DISS. ETH NO. 26578

Learning to Predict Pedestrians for Urban Automated Driving

A thesis submitted to attain the degree of

DOCTOR OF SCIENCES of ETH ZURICH

(Dr. sc. ETH Zurich)

presented by

Benjamin Völz

Dipl.-Ing.

born on February 15, 1989
citizen of Leonberg, Germany

accepted on the recommendation of

Prof. Dr. Roland Siegwart, Examiner
Prof. Dr. Margarita Chli, Co-examiner
Dr. Rüdiger Walter Henn, Co-examiner

2020

Autonomous Systems Lab
Department of Mechanical and Process Engineering
ETH Zurich
Switzerland

Abstract

Navigating through densely populated urban areas is one of the most important challenges for self-driving vehicles. Accurate predictions are required to enable safe and efficient interactions with other road users. Pedestrians in particular pose major problems for current state of the art prediction systems. Apart from well-understood short-term predictions, used for Automatic Emergency Braking Systems, long-term predictions remain largely unresolved.

In this thesis, we aim to advance pedestrian prediction systems to enable human-understandable automated driving. In view of a vehicle-centred road infrastructure with high vehicle speeds and scarce pedestrian crossings, early detection of pedestrian intentions and movements are the key to enabling such behaviour. These detections can enable automated vehicles to perform light brake manoeuvres at an early stage in order to let a pedestrian pass. This can eliminate the need to stop, which could significantly improve traffic flow and at the same time increase overall safety.

Long-term full trajectory predictions are both costly and error-prone. Therefore we propose a hierarchical prediction system that splits the prediction into multiple simplified sub-problems using domain knowledge. Each sub-problem is designed to predict a meaningful part of pedestrian movement and to detect and remove pedestrians that are irrelevant for the current scenario as early as possible. Utilizing the given road geometry to identify crosswalks we first predict the pedestrians' hidden intent to cross the road. For all crossing pedestrians we then propose a sparse motion prediction, providing a small set of key figures instead of a full trajectory. We claim that these domain-specific key figures, namely a time-to-cross and designated crossing point, are more than sufficient to describe future pedestrian motions for the planning system of an automated vehicle. To overcome problems from over-confident single value predictions we propose to utilize Quantile Regression techniques to predict reasonable uncertainties.

Our evaluations show that we are able to robustly classify the pedestrians' hidden intent using both standard and deep learning algorithms. Additionally we show that our hierarchical prediction system, including the sparse motion prediction, is suitable for a real-time system integration. With our large real-world dataset, featuring recordings from different crosswalks and days, we provide an evaluation regarding prediction accuracy, computational load and generalizability. During this analysis, we also found indications that it might be possible to transfer trained models to previously unseen pedestrian crossings if the road geometry has at least approximately the same pavement dimensions. Furthermore, we show how our sparse motion prediction can be integrated into a situation-based planning approach to allow safe and efficient real-time interactions with other traffic participants. For

this we evaluate different interaction scenarios regarding safety, time efficiency and comfort impairment. We were able to show that in most of the scenarios it is possible to minimize movement jerks and eliminate the need to stop. The overall performance is only limited by very high traffic densities.

Zusammenfassung

Die Navigation durch dicht besiedelte Stadtgebiete ist eine der größten Herausforderungen für selbstfahrende Fahrzeuge. Um eine sichere und effiziente Interaktion mit anderen Verkehrsteilnehmern zu ermöglichen, sind genaue Vorhersagen erforderlich. Vor allem Fußgänger stellen für die, dem aktuellen Stand der Technik entsprechenden, Prädiktionssysteme ein großes Problem dar. Abgesehen von gut verstandenen Kurzzeitvorhersagen, die für automatische Notbremssysteme verwendet werden, bleiben Langzeitvorhersagen weitgehend ungelöst.

In dieser Dissertation wollen wir Fußgängerprädiktionssysteme weiterentwickeln, um menschenverständliches automatisiertes Fahren zu ermöglichen. Angesichts einer fahrzeugzentrierten Straßeninfrastruktur mit hohen Fahrzeuggeschwindigkeiten und seltenen Fußgängerübergängen ist die frühzeitige Erkennung von Absichten und Bewegungen der Fußgänger der Schlüssel, um solch ein Verhalten zu ermöglichen. Diese Erkennungen geben unserem automatisierten Fahrzeug die Möglichkeit, frühzeitig leichte Bremsmanöver durchzuführen, um einem Fußgängern passieren zu lassen. Dadurch kann die Notwendigkeit anhalten zu müssen entfallen, was den Verkehrsfluss deutlich verbessern und gleichzeitig die Gesamtsicherheit erhöhen könnte.

Langfristige Prognosen über die vollständige Trajektorie sind sowohl kostspielig als auch fehleranfällig. Daher schlagen wir ein hierarchisches Vorhersage-System vor, das unter Verwendung von Domänenwissen die Vorhersage in mehrere vereinfachte Teilprobleme unterteilt. Jedes Teilproblem wurde entwickelt, um einen sinnvollen Teil der Fußgängerbewegung vorherzusagen, und Fußgänger, die für die aktuelle Situation irrelevant sind, so früh wie möglich zu erkennen und zu entfernen. Unter Verwendung der gegebenen Straßengeometrie zur Identifizierung von Gehwegen prädizieren wir zunächst die versteckte Absicht der Fußgänger, die Straße zu überqueren. Für alle querenden Fußgänger schlagen wir dann eine sparse Prädiktion der Bewegung, die anstelle einer vollständigen Trajektorie einen kleinen Satz von Kennzahlen liefert, vor. Wir argumentieren, dass diese domänenspezifischen Kennzahlen, namentlich eine benötigte Zeit bis zum überqueren und ein ausgewiesener Kreuzungspunkt, mehr als ausreichend sind, um die zukünftige Bewegung der Fußgänger für das Planungssystem eines automatisierten Fahrzeugs zu beschreiben. Um Probleme durch zu selbstbewusste Einzelwertprognosen zu überwinden, setzen wir darüber hinaus Quantile Regression ein, um angemessene Unsicherheiten vorherzusagen.

Unsere Auswertung zeigt, dass wir in der Lage sind, die versteckte Absicht der Fußgänger sowohl mit Standard- als auch mit Deep Learning-Algorithmen robust zu klassifizieren. Zusätzlich zeigen wir, dass unser hierarchisches Prädiktionssystem, einschließlich der sparsen Bewegungsvorhersage, für eine Echtzeit-Systemintegration

geeignet ist. Mit unserem großen realen Datensatz, der Aufzeichnungen von verschiedenen Zebrastreifen und Tagen enthält, bieten wir eine Auswertung hinsichtlich Vorhersagegenauigkeit, Rechenlast und Generalisierbarkeit. Bei dieser Analyse fanden wir auch Hinweise darauf, dass es möglich sein könnte, trainierte Modelle auf bisher ungesehene Fußgängerüberwege zu übertragen, wenn die Straßengeometrie mindestens annähernd gleiche Gehwegabmessungen aufweist. Darüber hinaus zeigen wir, wie unsere sparse Bewegungsvorhersage in einen situationsorientierten Planungsansatz integriert werden kann, um eine sichere und effiziente Echtzeit-Interaktion mit anderen Verkehrsteilnehmern zu ermöglichen. Dazu bewerten wir verschiedene Interaktionsszenarien hinsichtlich Sicherheit, Zeiteffizienz und Komforteinschränkung. Wir konnten zeigen, dass es in den meisten der genannten Szenarien möglich ist, Bewegungsrucke zu minimieren und die Notwendigkeit anzuhalten zu eliminieren. Die Gesamtleistung ist nur durch sehr hohe Verkehrsdichten begrenzt.

Acknowledgements

This thesis was written during my research work in the Department for Vehicle Safety and Automated Driving in corporate research (CR/AEV) at Robert Bosch GmbH. I dedicate the following paragraphs to those colleagues and friends that accompanied and supported me during the creation of this thesis.

I would like to express my deepest gratitude to Prof. Dr. Roland Siegwart for giving me the opportunity to write this thesis, as well as the continuous support towards the completion of this thesis. I also would like to thank Prof. Dr. Margarita Chli and Dr. Rüdiger Walter Henn for supporting my thesis as co-examiners.

This thesis would not have been possible without my supervisors. I would like to thank Dr. Holger Mielenz for the close supervision, as well as, many deep and technical discussions. Theses discussions helped me a lot to both identify the best possible path for my research, and staying focused during the whole process. A big thank you to my many superiors at Bosch, Dr. Christian Danz, Prof. Dr. Frank Niewels, Arno Schaumann and Axel Stamm, for providing the best possible environment for my research and the continuous support.

My deepest gratitude to my numeruous supervisors at the Autonomous Systems Lab. Dr. Paul Furgale and Dr. Gabriel Agamenonni who guided me during the first year of my thesis and supported me with many discussions while shaping my research direction. Dr. Juan Nieto and Dr. Igor Gilitschenski, I would like to thank for the supervision during the majority of my thesis, filled with various discussions as well as unconditional support during the creation of this thesis and our main related papers.

A special thanks to my fellow doctoral students at Bosch, Dr. Jan Rohde and Dr. Jan Stellet for close and friendly cooperation as well as various distractions during our common doctoral period. Unfortunately there is no space to thank all my great co-workers, but I would like to extend a big thank you to Dr. Lutz Bürkle and Dr. Claudius Gläser for jump-starting my difficult, but highly important, data collection and pre-processing campaign with their profound knowledge and active contribution. I would also like to thank my numerous students for contributing to this thesis within the scope of their studies.

Many thanks also to Dietmar Martini and Peter Claus for keeping our test vehicles running and immediate repair of any occurring errors. I would like to thank Stefania Ambrosi, Melissa Böhm, Luciana Borsatti and Cornelia Della Casa for extensive administrative support in all situations.

I would also like to thank all my friends for providing great support and distractions whenever needed. A special thanks to Alexander Pleßmann for numerous hours of stress relieving gaming and discussions on many interesting topics.

Lastly, I would like to express my deepest gratitude to my parents and the whole family for the unconditional support and love through all these years. Without you I would not be where I am now.

June 6, 2020 *Benjamin Völz*

Financial Support

This reasearch was funded by the Robert Bosch GmbH, Corporate Research, Germany. It received partial funding from the European Community's Seventh Framework Programme (FP7) under grant-agreement n. 269916 (V-Charge). Further partial funding was recieved from the German Federal Ministry for Economic Affairs and Energy (BMWi) within the technical program "New Vehicle and System Technologies" (German: "Neue Fahrzeug- und Systemtechnologien"; Project "MEC-View").

Contents

Preface

This is a cumulative doctoral thesis and as such consists of the most relevant publications. The publications are attached at the end. In addition to the individual publications an overarching introduction is provided in Chapter 1. We start with explaining the relevance of this thesis, followed by the objectives and the approach taken to fulfill these. For each contributing publication we explain how it embeds into the overall goals of this thesis and highlight the relevance of the research work in Chapter 2. Furthermore, we show how each paper is related to our other publications. We close this thesis by a summary of the achievements and provide an outlook for future directions and research in Chapter 3.

