crosswalk. Since both trajectories are more or less parallel at their beginning, they are almost indistinguishable and result in most of the false classifications in this area.

though some of the problems may be unique to our combination of tracking, labeling and prediction, they all have underlying difficulties, which will potentially limit the performance of any prediction system. Apart from the typical errors which result from poor training, either due to outliers, missing data, or inappropriate or badly tuned algorithms, we identified additional error sources within atypical pedestrian trajectories. These trajectories can be characterized usually with at least one of the following points:

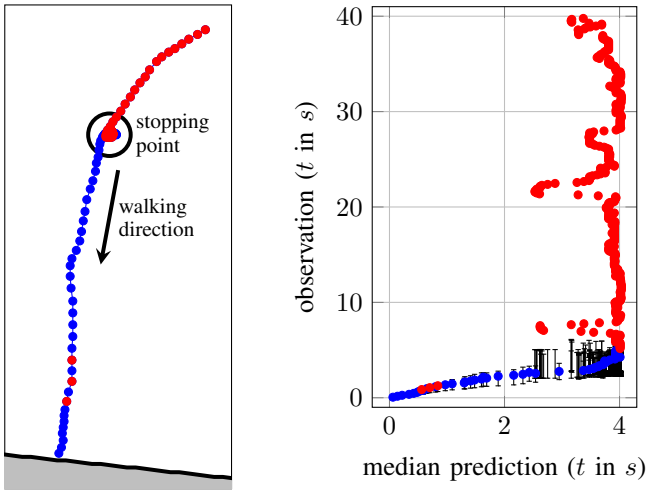
- high accelerations (or decelerations),
- sharp turns,
- stopping, usually combined with some movement on the spot.

To explain the problems, we first should recall the previously described automatic labeling procedure VI-B. We are doing both offline training and testing, therefore we can assume that all tracks are known. Hence we know, if a pedestrian in our database has crossed the street and, if applicable, where and when she has crossed it. Therefore we can infer labels for each time step according to the observed event. Even though this method has the advantage of being automated, it can produce systematic errors in combination with the above-mentioned pedestrian behavior. We will illustrate this problem with some figures from the QRF based time-to-cross evaluation.

Figure 12 depicts a pedestrian who will cross the street, but suddenly stops and waits at the roadside for several seconds. Since our automatic labeling framework is not able to detect this stop, our algorithm provides a theoretically wrong prediction (Figure 12(b)). However, if we take a closer look at the exact prediction, we can see, that during the whole standing time, the prediction estimates a remaining crossing time of approximately 4s, which would be the correct prediction, if the pedestrian would immediately starts to move[3]. If we combine this prediction with a detector for standing pedestrians (e.g. the IMM tracker from Section III-B), the prediction remains useful as it provides an estimate for the case that the pedestrian starts moving again. I.e. we could treat this prediction as a "what if" scenario: What would happen, if the pedestrian would immediately start to move towards the crosswalk? In this case we can ignore the prediction as long as our IMM tracker detects the pedestrian as stationary. The main challenge in this scenario is given by our main goal of detecting the pedestrians movement as early as possible and predicting with the longest time horizon possible.

A different example which illustrates the combined error due

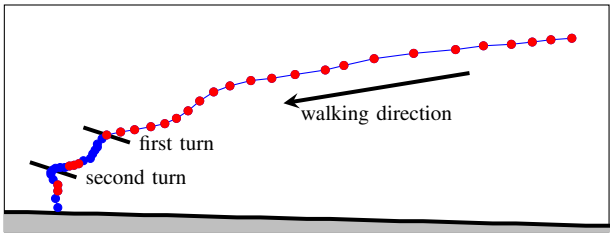
[3]Please note: the prediction is a bit noisy around the standing area. The reasons for this is, that the pedestrian is not standing perfectly still but significantly moving on the spot.



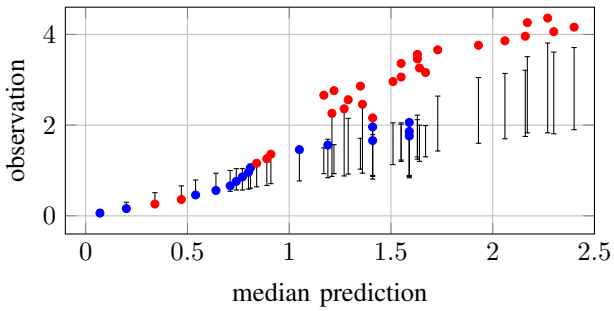
(a) Trajectory: the stopping location is marked with a circle. Every dot represents one time step of the trajectory and is marked either in blue for a correct prediction or red otherwise. The road is depicted at the bottom of the image.