Chapter 1

Introduction

Urban Automated Driving (UAD) is widely regarded as a potential enabler for safer, cheaper, more efficient and, if combined with electrified vehicles, climate-friendlier modes of transportation [102]. Currently, traffic can be related to a large amount of deaths worldwide. According to the World Health Organization (WHO), more than one million people die annually from traffic accidents [113]. Another three million deaths are caused by air pollution [114], where traffic contributes a potentially large amount. Predictions from [54] hint, that autonomous driving could potentially reduce road accidents by 90%, vehicle related pollution by 80% and congestion by 60%. Furthermore, it is expected that the overall amount of vehicles can be significantly reduced. This potentially reduces both the current demand for large, wide roads and parking lots in already densely populated urban areas. According to [63] up to 50% of the currently paved surface could be used for different purposes, like extended pedestrian areas, parks and living space.

Although UAD has been a major research area in the last years, there are major problems remaining. Recent forecast claim that due to the severity of these problems fully self-driving cars could still be a decade away [44]. The underlying remaining problems are manifold. They range from serious performance issues in real world traffic, to unknown safety and validation concepts [37].

In this thesis, we aim to address one of these major challenges: automated vehicles interacting with pedestrians in mixed traffic in urban areas. To enable safe and efficient interactions, the prediction of other road users and the resulting interactions is crucial. Especially pedestrians in dense (e.g. European) urban areas are of major importance.

Section 1.1 motivates the necessity of urban pedestrian prediction for automated driving and highlights the specific challenges arising from that. ntIn Section 1.2 we outline the objectives we aim to solve with this work. Afterwards the approaches utilized in this work are presented in Section 1.3. The remainder of this thesis details

our individual papers. Chapter 2 introduces their contributions and interrelations to create a full-scale real-time pedestrian prediction system. Chapter 3 summarizes the presented work and outlines possible future research topics. Finally, the appendix contains the complete list of contributing papers.

1.1 Motivation

Predicting pedestrian motions in urban areas is a major challenge and has therefore seen a significant research focus in the last years (comprehensive survey in [84]). Due to many different applications and deployment areas, the research can be divided into three different areas with varying amount of attention throughout the research community.

Considering pedestrians on urban roads the probably most researched and well understood problem is a short time prediction as required for pedestrian emergency braking systems [14, 43, 49, 52, 65, 82, 85, 87, 91]. The main challenge for such emergency systems is to identify whether a pedestrian is going to stop at a curb or enter the road directly in front of the car, yielding a very small time-to-collision (TTC). Pedestrian Automatic Emergency Braking (AEB) systems are now part of the important European New Car Assessment Programme (Euro NCAP) and therefore required for any new, 5-star rated, vehicles sold in the European Union [25].

Another important research area is related to so-called shared spaces, open areas that are shared between typically many pedestrians and at least one robot. There has been intensive research on analysing and predicting pedestrian movements through densely populated areas [4, 6, 59, 117], focussing on modelling interactions between pedestrians [2, 19, 80, 93, 97, 106, 116], as well as, group assembling and movement [10, 118]. Usually the robots in such areas are small guidance or service robots that match the low speed profile of the pedestrians (i.e. $v \leq 2$ [m/s]). The main objectives of such robots is typically to cross a shared space as fast as possible, without hitting or obstructing any pedestrians [18, 27, 35, 45, 71, 94, 99, 103, 104].

The third research area can be considered as a grouping of the many different pedestrian-related situations any vehicle encounters in urban areas. These include intersections (signalized and un-signalized), zebra crossings, traffic islands and arbitrary/ random pedestrian crossings. Although such situations account most likely for the vast majority of pedestrian encounters any vehicle has in urban areas, the overall research interest has been low. Especially when we began with this work, there was not much related research to be found. Over the last few years, the interest has been slowly, but steadily increasing (e.g. [42]). Nevertheless, the area remains widely unexplored and unsolved. Altogether proper handling of interactions with pedestrians in urban areas is of major relevance for any automated vehicle. Overall, this research area remains very large, therefore we provide a further structure.

Due to a closer analysis of these situations, we are again able to identify two specific groups, based on legal right of way definitions. Signalized intersections are

heavily regulated, essentially reducing the complexity for all vehicles. The main challenge here is posed by semi-unprotected turns, i.e. if both the vehicle and the pedestrian have a green traffic light. In such cases, the pedestrians typically have the right of way, therefore vehicles must yield. This problem definition is equal to that of a zebra crossing, here also the pedestrian has priority over the vehicle. Because of this similarity it has become more common for (at least European) cities to replace right turning traffic lights by zebracrossings. Altogether, we group all un-signalized scenarios into our second cluster. This includes namely traffic islands, random crossings and disabled/ defective traffic lights, if not part of the first cluster. In other words, the second cluster contains all scenarios where, from a legal perspective, vehicles have priority over pedestrians.

Analysing the two clusters and the challenges they implicitly pose we decided that a viable solution for all situations in the first cluster is much more pressing for early introduction of automated vehicles in urban areas, since our vehicle has to actively yield to pedestrians. The specific problems, challenges and research objectives arising from this decision are further outlined in the following Section 1.2.

Finally, we want to highlight some of the thoughts why the second cluster is less pressing and challenging as a research area: Assuming human behaviour as a baseline. Both pedestrians and human drivers are typically aware of the given priority rules. In addition, we assume that pedestrians generally move and act cautiously. Hence, the pedestrians will only enter the road, if approaching vehicles are far away. This yields two possible situations.

First, the pedestrian enters the road at a safe time-to-collision (TTC), and moves directly, i.e. usually linearly, to the opposite side of the road. In this case we only need a standard constant velocity (CV) prediction to estimate how long the pedestrian will need to cross the road [89]. Based on this estimate we can adjust our current speed to keep a safe minimal time-to-collision at any time.

Secondly, we could consider those pedestrians who enter the road close to a vehicle. Two typical reasons for such behaviours are either distraction (e.g. due to a conversation or a smartphone) or occlusions, i.e. because of visual obstructions the pedestrians has to enter the road to observe it. These situations are by definition equal to our first research area, and therefore can and should be handled by an Automatic Emergency Braking (AEB) system. Obviously, there is a grey zone in between these two situations, but we argue that this is both small and less relevant than all the combined situations of the first cluster.

Additionally, anticipating our detailed concepts and approaches that will be introduced in Section 1.3, we aim to analyse the potential of different Machine Learning (ML) algorithms to solve these prediction problems. One of the major requirement for any Machine Learning algorithm is the availability of large amounts of high quality data. Given limited data collection capabilities and since all situations of our second cluster (where the vehicle has priority over the pedestrian) are extremely rare, we decided to focus our work on the first cluster (where the pedestrian has the right of way). Altogether, data collection and labelling for the first cluster was still challenging, but feasible.

1.2 Objectives

Our research on predicting pedestrian motions in urban areas focusses on handling the majority of regular encounters where a vehicle has to yield to pedestrians in urban areas. For all further explanations and examples, we use zebra crossings as explanatory baseline. Further details on scenario selection and the associated data collection are provided in Section 1.3.

One major requirement for our automated vehicle is to provide human-like or at least human-understandable behaviours. Since our vehicle has to yield to pedestrians, one origin of this requirement can be easily visualized as follows. Because of human safety requirements, we try to protect our own life and therefore usually act safe and typically wait for conclusive clues that an approaching car will stop, before entering a crosswalk. Therefore, our automated vehicle has to include some kind of Human Machine Interface (HMI) [77]. Such interfaces can contain active communication, like external displays [20], or projections onto the road surface [21, 64]. Another possible HMI can be implicitly provided through more or less subtle clues encoded in the vehicles driving behaviour. A vehicle can for example execute an early, slight and comfortable braking manoeuvre to visualize, that it has seen the pedestrian and is actively yielding. A full/ emergency brake directly before the intersection would fulfil the same main objective (collision free stopping in front of the zebra crossing), but would be considered as unsafe, possibly frightening and very uncomfortable.

Considering a typical urban travel speed of 14 [m/s] ($\approx$ 50 [km/h]) and a comfortable deceleration of maximal 2 [m/s^2] [60], we could calculate a deceleration time (until standstill) of roughly 7 seconds. For any prediction, this obviously is a very long time, especially when considering additional reaction and processing times of up to 1 second. To reduce this time one viable solution could be to precautionary reduce the travel speed to 8 [m/s] ($\approx$ 30 [km/h]), which would reduce the deceleration time to roughly 4 seconds. Based on these simple calculations we define our targeted pedestrian prediction time to be at least 5 seconds. Obviously any larger value could potentially increase the system performance dramatically, by either allowing smoother braking, faster driving, or other manoeuvres, like early slight braking towards an intermediate cruise velocity.

In summary, the overall objective of this thesis is to develop an approach for long-term prediction in urban areas. This will enable automated vehicles to properly yield to pedestrians, where they have the right of way. The resulting minor objectives can be summarized as:

- Enabling an automated vehicle to execute motions that are understandable for pedestrians, i.e. effectively encoding an implicit HMI into the vehicles motions.
- Increasing the traffic safety and passenger comfort due to reduced deceleration amplitudes.

- Optimizing traffic flow, due to reduced decelerations and at least partially eliminated stops. (Stops in dense traffic, can lead to micro traffic jams [98].)

Additionally the system should be designed to be feasible for real-time integration into an automated test vehicle, given an appropriate real-time pedestrian detection and tracking system.

As hinted in Section 1.1, there are a few problems and situations which are by definition and design excluded from this thesis. This includes all emergency situations, which are to be handled by a parallel, independent Automatic Emergency Braking (AEB) system. In addition, we focus on the prediction of pedestrians before they enter the road. Previous research has shown that for their actual crossing pedestrians usually use a simple shortest path logic [89]. This logic can most likely be anticipated by standard short-term Constant Velocity (CV) predictions, utilizing e.g. existent Kalman Filters from the tracking system. Altogether, we focus on situations where we are able to easily collect a significant amount of data given our limited collection and labelling budget. Because of this, we have to exclude all traffic islands and random crossings. More details on data collection and associated scenarios are described in Section 1.3.

1.3 Approach

To achieve our main overarching objective (designing a real-time long-term urban pedestrian prediction system), we decided to split the problem into multiple sub-problems, to be solved by a specialized prediction component. The basic idea is, to create reduced/ simplified problems that are both easier to solve individually and remove pedestrians, which are irrelevant to the current scenario, before moving to the next component and its respective sub-problem. Summarized, we tend to minimize the computational load by calculating only relevant information for relevant pedestrians. Therefore, each component is responsible for detecting/ predicting a reduced part of the pedestrians' future motion. Section 1.3.2 introduces the individual sub-problems and the respective approach for solving them. Afterwards we present our approach for a hierarchical system that intelligently combines these components, while achieving real-time computation for arbitrarily many pedestrians (Section 1.3.3).

For increased comprehension of our specific design and architectural choices we first will introduce our database in the following Section 1.3.1. There we provide an insight on pre-defined limitations during data collection, as well as specifics of our pre-processing pipeline and its influence on our real-time capabilities.

1.3.1 Data Collection and Pre-Processing

For our data collection, that started in 2014, we used the test vehicle as depicted in Figure 1.1. At the time, the vehicle had only a limited sensor set, including:

- a Bosch series front camera with a horizontal field-of-view of 50° [79].

Figure 1.1: Automated Test Vehicle, equipped with multiple sensors. 360° surround view LiDAR mounted on the roof of the vehicle.

- a Velodyne HDL-64E, a high resolution 360° laser scanner (LiDAR) mounted on the roof of the car [105].
- an Automotive Dynamic Motion Analyzer (ADMA), a highly precise inertial measurement unit with Differential GPS [28].

Based on these sensors we identified two potential data collection scenarios. We can either drive around densely populated urban areas while recording pedestrian crossings in front of our vehicle, or record both pedestrians and other vehicles from a stationary roadside position. Altogether, we concluded that it would not be feasible to record the required significant amount of data by simply driving around a city and essentially hoping for pedestrians to cross our path at the right moment. Because of that, we identified multiple highly frequented zebra crossings with close by parking spaces from where we recorded the traffic. By this, we also limited the distracting influence of our vehicle on the pedestrians' movement while crossing the road.

Since our camera had a very limited field-of-view, we decided to only work on the LiDAR data. Due to the 360° surround view, we identified two main advantages of the LiDAR. It can record all objects approaching from different directions, and we are therefore potentially able to represent and learn from object interactions.

For our further experiments, we required intensive pre-processing to extract object tracks from raw point clouds. For this we implemented a multi-layer pipeline roughly following [101]. First, we eliminated the ground plane in our laser data

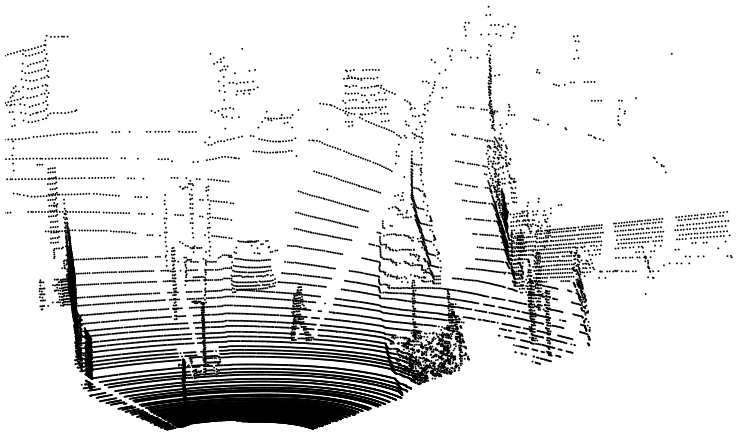
based on the intrinsic and extrinsic calibration, i.e., from the calibration we know for each individual laser at which distance it should hit the ground and consequently remove all matching points. In the second step, we created a spherical 2.5D panoramic image from the sensors raw data. The image had a size of 64 pixels in the height (equivalent to 64 lasers) and 2048 pixels in the width (equivalent to a 360° field-of-view with an angular resolution of 0.18°). Each pixel of the image can encode either the depth or intensity, measured by the corresponding laser. The clustering algorithm now simply grouped all neighbouring pixels in adjustable range with an approximately equal distance to the sensor. Finally, following a so-called track-before-detect algorithm we tracked the individual clusters over time by associating close-by clusters in the panoramic image plane. After the association in the image plane, the objects where tracked in the target 3D plane using a so called Interactive Multiple Model (IMM) filter [58]. An IMM is an intelligent combination of different Kalman Filters, where each individual filter represents a different motion model. We carefully selected and tuned a combination of Constant Velocity (CV, for the majority of the time steps) and Constant Acceleration (CA, for pedestrian-typical swift direction and acceleration changes) filters [90]. Altogether all tracks that exceeded a minimal detection length where classified using a learned classifier comparable to [101]. To illustrate the individual steps, we provide an exemplary set of panoramic images for the intermediate steps in Figure 1.2 and a

(a) Raw distance measurements visualized as grayscale image.

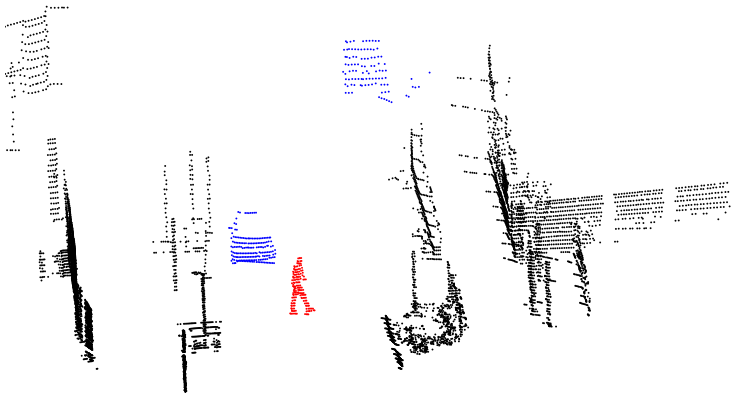
(b) Raw objects after ground plane removal and tracking. I.e. all objects that could not be associated for long-term tracking are removed as well. Each potential object is displayed with an individual colour.

(c) Classified objects visualising only tracked and classified objects: background (black), cars (blue) and pedestrians (red).

Figure 1.2: Snippet of the panorama images during various pre-processing steps. Each image has a resolution of 64 by 500 pixels.



(a) Raw (unprocessed) point cloud.



(b) Processed point cloud, containing classified objects: background (black), cars (blue) and pedestrians (red).

Figure 1.3: Point cloud before and after all pre-processing steps. The shown fraction of the full cloud corresponds to the data slice presented in Figure 1.2.

comparison between a raw 3D point cloud and a classified cloud in Figure 1.3.

Overall, we collected 2000 relevant pedestrian trajectories that at least approach one of our zebra crossings with a total of 100000 individual time steps. Since we aim to analyse the possible benefit of machine learning algorithms, we also require labels for our data. Because we calculate our tracks offline, the labelling process is trivial, i.e. implicitly given by the pedestrians' trajectory. This includes, whether the pedestrian has crossed the street, as well as the point where the pedestrian entered the road, and, for each time step, the corresponding time to cross.

Although these data allow proper offline training and detection, neither of these algorithms is currently real-time capable. Additionally the real-world applicability is partly limited due to the detection range of the LiDAR. Although the official range is given as 120 meters, we require a significant amount of detections to distinguish between objects. From our data, we deducted reliable detection ranges in cluttered urban areas as roughly 80 meters for cars and 40 meters for pedestrians. The low range for pedestrian detection and classification can be explained by two main observations. First, pedestrians are much smaller than cars, and second, urban areas contain many objects of similar shape, like trees, boxes and bushes. If we now consider a fast moving pedestrian (assuming 3 [m/s]) and a already precautionary slow car (roughly 8 [m/s], as discussed in Section 1.1) With our required, targeted prediction time of 5 seconds, we can easily calculate a minimal required detection range of 55 meters. This range obviously scales quickly and badly with our vehicles speed (roughly 136 meters detection range required for a vehicle driving with 14 [m/s] ($\approx$ 50 [km/h])), or faster pedestrians. Additional limitations arise due to frequent occlusions from e.g. parked cars or houses.

To overcome all these problems our most recent research tries to enhance the vehicles field-of-view by incorporating infrastructure-mounted sensors through Vehicle-2-Instrastructure (V2I) communication. Within the German publicly funded project MEC-View [61] we equipped an un-signalized intersection featuring massive occlusions (Figure 1.4) with infrastructure sensors and local communication devices. Figure 1.5 illustrates the extended field-of-view. Unfortunately, the sensor system is not fully functional yet, therefore we only receive car-like objects. Nevertheless we will use these data for a general proof of concept of our real-time system integration strategy (Section 1.3.3).

1.3.2 Components

We propose to approach our specific problem with three main components. The first, trivial, component filters pedestrians based on their relative location to the road. I.e. we consider only those pedestrians for further estimation that are moving towards the crosswalk and are able to reach the crosswalk before our automated vehicle. To avoid overly optimistic pruning, we consider a worst-case assumption, where each pedestrian immediately starts running. Altogether, the specific effect of this first component depends strongly on the sensoric capabilities of the automated vehicle

Figure 1.4: Drone footage of the MEC-View pilot intersection. Due to the long house at the intersection, approaching vehicles usually have to come to a stop at the yield line, before being able to observe the intersection.

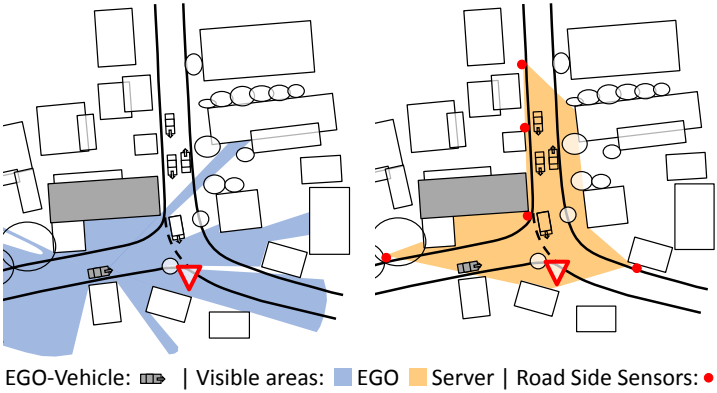


Figure 1.5: Comparison of the field-of-view of an automated vehicle with and without additional infrastructure sensors. The images depict the intersection as seen in Figure 1.4.

and its surroundings. Usually there are only very few prunable objects, if there are only vehicle mounted sensors available. If we consider additional infrastructure mounted sensors, we potentially receive a much larger amount of objects (compare Section 1.3.1).

The second component reduces the prediction problem to a simple binary classification question: does the pedestrian want to cross the road? We refer to this as the (potentially hidden) intent of the pedestrian. The main idea behind the intent is that each pedestrian has a hidden goal. Such goal could for example be to reach the subway station on the other side of the road. As long as we do not have, e.g., navigation information about the pedestrian we reduce the problem to inferring whether the pedestrian has to cross the road to reach this goal. If we manage to identify this goal reliably, we can eliminate all pedestrians that do not intend to cross the road from any further calculations and evaluations, potentially reducing the overall computational load.

As mentioned before, part of our main objective was to analyse the problem using Machine Learning algorithms. Therefore, we experiment with a subset of both classical and deep learning algorithms. Additionally we designed a set of powerful handcrafted features and ran a feature selection algorithm to identify both the most relevant features and the potential impact of vehicle movements on these hidden intents.

The basic idea of the third component is to calculate a goal-oriented estimate of the pedestrians' future trajectory. Given goal-specific ground-truth labels of our trajectories, we are able to clearly separate between crossing and non-crossing trajectories. Based on these crossing trajectories we intend to calculate an abstract estimate of the remaining pedestrians' trajectory for any observed state. The most common and well-known approach in recent research feature predicting the whole trajectory for a specified horizon. Unfortunately the majority of these algorithms produce a very high error with increased prediction horizon, as discussed in Section 1.1.

We decided to try a different approach, essentially simplifying the problem given our domain knowledge. Based on the prior that the pedestrian wants to cross the road, we claim that only a small set of important variables has to be computed. These variables are namely the pedestrians' time-to-cross, i.e. the predicted time until the pedestrian will enter the road, and the designated crossing point. The crossing point is important to identify, if the pedestrian is going to enter the road before the actual crosswalk, which effectively reduces the vehicles available braking distance Additionally the crossing point can be used to estimate, when the pedestrian will potentially leave the vehicles driving corridor. For increased safety, we extend the prediction by also learning uncertainty estimates, via Quantile Regression. The overall results are key figures calculated as regression with uncertainty. A high uncertainty can effectively be considered to increase the vehicles safety distance and time-to-collision.

1.3.3 Hierarchical Real-Time System

In order to compile the results into a full prediction that can be used by a situation based planning system, to properly react to pedestrians and other objects, we will now introduce our overarching hierarchical prediction system. The system essentially combines the three components by the following rules:

- Prune all pedestrians that are irrelevant for the current scenario.
- Classify the remaining pedestrians' intents as either crossing or not crossing the road.
- Estimate the time-to-cross and crossing point of all crossing pedestrians.
- Forward the calculated estimates to the situation based planning system.