(b) QRF prediction: the red dots represent the globally wrong measurements (observed time $\gg$ predicted time) and blue the correct ones. The prediction represents the time a pedestrian would need to cross the street if he would continue walking in similar manner. The prediction can therefore be seen as locally correct.

Fig. 12. Trajectory and resulting QRF prediction for a pedestrian temporary stopping at the roadside.



(a) Trajectory: every dot represents one time step of the trajectory and is marked either in blue for a correct prediction or red otherwise. The road is depicted at the bottom of the image.



(b) QRF prediction: the red dots represent the globally wrong measurements. The prediction both before the first turn has globally large errors, but correctly represents the time-to-cross if the pedestrian would continue walking with the same high speed. After the second turn the pedestrian again accelerates which results in a shortly wrong prediction.

Fig. 13. Trajectory and resulting QRF prediction for a pedestrian doing several sharp turns and repeatedly changing her velocity.

to high acceleration and sharp turns is shown in Figure 13. This example features a pedestrian who is firstly running towards the crosswalk. The high velocity can be seen indirectly by means of the large gaps between two track frames in the x-y coordinate frame in Figure 13(a). The pedestrian then quickly decelerates and reaches the crosswalk after a series of sharp turns. As we can see, all frames before the first turn are marked as wrong. If we additionally consider the corresponding prediction (Figure 13(b)), we can again see that, although labeled as wrong, we got exactly the prediction which we need in a real environment. For the beginning of the track our algorithm predicts a time-to-cross of 1s to 2.5s for observed crossing times of 2s to 4s. Since there is no evidence for either the change of speed or walking direction before the first turn, our algorithm provided the correct prediction, which was that the pedestrian will continue running and reach the crosswalk much earlier. If we now take a closer look on the remaining trajectory after the first turn, we can see that our algorithm adapts very quickly to the new circumstances (new velocity and changed walking direction). Immediately after the first turn we receive correct predictions with reasonable uncertainties. The remaining errors are caused by minor deviations between the prediction and the observation.

The majority of false predictions in our results are produced by large accelerations and sharp turns. In the evaluated cases we have shown that our algorithms are capable of providing a locally correct prediction. We claim that the biggest challenge for any long-time prediction system is the fast adaptation to movement changes. The faster we are able to detect these changes the earlier it is possible to compute a reasonable prediction for the changed circumstances. This of course is only partly a prediction problem. The performance is naturally heavily dependent on the quality of the underlying tracking-system.

Finally we want to address one more challenge which can also be illustrated with Figure 12. The depicted scene features a pedestrian who stops near the crosswalk, but still has a large $dtcurb$. Let's consider the same scenario, but with a pedestrian who stops on, or very near to, the curb. Now if we additionally take into account that the car will approach the crosswalk after the pedestrian has stopped[4]. With our current system, and especially with our current feature set, we will not be able to predict reliably, if the pedestrian will cross the street or not. For this scenario we would need additional information on the pedestrians orientation, e.g. using the pedestrians' heading based on his upper body position [8].

F. Computation Complexity

Finally we want to discuss the computation complexity of the used algorithms and therefore the real time capabilities of our hierarchical approach. The estimated evaluation time for a single pedestrian and frame is shown in Table III. For this evaluation we used a single 2.4 GHz core of a standard laptop. Please note: due to the hierarchical prediction system, the more demanding continuous prediction (QRF, compare Section V) is only evaluated for actually crossing pedestrians. In our unbalanced raw data we have around 20% crossing pedestrians. The results show a low combined computation time that is real time capable, even if multiple pedestrians have to be predicted.

[4]This means we have not seen how the pedestrian has approached, i.e. whether she already has crossed the road, or is waiting for all cars to stop.

TABLE III
ANALYSIS OF THE COMPUTATION TIME AND THE CORRESPONDING NUMBER OF PARAMETERS FOR EACH ALGORITHM. THE TIME IS ALWAYS CALCULATED FOR ONE PEDESTRIAN AND ONE FRAME. FOR THIS TIMING ESTIMATION ALL ALGORITHMS RAN ON A SINGLE 2.4 GHZ CORE OF A STANDARD LAPTOP. THE AMOUNT OF ACTUALLY CROSSING PEDESTRIANS IN THE RAW DATABASE IS 20%. THEREFORE THE ESTIMATED COMBINED **MEAN** TIME OF SVM AND QRF IS CALCULATED AS: TIME OF SVM + 0.2 * TIME OF QRF. AS PARAMETERS ONLY THE NON-ZERO ONES ARE COUNTED.