Finally, the situation based planning system can utilize this information to choose an optimized motion for our automated vehicle. The optimization criteria can include high level-decisions, like: can we safely pass the crosswalk before the pedestrian? For all cases where we have to actively yield to a pedestrian, the planner can try to minimize the possibility of a stop, by e.g. braking early to both signal and motivate the pedestrian to cross the road. The resulting performance of the planner is thereby limited by the actual prediction performance, both late and unreliable predictions have to be met by increased deceleration amplitudes, reducing the passengers comfort and potentially creating more dangerous situations (e.g. identified by a low time-to-collision). The overall optimization criteria can be defined as: pass the crosswalk as fast as possible, with minimum ac- und deceleration amplitudes, while eliminating all stops. The last two criteria are extremely important when it comes to optimizing the traffic flow, because both sharp braking and stops, in dense traffic, can create micro traffic jams [98].

As described in Section 1.3.1 we currently do not have a reliably real-time pedestrian detection and tracking system available in the test vehicle. Therefore, we decided to show the possible integration into a situation based planning system by experimenting with a similar scenario involving only other vehicles. In this scenario, our automated vehicle has to yield to a priority road. To achieve the above-mentioned optimization criteria, it has to adjust its speed and acceleration profile to either merge into or pass through a small gap in moving traffic.

Chapter 2

Contribution

In this section all papers, as well as their contributions and interrelations, will be introduced individually. The overarching main contribution is the creation of a full-scale real-time prediction system. The system is designed hierarchically, creating multiple smaller sub-problems, which are solved by specialized components. We introduce our individual components for both high level intent recognition and the detailed motion prediction in Section 2.1. The overall integration into a hierarchical real-time system is presented in Section 2.2.

2.1 Prediction Components

Predicting pedestrian motions is a complex and error-prone task, especially when considering long-term predictions. To minimize these errors we separate the problem into smaller, well-defined sub-problems. This section introduces different components to solve each of these problems as introduced in Section 1.3.2.

Paper I

Benjamin Völz, Holger Mielenz, Gabriel Agamennoni and Roland Siegwart, "Feature Relevance Estimation for Learning Pedestrian Behavior at Crosswalks". In *IEEE 18th International Conference on Intelligent Transportation Systems (ITSC)*, 2015.

Context

One major challenge of our prediction pipeline is the identification of pedestrians' hidden intents using machine learning algorithms. Apart from the provisioning of both large quantity and high quality data, a key challenge for all Machine Learning algorithms is to find a proper feature set to create a simplistic environment representation that incorporates any a-priori known domain information. A main

challenge of this task arises from limited data collection abilities, as described in Section 1.3.1), i.e. the feature set has to be created solely from raw trajectory data.

To create a fast, real-time system, that is able to cope with many pedestrians, it is also crucial to keep the total number of features as small as possible. The majority of the machine learning algorithms scales non-linearly with the input complexity, therefore a good trade-off between performance and computational load has to be found.

Contribution

This paper focuses on defining relevant features that model the behaviour and motion of pedestrians in urban scenarios, to classify the hidden intent to cross the road at a given crosswalk. All scenarios feature a clear road infrastructure that contains a zebra crossing. This infrastructure is a-prior known, defining the position of road boundaries and crosswalks. We propose a set of 75 features that encode the raw motion state of objects (e.g. the velocity), the position of objects relative to the road (e.g. distance to the crossing, distance to the closest curb) as well as relations between pedestrians and road-bound vehicles (e.g. relative distance and speed). Additionally we encode time information by adding each feature with multiple time steps.

To estimate the benefits of each individual feature we propose to use a Recursive Feature Elimination (RFE) algorithm, to assess the relevance of each feature. As underlying classification algorithm, we utilized a classical Support Vector Machine (SVM).

Finally, we show in our experiments, that we are able to robustly classify the hidden intent given a reduced, small set of relevant features. We also show that interactions between pedestrians and vehicles have no measurable influence on the pedestrians' intent. All relevant features encoded only the pedestrians' motion relative to the road, including implicit time information through features from previous time steps.

Interrelations

Estimating a both strong and slim set of relevant features is of major importance for developing a real-time prediction system. Building on these features we experiment with further, more sophisticated classification algorithms in Paper II.

Additionally the insight, that vehicles on the road have no measurable impact on the pedestrians' hidden intent supports our claim to clearly separate the pedestrian prediction problem into a intent classification and a motion prediction step (Paper IV). The latter is of course expected to benefit greatly from inter-object relations.

Paper II

Benjamin Völz, Karsten Behrendt, Holger Mielenz, Igor Gilitschenski, Roland Siegwart and Juan Nieto, "A data-driven approach for pedestrian intention estima-

tion". In *IEEE 19th International Conference on Intelligent Transportation Systems (ITSC)*, 2016.

Context

Recent advances in Deep Learning showed a massive potential of increased performance compared to classical Machine Learning algorithms. Especially the advances in image processing promised potential improvements regarding implicitly incorporating raw pedestrian motion features. Enhancing the pedestrian intent recognition through both Deep Learning algorithms and image-based features could potentially outperform previously used classical Machine Learning algorithms.

Another line of Deep Learning research showed that so called Recurrent Neural Networks (RNN) are well suited to learn from time series data. These networks contain internal storage units that can be trained to capture even long-term data dependencies.

Contribution

This paper focuses on analysing the potential benefit of different Deep Learning algorithms. Based on our previously selected set of relevant features (Paper I), we compared the performance of our classical Support Vector Machine to a simple Neural Network. Through our experimental results, we could show that even a very simple Neural Network is able to significantly outperform a well-tuned classical approach.

Additionally, instead of explicitly encoding time-dependencies into the feature vector (i.e. by adding the same feature for multiple time steps), we utilized a Long-Short-Term-Memory (LSTM) neural network. LSTMs are recurrent neural networks designed to feature internal memory to store intelligently store time-dependent features for future processing steps. Especially long-term dependencies are usually captured very well with LSTM's. Unfortunately, through our experiments we found, that the LSTM was not able to outperform the above-introduced Neural Network trained with handcrafted features. We conclude that the feature set already adequately included the required short-time relations, while further long-time dependencies are less relevant for swiftly changing pedestrian motions.

Finally, we also tried to integrate image-based features through Convolutional Neural Networks (CNN). Since we still lack proper video-recordings, we tried to extract grayscale images from our intensity-based LiDAR panoramic images as described in Section 1.3.1. Although our networks showed minor improvements in some areas, it also increased the overall noise in the prediction. We concluded that the network was not able to learn any meaningful additional new features, probably due to the low resolution of the LiDAR images.

Interrelations

Utilizing a simple Neural Network with our small relevant feature set, we are able to significantly boost the accuracy for classifying the pedestrians' intent to

cross the road. Due to the simple network structure, we can also expect a low computational load. This allows for fast and reliable early elimination of non-crossing pedestrians, which completes an essential requirement of our targeted real-time system integration in Paper IV.

Paper III

Benjamin Völz, Holger Mielenz, Roland Siegwart and Juan Nieto, "Predicting Pedestrian Crossing using Quantile Regression Forests". In *2016 IEEE Intelligent Vehicles Symposium (IV)*, 2016.

Context

Predicting object motions is a key requirement for any state-of-the-art planning system. Considering an abstract planning system that only requires high-level state estimates and predictions to make strategic decisions, renders detailed full trajectory predictions unnecessary. Instead, a sparse prediction of relevant key figures could be sufficient. To avoid real-world integration problems arising from overly optimistic single value predictions, a proper uncertainty estimate could be of major importance.

Contribution

To avoid error-prone long-term prediction of full trajectories, we propose to simplify the motion prediction problem to a goal directed regression problem. Fitting the requirement of a situation-based planning system, we claim that it is sufficient to predict a combination of the pedestrians' time-to-cross and a designated crossing point, to adequately describe the pedestrians' further motion. To avoid problems from too strong trust into a single value we enrich our prediction with uncertainty measures, through different Quantile Regression algorithms. During our evaluation, we show that we are able to accurately predict the pedestrians' future motion using Quantile Regression Forests. Our result show that we are able to predict the future motions with relatively narrow uncertainties. We were also able to show, that the algorithm is both fast enough, to quickly react to pedestrian speed or direction changes, and robust against systematic labelling errors.

Interrelations

Predicting pedestrian motions is the last missing key component for our prediction pipeline. Within our work in Paper IV we were also able to show, that this prediction, although computationally heavy, is still fast enough for a real-time system integration with multiple potentially relevant pedestrians to predict. Furthermore in Paper V we show the real-world applicability of our proposed sparse motion prediction in a fully functional situation based planning system, running in real-time on a test vehicle.

2.2 Real-Time System Integration

A key challenge of this thesis is the development of a prediction pipeline that provides both fast and robust predictions to a real-time situation-based planning approach. This section introduces a hierarchical prediction system that intelligently combines the previously presented component predictions. Furthermore, a qualitative study on how such predictions could affect the planning system itself is presented.

Paper IV

Benjamin Völz, Holger Mielenz, Igor Gilitschenski, Roland Siegwart and Juan Nieto, "Inferring Pedestrian Motions at Urban Crosswalks". In *IEEE Transactions on Intelligent Transportation Systems*, 2019.

Context

Integrating different components estimating predictions at varying abstraction levels is one of the few remaining tasks of this thesis. Respecting real-time computation requirements, while keeping a good overall accuracy, is a key challenge for designing a fitting combined system.

Another important open question is the generalization, and resulting transferability to other, unseen scenarios. I.e. we aim to analyse whether we have to train a new model for every single crosswalk, or if we are able to deduct groups of similar scenarios that potentially could work with a common base model.

Contribution

This work introduces a hierarchical pedestrian prediction model that intelligently combines the previously introduced individual prediction components. The structure can be summarized as follows. All pedestrian detections are filtered by relevance, i.e. removing all pedestrians who, under worst-case assumptions, cannot physically reach the crosswalk before the automated vehicle. Then, all remaining pedestrians are classified regarding their intent to cross the road, effectively removing all pedestrians that, with high probability, will not cross the road. Finally, all remaining, potentially crossing pedestrians are further analysed regarding their individual time-to-cross and crossing point along the road.

To ensure real-time suitability, the combined system was evaluated regarding required execution time per pedestrian. Although the computation time of low-level motion predictions is quite high, the overall system can achieve real time performance on a standard laptop. This is mainly possible due to successful early pruning of pedestrians that are irrelevant to the current scenario. All tests ran on a single CPU core. The system is by design fully parallelizable, allowing predictions of even more pedestrians, if required.

Finally, this paper also introduced multiple additional locations and corresponding data, to allow a thorough generalization analysis for different road shapes and sizes.

A main finding of this analysis was that the size of the pedestrian sidewalk is of major importance for the model generalization. Models that have been trained on wide sidewalks, work great for other relatively wide sidewalks, but provide only limited performance in very narrow scenarios. Altogether we conclude, that we have to train individual models for different sidewalk sizes, but are otherwise able to apply models to previously unseen locations, requiring, if at all, only minor tuning or re-training.

Interrelations

Using the prediction components from Paper I, Paper II and Paper III, we designed a simplistic hierarchical prediction system. The system also allows to easily exchange the individual components to adapt the overall model to potential new advantages in e.g. future HD- or even 4K- image-base intent recognition. This obviously also holds for the computationally extremely complex detailed motion prediction. Due to the reduced number of pedestrians that require a detailed motion prediction, even a pedestrian-wise parallelization can be integrated. This would also allow the integration of even more complex motion predictions, without violating the real-time requirements.

Paper V

Benjamin Völz, Axel Stamm, Matthias Maier, Rüdiger-Walter Henn, Roland Siegwart and Juan Nieto, "Towards Infrastructure-Supported Planning for Urban Automated Driving". In *Robotics: Science and Systems (RSS). Workshop on Scene and Situation Understanding for Autonomous Driving*, 2019.

Context

Real-time pedestrian detection and tracking remains a challenging problem that limits the real-world applicability of our previously presented algorithms and systems. As hinted in Section 1.3.1 we still have no such system available in our test vehicle and are therefore not able to conduct any experiments regarding the integration into our planning system. To overcome this problem and still be able to assess the influence of our sparse motion prediction onto such a planning system, we came up with a different scenario, to analyse sparse motion predictions for vehicles in a yield situation. We claim that a zebra crossing on a regular road can be seen as approximately equal to an un-signalized intersection where our automated vehicle has to yield to a priority road. The main objective stays identical, i.e. the automated vehicle has to safely merge into moving traffic without endangering or obstructing priority traffic, while minimizing the time to complete the scenario and maximizing the passengers comfort.

Contribution

This paper provides a short overview and introduction into our situation-based planning system. To produce a sparse prediction of the vehicles motion on the priority road, we first predict the vehicles along the given lanes towards the intersection and calculate both a time and optimal crossing/ merging point from the gap between two predicted vehicles. For all further planning problems, we only consider this combination of time and position as available input. For robustness reasons we always calculate multiple gaps for further evaluation. The planner is now able to first select a safe gap, given the associated time and position is within the cars reachable physical limits. The, again hierarchically organized, planning algorithms, can now first select a feasible gap and then calculate trajectories to reach the given point in time, while varying speed and acceleration profiles. Afterwards a cost function selects an optimal trajectory for the above-defined criteria.

Our evaluation for different scenarios shows, that we are able to successfully merge into arbitrary gaps (if there is any, within our safety requirements) with only minimal jerk equalling minimal comfort costs, while eliminating stops. This result is mainly achieved due situation-specific early de- or acceleration.

Interrelations

The presented results show the general feasibility of sparse motion predictions, as introduced in Paper III and required by Paper IV, for planning safe, comfortable and time-optimal trajectories of an automated vehicle. We are confident that a similar performance could be achieved at urban crosswalks, given a real-time high-range pedestrian detection and tracking system.

2.3 List of Publications

In the context of the author's doctoral studies, the following publications were achieved. They are sorted by first author and year.

- 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.

- 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.

- J. Rohde, B. Völz, H. Mielenz, and J. M. Zöllner. Precise vehicle localization in dense urban environments. In *2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC)*, pages 853–858, 2016.

- B. Völz, K. Behrendt, H. Mielenz, I. Gilitschenski, R. Siegwart, and J. Nieto. A data-driven approach for pedestrian intention estimation. In *2016 IEEE*

19th International Conference on Intelligent Transportation Systems (ITSC), pages 2607–2612, 2016.

- B. Völz, H. Mielenz, I. Gilitschenski, R. Siegwart, and J. Nieto. Inferring pedestrian motions at urban crosswalks. *IEEE Transactions on Intelligent Transportation Systems*, 20(2):544–555, 2019.

- B. Völz, A. Stamm, M. Maier, R.-W. Henn, R. Siegwart, and J. Nieto. Towards infrastructure-supported planning for urban automated driving. In *Robotics: Science and Systems (RSS); Workshop on Scene and Situation Understanding for Autonomous Driving*, 2019. URL `https://sites.google.com/view/uad2019/`

Chapter 3

Conclusion and Outlook

In this chapter, we summarize our most important findings and highlight potential future research by analysing the limitations of our presented work.

3.1 Conclusion

We proposed and implemented a hierarchical real-time prediction system, to enable both fast and reliable pedestrian predictions in urban areas. For this, we split the underlying problem of predicting a pedestrians' trajectory into multiple smaller and easier to solve sub-problems. Each of these sub-problems can be solved by a specialised, lightweight and computationally efficient component.

Our first major component uses domain knowledge, namely road geometry including crosswalk positions and prior knowledge that pedestrians tend to use crosswalks, to create a high-level binary classification problem. This classification problem essentially estimates the pedestrians' hidden intent to cross the road. Based on a strong handcrafted feature set, containing both raw motions, as well as interactions between traffic participants with the road and each other, we showed that we are able to classify the pedestrians' intent reliably using either classical or deep learning methods.

Due to an extensive Feature Relevance Estimation, we showed that the pedestrians hidden intent can be deducted solely based on the pedestrians motion relative to the road. Interactions with other traffic participants proved to have, if at all, only a negligible impact on the pedestrians' intent. The modelling of these interactions is therefore only required for the detailed motion prediction of actually crossing pedestrians.

Based on a resulting minimalistic, relevant feature set, we managed to achieve a high classification accuracy together with low false accuracy and negatives. Our evaluation showed that we are also able to reliably predict pedestrians while they are

still far away from the crosswalk, yielding the required long-term prediction. During an intensive generalization test featuring multiple different crosswalks representing varying road geometries, we found that the width of the pedestrian sidewalk is of major importance for the pedestrians' trajectory and subsequently our intent classification performance. We conclude that we are able to transfer learned models to new, previously unseen crosswalks, as long as they feature an equally wide sidewalk. With increased absolute width, difference also the required amount of re-training with examples from the new crosswalk increases. Altogether, we claim that it is possible to train a small set of sidewalk-width-dependent base models that can be applied to new crosswalks with only minor tuning.

Given a proper intent classification, our second component focusses on estimating the pedestrians' future motion when crossing a road. To fit our long-term prediction requirements, we avoid the error-prone prediction of full trajectories. Instead, we simplify the prediction to a sparse set of key figures. We utilize a time-to-cross together with a designated crossing point to fully describe the prediction of crossing pedestrians. To avoid problems arising from overly confident predictions, we utilize Quantile Regression techniques to predict each key figure together with a proper uncertainty interval. During our evaluation, we selected Quantile Regression Forests for this task and proved that they are able to accurately predict both key figures while featuring relatively narrow uncertainty intervals even for high prediction horizons.

To achieve our main objective of creating a real-time capable urban pedestrian prediction system, we combined the above components into a hierarchical prediction system. The main idea of this system is to split the whole prediction into simplified smaller problems. By solving these problems hierarchically, we are able to remove pedestrians, which are irrelevant to the current scenario, on different levels and therefore avoid wasting costly computational power. Based on this system, we are able to execute the most costly motion prediction only for pedestrians that actually want to cross the road. Our evaluation shows that we are easily able to achieve real-time computation for up to 25 pedestrians (input to the classification level) on a single CPU core of a standard laptop.

Another major aspect of our work is the integration into a real-time situation-based planning approach. We integrated our sparse motion prediction into an existing planning framework to prove its feasibility for real-time and real-world applications. Due to a lack of both pedestrian detection range and a real-time detection and tracking algorithm, we conducted our experiments on a comparable scenario using only vehicles. Our experiment show that we are able to safely and comfortably merge into a gap in moving traffic by only considering the time and predicted position of this gap. Furthermore, as long as there was a valid gap, our planner was able to merge with minimal variation in the speed and acceleration profile, while eliminating stops completely. To guarantee the safety of the situation, we encoded the prediction uncertainty into the vehicle safety distances, which was mainly used by the tactical planning to select a safe gap.

3.2 Outlook

Overall, the presented classification component suffers from multiple limitations that would pose interesting research questions: The probably most important limitation results from insufficient semantic data, especially when considering newly detected stationary pedestrians. Based on our dataset it is only possible to estimate orientations from moving pedestrians, therefore we cannot distinguish between a person who waits for all cars to stop, before crossing and a person who e.g. stopped after the crossing to answer messages on a smartphone. Altogether, a precise, e.g. video-based, estimation of pedestrian body features [26, 74, 76], like body and head orientation or limb movements, could improve both the classification directly, and indirectly. A direct improvement could be a feature that provides the head orientation to analyse whether the pedestrian is observing the road and therefore potentially wants to cross. By including body and limb movements, the precision of the underlying pedestrian tracking system could be improved, yielding more precise trajectory information. This could significantly improve the detection of sudden movement changes, like a standing pedestrian starts moving; a walking pedestrian starts running; a pedestrian executing a sudden movement direction change (e.g. 90° orientation change in a fraction of a second). Since our system heavily relies on precise trajectory information this could significantly improve the performance for potentially high-risk corner cases.

Other helpful semantic information, especially for predicting detailed motions, could include age (child, adult or elderly) [8], clothing style (e.g. joggers), pedestrian group associations [32] and a general distraction level (e.g. chatting persons). A detection of either of these features could be utilized to customize the motion prediction and increase the overall performance and safety. This can be visualized by two examples. Elderly citizens could potentially be expected to have on average a slower walking speed and are much less likely to start running. Children on the other hand, especially unattended ones, are much more likely to move chaotically and could therefore be met with increased prediction uncertainties and safety margins.

Further performance improvements, for both the classification and regression problem, could be achieved by including additional local geographical features, or goals [78, 83], like frequently visited buildings. A subway station, for example, is much more likely to attract large amounts of pedestrians. If this knowledge could additionally be combined with real time train schedule information, the prediction could e.g. assume a higher a-priori probability for running pedestrians.

Due to their comparatively small size, reliable pedestrian prediction also requires long-range, high-resolution sensors as input for reliable detection and tracking algorithms. Recent advances in sensor technologies showed that this problem might be solved with high density LiDARs and 4K cameras. A probably even more important problem is the typical (at least European) dense urban structure. Due to a combination of roadside houses, trees, parked cars and frequent delivery trucks, occlusions pose a potentially intractable problem for all vehicle-mounted sensors. Especially when considering, that vision-based sensors (LiDAR and Video)

require to actually see their target. Such problem could potentially be solved by intensive Vehicle-2-X communication (V2X). The X can hereby be replaced by either another vehicle (V2V) [112] or some kind of infrastructure (V2I) [61]. Another vehicle, which could either be driving in front of us, or approaching from the opposite direction, could effectively enlarge our field-of-view, eliminating at least some viewpoint specific occlusions. Infrastructure sensors on the other hand, could provide a comprehensive overview on the current scene due to their elevated viewpoint. If the sensors are intelligently combined, e.g. by a local server, they could also collect almost unlimited, implicitly labelled data and therefore create an optimal prediction model for the observed scenario. Based on such a system it would also be possible to integrate a strong and reliable outlier detection and re-train/ update the model, if long-time observations show a significantly different pedestrians behaviour, as e.g. caused by construction site.

Paper

Feature Relevance Estimation for Learning Pedestrian Behavior at Crosswalks

Benjamin Völz, Holger Mielenz, Gabriel Agamennoni and Roland Siegwart

Abstract

For future automated driving functions it is necessary to be able to reason about the typical behavior, intentions and future movements of vulnerable road users in urban traffic scenarios. It is crucial to have this information as early as possible, given the typical reaction time of human drivers. Since this is a highly complex problem, it needs to be addressed in small portions. In this paper we will focus on the behavior of pedestrians at crosswalks. We use a database of real pedestrian trajectories to learn a model which is able to predict if a pedestrian will cross the street. Therefore, we first introduce a large set of possible features that could be suitable to describe the behavior. Afterwards, we perform relevance determination to identify those features that are necessary to reach the best possible generalisation performance. We provide experimental results on data collected at a pedestrian crossing in a city in southern Germany. Our results show, that a very sparse set of features, which depends only on the pedestrians' trajectory, gives the best result.