Algorithm	t [ms]	Parameters
SVM	1.46	19,110
QRF	12.52	10,000
combined mean	3.96	
combined max	13.98	

Considering an input (perception) cycle of 10 Hz (100 ms) we are able to predict up to 7 actually crossing pedestrians (max calculation time for crossing pedestrians: 13.98 ms) or theoretically up to 25 pedestrians in general (mean calculation time: 3.96 ms). The presented approach can by design be parallelized, and therefore also evaluate more objects, if required.

VII. CONCLUSION

In this paper we introduced a holistic prediction model for pedestrians crossing the street in urban environments. The model has a hierarchical structure that utilizes different machine learning algorithms for different sub-problems. First we used an SVM to predict the pedestrians' intention to cross the street. Afterwards, for all identified crossing pedestrians, we focused on providing a more detailed prediction of specific important events on the future trajectory of these pedestrians. Namely we used *Quantile Regression* to predict both the pedestrians time-to-cross and crossing point with uncertainty.

In the evaluations, we have shown how the proposed approach generalizes, training a model at one crosswalk and testing it at another. We analyzed the performance relative to specific crosswalk types which mainly differ in their geometric shape. The crosswalk geometry can be characterized both by the shape of the road (straight or roundabout) and the size of the corresponding sidewalk (narrow or wide). During our evaluation we showed that we are able to provide good predictions for all described sub-problems, if we are able to train our model with data from the same or at least a geometrically similar crosswalk. Although it is possible to create a model for similar crosswalks, we found that our approach cannot guarantee to hold its performance among crosswalks with largely differing geometry. Altogether we can conclude, that we are able to predict pedestrians' movements in urban environments with a small amount of models trained for specific unified road geometries.

REFERENCES

[1] S. Lefevre, D. Vasquez, and C. Laugier, "A survey on motion prediction and risk assessment for intelligent vehicles," in *ROBOMECH Journal*, 2014, **1**:1.

[2] G. Agamennoni, J. I. Nieto, and E. M. Nebot, "Estimation of multivehicle dynamics by considering contextual information," *IEEE Transactions on Robotics*, vol. 28, no. 4, pp. 855–870, 2012.

[3] C. Braeuchle, J. Ruenz, F. Flehmig, W. Rosenstiel, and T. Kropf, "Situation analysis and decision making for active pedestrian protection using bayesian networks," in *Proc. of the 6. Tagung Fahrerassistenz, München*, 2013.

[4] B. Völz, H. Mielenz, G. Agamennoni, and R. Siegwart, "Feature relevance estimation for learning pedestrian behavior at crosswalks," in *IEEE Int. Conf. on Intelligent Transportation Systems (ITSC)*, 2015.

[5] B. Völz, H. Mielenz, R. Siegwart, and J. Nieto, "Predicting pedestrian crossing using quantile regression forests," in *Proc. of the IEEE Intelligent Vehicles Symposium (IV)*, 2016.

[6] R. Koenker, *Quantile Regression*. Cambridge University Press, 2005.

[7] C. G. Keller and D. M. Gavrila, "Will the pedestrian cross? a study on pedestrian path prediction," *IEEE Trans. Intell. Transp. Syst.*, vol. 15, pp. 494–506, 2013.

[8] J. Kooij, N. Schneider, F. Flohr, and D. Gavrila, "Context-based pedestrian path prediction," in *Proc. of the European Conference on Computer Vision (ECCV)*, 2014.

[9] S. Schmidt and B. Färber, "Pedestrians at the kerb - recognising the action intentions of humans," *Transportation Research Part F: Traffic Psychology and Behaviour*, vol. 12, pp. 300–310, 2009.

[10] S. Köhler, M. Goldhammer, S. Bauer, S. Zecha, K. Doll, U. Brunsmann, and K. Dietmayer, "Stationary detection of the pedestrian's intention at intersections," *IEEE Intell. Transp. Syst. Mag.*, vol. 5, pp. 87–99, 2013.

[11] R. Quintero, J. Almeida, D. F. Llorca, and M. A. Sotelo, "Pedestrian path prediction using body language traits," in *Proc. of the IEEE Intelligent Vehicles Symposium (IV)*, 2014.

[12] K. P. Murphy, "Dynamic bayesian networks: Representation, inference and learning," Ph.D. dissertation, University of California, Berkeley, USA, 2002.

[13] N. Schneider and D. M. Gavrila, *Pedestrian Path Prediction with Recursive Bayesian Filters: A Comparative Study*. Springer Berlin Heidelberg, 2013, pp. 174–183.