Published in:
IEEE 18th International Conference on Intelligent Transportation Systems (ITSC), 2015
DOI: 10.1109/ITSC.2015.144

1 Introduction

On the way to fully automated driving in urban environments, computers have to cope with an increasing amount of tasks, which are currently handled by humans only. One of the most complex and important tasks is the reasoning about other traffic participants. Pedestrians, especially, provide a major challenge. Even though they have a much lower velocity than other traffic participants, e.g. cars, they are usually much more agile. Pedestrians can change both their direction of movement and their velocity within fractions of a second. In an urban scenario, this could mean, that a pedestrian who was walking parallel to the street for a long time suddenly crosses the street.

The urban environment is very complex and poses a lot of situations which are difficult to anticipate. Exploiting the structure of the environment is key to accurate prediction in such situations. There are a lot of structural measures, like streets, sidewalks, intersections or crosswalks, which provide some important guidelines for the movement of all traffic participants. Furthermore, there are both legal and social rules, which define how all traffic participants have to move or at least should move in the given environment. These are, for example, right-of-way rules at intersections or crosswalks. Usually, traffic participants are acting rationally. They follow the rules and try to avoid accidents, but they also often insist on their right of way (e.g. at a crosswalk). Although pedestrians sometimes violate the traffic rules, e.g. by running over a red light, they still try to avoid accidents. This typically results in a risk-minimizing behavior such as only crossing a street if it is empty or if the closest car is still far away [86].

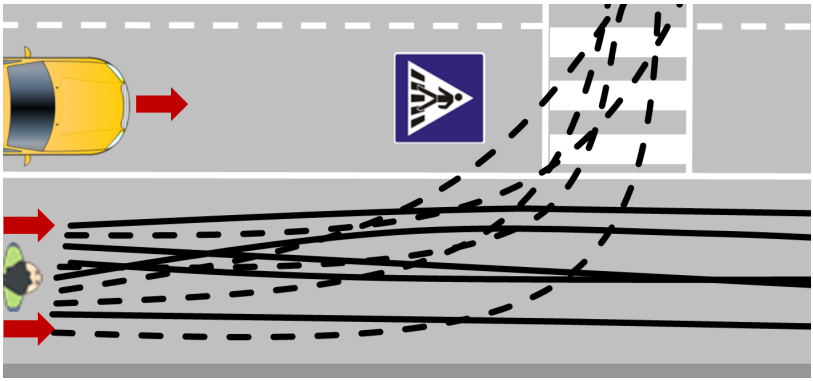


Figure 1: Typical pedestrian trajectories at a crosswalk. Both the structure of the environment (separated road and sidewalk) and the complexity of the situation, due to several present pedestrians with different intentions, is shown.

In this paper we focus on the modelling of the behavior of vulnerable road users, like pedestrians, in urban scenarios. A typical scene is shown in Figure 1. In the context of urban automated driving, it is very important to detect a crossing pedestrian as early as possible. Due to this, we focus on whether a specific pedestrian will cross the road or stay on the sidewalk. We do not predict an exact trajectory. Instead we infer the pedestrian's intention to cross the street. This intention can take one of two values: a) *does want to cross the street*; and b) *does not want to cross the street*.

This is essentially a classification problem. To solve this problem we could use a combination of a linear prediction model and a simple logic. For instance, if the pedestrian is close to a legal crossing point (e.g. a crosswalk) within the next seconds, then he will definitely cross the street, otherwise not. However, because of the previously described agility of pedestrians this prediction would often provide poor and untrustworthy results, especially for higher prediction times. We claim that it is best to let the data determine which features are important for classification. Our experimental results support this claim. In this paper we build a classifier that estimates pedestrians' intentions from data. We use a combination of real, labeled data together with knowledge about the environment, which is provided by a map. This paper addresses two important research questions: a) what information is necessary to capture pedestrians' intentions; and b) to what degree are their intentions affected by other road users.

To address these questions, we define a large set of possible features for our learning algorithm. Our features can be divided into two different basic types: Features that describe the movement of the pedestrian in a local coordinate frame and features that characterize the interaction beween pedestrians and cars. Especially the influence of present cars on the pedestrians' behavior is of great interest. Using these features we train several different models and evaluate them according to a quality measure. We use an iterative feature elimination algorithm to determine the most meaningful features. This process is also known as Relevance Determination [36] or Backward Stepwise Selection [38, Chapter 3].

In Section 2 we describe the current state of the art in the field of analyzing traffic situations and predicting pedestrian trajectories, behaviors and intentions. In Section 3 we provide a short overview on the used dataset, the corresponding sensor and the data-preprocessing. Section 4 covers our approach, including a description of the proposed features and the relvance determination methods. In Section 5 we perform an evaluation on our dataset. The conclusion and some future work is presented in Section 6.

2 State of the Art

The analysis of traffic situations is an important task for both future driver assistance and automated driving systems. A vast part of the recent research was driven by two specific use cases: fully automated highway driving [29], and collision avoidance systems [11, 40, 55, 92, 96, 120]. Since we are only looking on scenarios that

include at least one vulnerable road user, we will not take a closer look on the fully automated highway driving.

The collision avoidance systems can be divided into two different (sub) use cases. The first use case considers only cars at intersections [40, 55, 92, 96, 120]. The main goal is usually the identification of left-turning cars and their related collision scenarios. This information is used to either display a warning or execute an emergency action, like automated braking or evasive steering. [40] utilizes a database-driven trajectory matching approach. They use a labeled database of car trajectories at different intersections. For a car approaching the intersection, they now use a probabilistic matching approach to find a matching trajectory in the database and infer the future behavior, e.g. turning left, of the car. Another field of research uses probabilistic models to infer the future driving behavior. [96] utilizes a Hidden Markov Model (HMM) and a large dataset to predict the drivers intention to turn at the intersection. Another important and well used model is the Dynamic Bayesian Network (DBN), it is used by [29], [120] and [55] for both state estimation and prediction of traffic situations. A different approach that makes use of several time metrics in combination with a predefined conflict area is presented in [92].

The second collision avoidance use case covers pedestrian protection systems. [11] uses a pedestrian motion model in combination with a Bayesian Network (BN) to estimate the collision risk. [86] presents two interesting studies on the behavior of pedestrians at the curb. It analyses if a car is approaching when a pedestrian will cross the street. Secondly, it determines which information about the pedestrian and surrounding traffic are used by humans in order to decide whether to cross the street or stop at the curb.

All aforementioned papers use the special structure of an urban environment to model the behavior of two road users together. The interaction of the two road users is a crucial part of all of these systems. In the next paragraph we will present the state of the art that addresses solely the movement of a pedestrian. However, since the local infrastructure still provides some valuable information, it is used in most of the following papers.

The recognition of pedestrians action intentions and the prediction of their trajectories are solely based on image processing. [47, 48] uses the contour of the pedestrians motion to infer their intention to cross the street. State-of-the-art performance is achieved due to the implicit modeling of body language traits, like the body bending and the spread of legs. Two similar approaches are presented in [43]. One is based on dense optical flow, while the other one uses a low-dimensional histogram of the optical flow. This method, which can be seen as a variant of trajectory matching, uses the measured pedestrian's position, together with so-called motion features that capture leg and upper-body movements. A different approach, that also uses body language traits is presented in [75]. The main difference is given by the combination of a sparse representation and a larger variety of used body parts. [30] uses static cameras to find and track the heads of pedestrians and uses this for the trajectory estimation.

A common limitation of these approaches is that, although the accuracy is high,

the prediction horizon is typically in the order of hundreds of milliseconds. While this is perfectly adequate for the targeted collision warning and collision avoidance systems, we argue that it is not sufficient for urban automated driving. One of the main goals of urban automated driving is to provide a comfortable drive. In other words, large accelerations, both in longitudinal and lateral direction, have to be avoided. Initiating smooth braking manoeuvres as soon as possible requires a relatively long prediction horizon.

3 Dataset

Within the current state of the art, some type of camera is often the preferred sensor for predicting the pedestrians intention. Due to the improvements in image processing in general and stereo vision in particular, they provide good results on a low cost hardware. Unfortunately cameras usually have a very limited field of view. There is some interesting and promising research in combining several cameras and providing a full 360 degree field of view, which could be used for multi object detection and tracking [9]. However these algorithms still need a lot of work in terms of stability, accuracy and range.

For this work, it is absolutely crucial to have a very precise sensor, which is both able to cover the full 360 degree field of view and provide a decent range. Because of this we have decided to use a Velodyne laser scanner. The preprocessing of the data, namely the clustering, data association and classification of arbitrary objects, is implemented according to the approach presented in [101]. We use a Kalman Filter for the tracking of positions and velocities.

Each dataset is linked to a simple but precise map of the static environment. These maps contain information on the road geometry and the accurate position of curbs and crosswalks.

This paper focuses on analysing and modeling the pedestrians' behavior at crosswalks. The data, which are used in this paper, where recorded at the crosswalk shown in Figure 2. The figure also displays trajectories both for a crossing, and a non-crossing pedestrian. Figure 3 shows a preprocessed point cloud.

The dataset contains several hours of pedestrian and car trajectories at this crosswalk. The trajectories were recorded on three different days, both at midday, and in the evening.

4 Procedure for the Feature Relevance Estimation

In this paper we use real data, to learn the behavior of pedestrians at crosswalks. For this, the labeled dataset is used together with a nonlinear[1] Support Vector Machine (SVM) [15]. A SVM is ideal for this task, because the decision, if a pedestrian is crossing or not, is binary. The main goal of the paper is the identification of the

[1] A nonlinear SVM is used to achieve the best possible prediction performance. A simple test with a linear SVM produced inferior results.

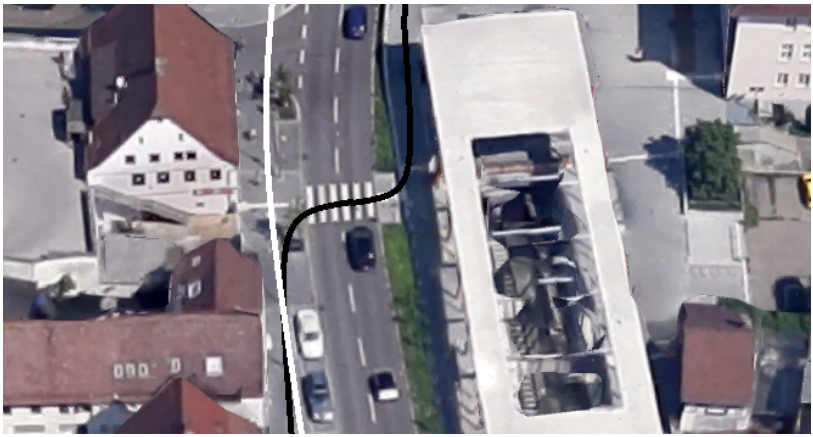


Figure 2: Google Maps image of a crosswalk with example trajectories of a crossing (black) and a not crossing (white) pedestrian. The large building on the right side is a frequently used subway station. ©2015 Google Map data (©2009 Geo-Basis-DE/BKG), accessed March 26, 2015.