[14] D. Ellis, E. Sommerlade, and I. Ried, "Modelling pedestrian trajectory patterns with gaussian processes," in *IEEE 12th International Conference on Computer Vision (ICCV) Workshops*, 2009.

[15] J. F. P. Kooij, G. Emglebienne, and D. M. Gavrila, "Mixture of switching linear dynamics to discover behavior patterns in object tracks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 38, pp. 322–334, 2016.

[16] A. Bera, S. Kim, T. Randhavane, S. Pratapa, and D. Manocha, "Glmp - realtime pedestrian path prediction using global and local movement patterns," in *IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2016.

[17] A. Alahi, K. Goel, V. Ramanathan, A. Robicquet, L. Fei-Fei, and S. Savarese, "Social lstm: Human trajectory prediction in crowded spaces," in *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[18] V. Karasev, A. Ayvaci, B. Heisele, and S. Soatta, "Intent-aware long-term prediction of pedestrian motion," in *IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2016.

[19] T. Gindele, S. Brechtel, and R. Dillmann, "Learning context sensitive behavior models from observations for predicting traffic situations," in *IEEE Int. Conf. on Intelligent Transportation Systems (ITSC)*, 2013.

[20] X. R. Li and V. P. Jilkov, "Survey of maneuvering target tracking. part v: Multiple-model methods," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 41, pp. 1255–1321, 2005.

[21] R. Schubert, E. Richter, and G. Wanielik, "Comparison and evaluation of advanced motion models for vehicle tracking," in *11th Int. Conf. on Information Fusion*, 2008.

[22] I. Guyon, J. Weston, S. Barnhill, and V. Vapnik, "Gene selection for cancer classification using support vector machines," *Journal of Machine Learning Research*, vol. 3, pp. 1439–1461, 2003.

[23] N. Meinshausen, "Quantile regression forests," *Journal of Machine Learning Research*, vol. 7, pp. 983–999, 2006.

[24] L. Breiman, "Random forests," *Machine Learning*, vol. 45, pp. 5–32, 2001.

[25] A. Teichmann, J. Levinson, and S. Thrun, "Towards 3d object recognition via classification of arbitrary object tracks," in *IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2011.

Benjamin Völz Benjamin Voelz is both a Research Engineer with Bosch Corporate Research, Germany and a Ph.D. candidate with the Autonomous Systems Lab at ETH Zurich, Switzerland. He received a Dipl.-Ing. degree in electrical engineering from Dresden, Technical University, Germany in 2013. His research interests include situational awareness, decision making and behavior planning for autonomous vehicles.

Holger Mielenz Holger Mielenz is Senior Project Manager for urban automated driving at Bosch Chassis Systems Control, Germany. He received his Ph.D. in computer science from the University of Tbingen, Germany. His research interests focus on system engineering and concept development for automated driving.

Igor Gilitschenski Igor Gilitschenski is a Senior Postdoctoral Associate within the Computer Science and Artificial Intelligence Lab (CSAIL) at the Massachusetts Institute of Technology (MIT). Prior to that, he worked as a Senior Researcher with the Autonomous Systems Lab at the Swiss Federal Institute of Technology (ETH) Zurich. Dr. Gilitschenski received his Ph.D. degree from the Institute of Anthropomatics and Robotics at the Karlsruhe Institute of Technology (KIT) in 2015. Before joining KIT, he obtained a diploma degree in mathematics from the University of Stuttgart. His research interests include probabilistic and statistical methods for robotic perception and learning in complex dynamic environments.

Roland Siegwart Roland Siegwart is full Professor of Autonomous Systems at ETH Zurich since July 2006 and Founding Co-Director of the Wyss Zurich. From January 2010 to December 2014 he took office as Vice President Research and Corporate Relations in the Executive Board. He received his Diploma in Mechanical Engineering in 1983 and his Doctoral Degree in 1989 from ETH Zurich. Roland Siegwart's research interests are in the design and control of systems operating in complex and highly dynamical environments. His major goal is to find new ways to deal with uncertainties and enable the design of highly interactive and adaptive systems. Prominent application examples are personal and service robots, autonomous micro-aircrafts, walking and swimming robots and driver assistant systems.

Juan Nieto Juan Nieto is Deputy Director at the Autonomous Systems Lab, ETH Zurich, Switzerland. Before that he was a Senior Research Fellow at the Australian Centre for Field Robotics, the University of Sydney, Australia. He received his Ph.D. from the University of Sydney. His main research focus is on perception and navigation for mobile robots.