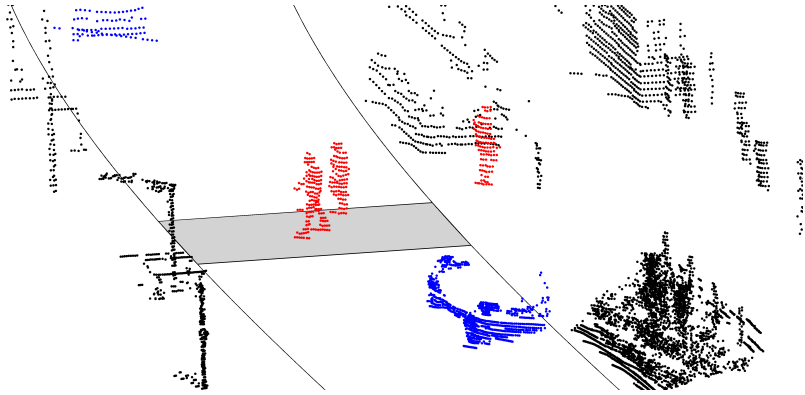


Figure 3: Example of a Velodyne Point Cloud with an underlying sketch of the street. The two black lines mark the curbs and the grey box symbolizes the position of the crosswalk. The image contains the following Objects: cars (blue), pedestrians (red) and background (black).

most meaningful features, which are necessary to make a solid guess about the pedestrians' future behavior. Our approach contains the following steps:

(1) Define a set of possible features.

(2) Train the SVM with a reasonably chosen training set.

(3) Evaluate the resulting model with a separate test dataset.

(4) Calculate the least meaningful feature and remove it from the feature-space. (Compare Section 4.4.)

(5) Repeat from step 2, until appropriate termination criteria are met. (Compare Section 4.4.)

(6) The evaluation results of step 3, of all iterations, are used for a final evaluation. This will be presented in Section 5.

The two datasets, used for the training and the evaluation, are completely distinct. Namely, the training set contains only the trajectories recorded on one day (but at different daytimes), while the test dataset consists of those trajectories recorded on other days. Therefore the evaluation essentially represents a generalization test.

There are several possible termination criteria. Considering the two distinct datasets, we could terminate the iteration if a certain classification accuracy has been achieved. Given only the training set, we could utilize the cross validation (CV) error. The termination criteria would be met if this error is below a predefined boundary.

4.1 Pedestrian Features

There are several possible physical values, which could describe the movement of pedestrians. The velocity of the pedestrian v_{ped} is of high interest. For better separability both the 2d coordinates $v_{ped,x}$ and $v_{ped,y}$, and the absolute value $|v_{ped}|$ are used.

Using the provided map together with the measured position of the pedestrian, the following relative distance measures can be calculated:

- Distance to the Curb *dtcurb*: minimal orthogonal distance to the closest curb.

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

- Distance to the Crosswalk *dtcross*: minimal distance to the crosswalk.

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

Additionally, the distance traveled between the last and the current time step s_{ped} is used.

$$s_{ped}(t) = \begin{vmatrix} x_{ped}(t) - x_{ped}(t-1) \\ y_{ped}(t) - y_{ped}(t-1) \end{vmatrix} \tag{1}$$

4.2 Relation between Pedestrians and Cars

The presence of a car might have an impact on the pedestrians' movement. Because of this, we add several features for the car that would have the biggest impact in the pedestrians' movement. The following procedure is used: First, according to their distance to the pedestrian, all relevant cars are identified. Second, all cars, which have already passed the crosswalk are eliminated. Third, a simple cost function is used to choose the car, that is both close to the pedestrian and will reach the crosswalk at an equal point in time.

For the chosen car the following measurements are calculated. Similarly to the proposed features for the pedestrians, the velocity of the car, both in 2d coordinates ($v_{car,x}$, $v_{car,y}$) and as absolute value $|v_{car}|$, is used. Same applies to the previously traveled distance of the car s_{car}, which can be expressed with an equation equal to (1).

Utilizing the map information, the distance to the crosswalk $dtcross_{car}$ is calculated as one relative distance measure.

$$dtcross_{car} \begin{cases} = 0 & \text{if the car is on the crosswalk} \\ \geq 0 & \text{otherwise} \end{cases}$$

Considering the relation between the pedestrian and the car, both the distance between them

$$d_{ped,car} = \begin{vmatrix} x_{ped} - x_{car} \\ y_{ped} - y_{car} \end{vmatrix},$$

and their relative velocity

$$v_{rel} = v_{car} - v_{ped}$$

can be included into the feature space. As with all previous velocity values, $v_{rel,x}$, $v_{rel,y}$ and $|v_{rel}|$ are used.

4.3 Track-History for an extended Feature-Space

For each time step, the history of all features from the last four time steps will be included into our feature-space. For example: instead of just $s_{ped}(t)$ the values: $s_{ped}(t)$, $s_{ped}(t-1)$, $s_{ped}(t-2)$, $s_{ped}(t-3)$ and $s_{ped}(t-4)$ would be used. In total there are five values for every feature and a total of 75 features.

4.4 Recursive Feature Elimination (RFE)

Besides the definition of possible features, the main goal of this paper is the identification of the most relevant features. In this Section we will give a short overview on an algorithm called Recursive Feature Elimination (RFE) [36]. The algorithm contains the following steps, which are processed iteratively, until appropriate termination criteria are met:

(1) Train the SVM.

(2) Compute a ranking criterion for all features. The ranking criterion is based on the raw SVM weights.

(3) Remove the feature with the smallest ranking criterion.

In order to analyse the importance of every feature, we ran the selection process until all features were eliminated (no specific termination criteria). The output of this algorithm is a ranking over all features. Additionally the trained SVM's from step 1 are used for separate tests with a second datasets. The results are evaluated in the next Section.

One important aspect of the RFE is the so called group elimination. For this mode, all features can be named as part of arbitrary groups. Therefore the steps 2 and 3 of the algorithm are altered. The ranking criterion is adjusted such that it provides a suitable ranking over all groups. After that, all features in the group with the smallest ranking are eliminated together.

5 Evaluation

For the evaluation we used the previously described RFE to train and test multiple SVM's. Within this Section we first present the results of the group- and single-elimination. Afterwards we take a closer look at the classification results for one particular feature set.

We use the following notation:

During the classification, a decision is made on whether the pedestrian will cross or not cross. In this Section we will refer to this, as a *positve* (for a crossing pedestrian) or *negative* (for a non-crossing pedestrian) result. As a consequence of this, we can define the *true positive rate* as percentage of all crossing pedestrians who are labeled correctly. Accordingly, the *true negative rate* can be defined as percentage of all pedestrians correctly labeled as non-crossing. In the same manner, we can also define the *false positive* and the *false negative rate*. The percentage of labels that have been recognised correctly is known as *accuracy*.

5.1 Group Elimination

Two different types of groupes are used. First, the data are separate according to their real physical measure. This generates 15 groups with five members each. Each group belongs to one of the original single features and contains their related time steps. For example: one group contains the values $s_{ped}(t), \ldots, s_{ped}(t-4)$ (compare Section 4.3). The resulting ranking is showed in Table 1 together with the corresponding classification accuracy.

The second group type splits the data into time steps. We get five groups with 15 members each. Table 2 shows the result of this group elimination.

The *Overall Classification Accuracy* value in both tables (1 and 2) shows the performance achieved by the model containing the groups in the current row, and all the rows above it.

Table 1: Result of the first group elimination with the corresponding overall classification accuracy and the false positive rate. The first group ranking covers real physical measures (15 feature groups with 5 members each).

Number of Groups	Physical Measure	Classification Accuracy	True Positive
1	$v_{ped,x}$	0.6518	0.0514
2	$dtcross$	0.8777	0.8401
3	$\boldsymbol{dtcurb}$	**0.9167**	**0.9120**
4	$\lvert v_{ped}\rvert$	0.8981	0.8831
5	$v_{ped,y}$	0.8733	0.8472
6	s_{ped}	0.8502	0.8289
7	$v_{rel,x}$	0.8420	0.8437
8	$\lvert v_{rel}\rvert$	0.8276	0.8514
9	$d_{ped,car}$	0.8239	0.8507
10	$v_{car,x}$	0.8048	0.8510
11	$\lvert v_{car}\rvert$	0.7917	0.8500
12	$v_{rel,y}$	0.7855	0.8493
13	$v_{car,y}$	0.7768	0.8437
14	$dtcross_{car}$	0.7778	0.8429
15	s_{car}	0.7679	0.8418

Table 2: Result of the time step based group elimination with the corresponding overall classification accuracy and the false positive rate. Every group contains all features at one timestep (5 feature groups with 15 members each).

Number of Groups	Time Steps	Classification Accuracy	True Positive
1	t	0.6845	0.6768
2	$\boldsymbol{t-4}$	**0.8331**	**0.8232**
3	$t-1$	0.8078	0.8451
4	$t-3$	0.7867	0.8563
5	$t-2$	0.7679	0.8418

There are two important points, which can be concluded from Table 1. The best-ranked feature groups for physical separation are: $v_{ped,x}$, *dtcurb* and *dtcross*. Together they reach a overall classification accuracy of 91.67%, which is the best result in the whole table. The second point is that all features that are related to a car are at the end of the ranking and provide inferior classification results if added to the feature space. For this dataset there is no positive, measurable impact of a present car on the pedestrians' intention to cross the street. However, there is most likely an impact on the exact pedestrian trajectory. Therefore, this information will be necessary if we want to answer the questions: Where and When will the pedestrian enter the street? This will be part of our future work.

Now we take a closer look at the second group elimination (Table 2). The best classification result is generated for the combination of the time steps t and $t-4$. Together they achieve an overall classification accuracy of 83.31%. This suggests that it might not be necessary to provide the history of the tracks in the given level of detail. If the results of the two different group eliminations are compared, it can be seen that the second group elimination produces inferior results. This confirms again that some of the 15 original features do not provide additional usefull knowledge. But it can be assumed that the result of the first group elimination can be improved by removing some of the included timesteps.

5.2 Single Elimination

In this Section we use the standard RFE to eliminate all features independently.

- Accuracy
- True Positive Rate
- True Negative Rate

All classification rates are evaluated on a time step basis. Accordingly, the pedestrian trajectories are not evaluated as a whole and no track-labels are inferred. The results are plotted in Figure 4. Since we have to choose a particular feature set, all classification rates have to be evaluated together. To get the best overall performance, a feature set has to be found that maximizes all curves. Especially a high *true positive rate* is of particular importance. The reason for this is that a *crossing* pedestrian who is wrongly labeled as *not crossing* presents a high security risk, if he is already close to the curb. This wrong classification could possibly lead to an accident and has to be considered as the *worst-case* scenario. By taking a look at all three curves together, the optimum can be found for exactly 10 used features. The very important *true positive rate* reaches a global maximum at this point. Even though there are feature combinations that provide a similar or slightly better *accuracy* and *true negative rate*, the corresponding decline of the *true positive rate* forbids the usage of these feature sets.

For a higher number of used features, one can see that, although the large feature sets still provide good results, the *accuracy* is decreasing steadily. This decline belongs again to the car-related features. Figure 4 shows a small dip in the *accuracy*

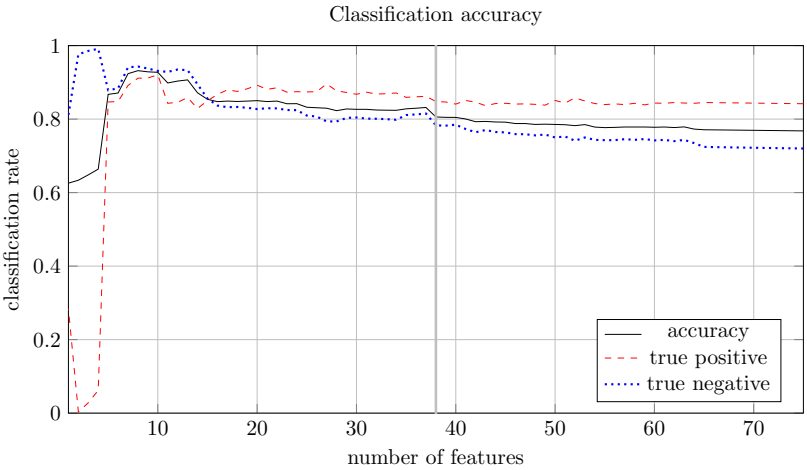


Figure 4: Result of the single feature elimination. The classification accuracies are plotted over the number of used features. The Evaluation is carried out on a time step basis, therefore the results show the percentage of correct classified time steps of all trajectories. This is not an evaluation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

somewhere between 35 and 40 used features. This is roughly the point, where the first car-related features are added to the feature space. In contrast, the 10 features that provide the best result are again part of the groups: $v_{ped,x}$, *dtcurb* and *dtcross*.

5.3 Evaluation of the Best Feature Set

In the last Section we have found the feature set that provides the best classification rates. We now take a more detailed look on the classification result of this particular set. So far we have carried out the evaluation only on the basis of time steps, i.e. by considering the data in each trajectory as independent observations. We will now switch to an evalutation that is based on trajectories. Therefore we introduce a new measure for determining trajectory labels, based on the labels of their associated time steps. Specifically, we label a trajectory as correct or incorrect according to the fraction of time steps that are correctly classified:

$$\text{label} = \begin{cases} \text{correct} & \text{if } \frac{\text{number of correct time steps}}{\text{number of all time steps}} > y \\ \text{not correct} & \text{otherwise} \end{cases}$$

with $y = (0, 1)$.

Table 3: Evaluation of the Best Feature Set: Total number of *postive* and *negative* trajectories for the discretized distance measures *dtcross* and *dtcurb*. For all cases, the number of correctly labeled (true) trajectories is displayed. A trajectory is marked as true, if the percentage of true time steps is greater than y.

	$x = dtcross$					
	positive (cross)			*negative* (not cross)		
Distance	all	*true*		all	*true*	
x in m		$y = 0.5$	$y = 0.8$		$y = 0.5$	$y = 0.8$
≤ 1	36	36	36	52	52	52
$1 - 2$	26	26	25	39	39	38
$2 - 3$	20	19	16	28	28	25
$3 - 4$	28	23	18	38	31	26
$4 - 5$	15	12	8	41	37	33
> 5	18	12	11	30	27	26
	$x = dtcurb$					
≤ 1	38	38	38	55	55	55
$1 - 2$	28	27	26	39	39	38
$2 - 3$	26	22	21	33	29	28
$3 - 4$	28	22	17	50	45	40
$4 - 5$	10	7	7	17	15	15

Figure 5 shows boxplots of the achieved *true positive* and *true negative rates* as a function of the distance measures *dtcross* and *dtcurb*, respectively, the distance to the nearest pedestrian crossing, and the distance to the curb. The distance measures have been discretizised to obtain a better readability. Table 3 provides a short overview on the number of trajectories that belong to the different discrete distance intervals. Additionally, the number of *true positve* and *true negative* trajectories, for $y = 0.5$ and $y = 0.8$, are presented. Together the figure and the table provide an overview of the overall classification performance, particularly with respect to the possible prediction horizon.

We will now analyse the presented classification rates. Figure 6 shows the general layout of the street, crosswalk, and sidewalk together with some generic pedestrian trajectories. Image 6a shows two typical pedestrian trajectories. For one, there is a pedestrian who is passing the crosswalk with a constant *dtcurb*. The second trajectory shows a crossing pedestrian, who is walking parallel to the crosswalk for a long time, before turning towards the street. Accordingly, these two trajectories are, at their beginning, very hard to separate. Therefore, they represent the main cause for the *inferior* classification rates, especially for the *true positive rate*, in the interval $3m < dtcurb \leq 5m$.

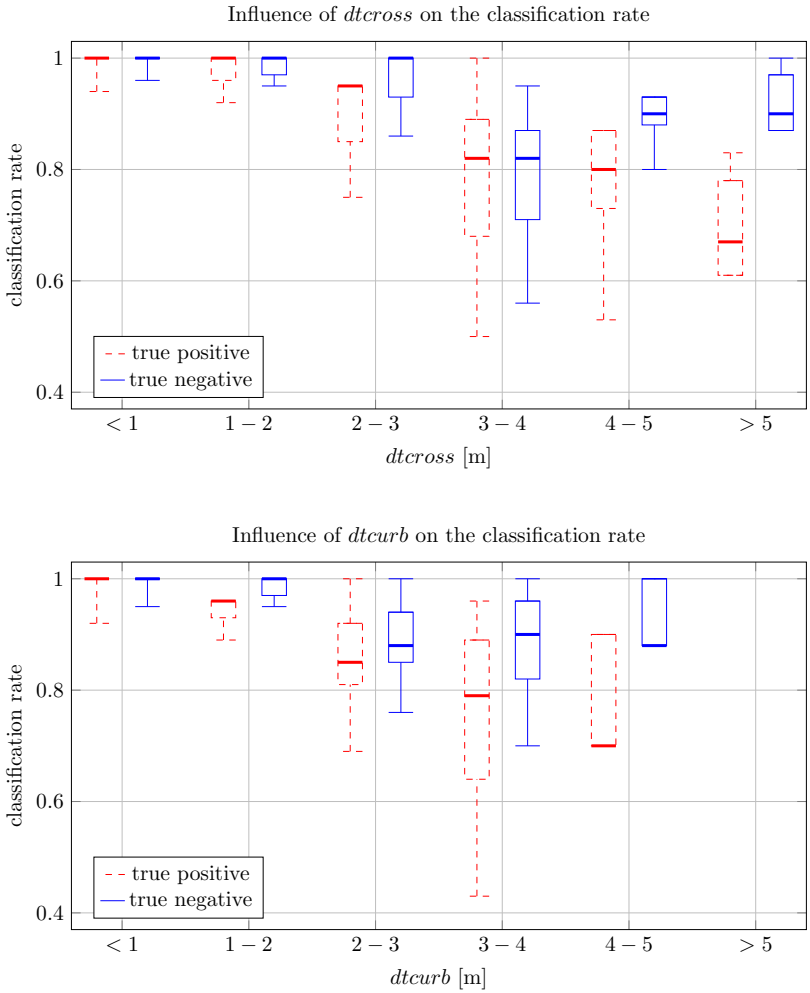
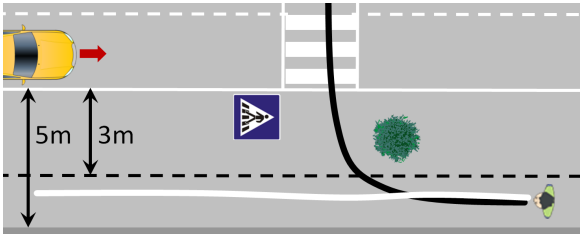
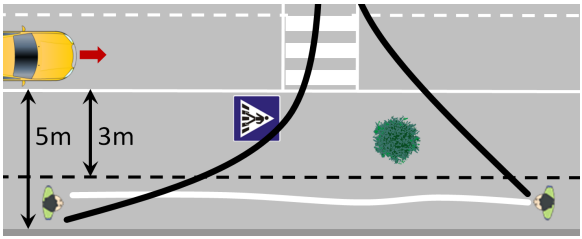


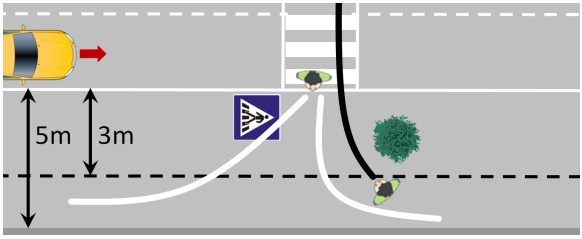
Figure 5: Evaluation of the *true positive* and *true negative rate* for the best feature set from Section 5.2. The results are plotted over the discretized distance measures *dtcross* and *dtcurb* and provide an impression of the possible prediction horizon. Together with the (known) pedestrian velocity pedestrians the results can be used to estimate the prediction horizon (e.g. with $v = 1.5m/s$, the prediction time for $dtcross > 1.5m$ would be $> 1s$).



(a) White: Typical trajectory of a pedestrian, who passes the crosswalk with a constant *dtcurb* of $3 - 5m$. 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.



(b) The black trajectories are easily identified as *Crossing* trajectories, because they show a constant decrese in both the *dtcurb* and *dtcross* value over time. This results in many correct classifications even for large distance values.



(c) Trajectories for small values of *dtcross* (and *dtcrub*). Here the white, *Not Crossing*, trajectories belong to pedestrians who already have crossed the street. They can be easily separated from the black, *Crossing*, trajectories, since the values of *dtcross* and *dtcurb* are either constantly increasing (white) or decreasing (black) over time.

Figure 6: Selection of typical pedestrian trajectories, either crossing (black) or non-crossing (white), at the crosswalk. For clarity only one side of the road is shown.

In contrast, Image 6b shows those crossing trajectories that can be easily separated from the passing pedestrian, because they can be identified e.g. due to a steadily decreasing value of *dtcurb*. The majority of the crossing trajectories shows this behavior. This results in the strong foundation of the *true postive rate*, even for large distances.

For the interval of $dtcross < 3m$ Figure 5 shows that the median of the *true positve rate* is always above 95%, while the median of the *true negative rate* is close to 100%. Image 6c shows the typical trajectories of pedestrians that are close to the crosswalk. Please note: In this context, a non-crossing pedestrian is often a pedestrian who has crossed the street in the past and is now moving away from the crosswalk. These trajectories are rather easy to separate, since the values of *dtcross* and *dtcurb* are usually identical and either *decreasing* (for a crossing pedestrian) or *increasing* (for a not crossing pedestrian) over time. Altogether, it is almost impossible to confuse these distinct movements. Accordingly the classification accuracy is always very high for small values of *dtcross*.

6 Conclusion and Future Work

In this paper we proposed a large feature set, which has been used for the training of a binary classifier in order to predict pedestrian's intentions. We defined a comprehensive set of features, and used relevance determination to determine a subset of strongly predictive features that produce high generalisation performance. Afterwards we used a relevance determination algorithm to identify the most meaningful features within our feature-space. Our results show that a small subset of these features, which only depend on the pedestrians' trajectories and a local map, gives the best results. All features that depend on the presence of other road users are of less importance. They seem to provide no additional information for this specific task. For a automated car the presented results are the first step towards a complex system that is able to predict the pedestrians' movement in arbitrary situations.

For our future work we are planning on analysing the problem further. Specifically, we are planning on collecting data in other locations (i.e. intersections). We also want to analyze whether a model generated for a particular location is apllicable to other locations of the same type (e.g. crosswalks). This is straightforward, since our features are not specific to the local structure of the road network. Another interesting question is, if it is possible to train some kind of *global* behavior model, which can be used for arbitray locations.

Another promising direction for future research, is the learning of continuous information from data, e.g. predicting where and when a pedestrian will enter the street. This knowledge is potentially valuable for future automated vehicles and Advanced Driver Assistance Systems (ADAS).

Paper II

A data-driven approach for pedestrian intention estimation

Benjamin Völz, Karsten Behrendt, Holger Mielenz, Igor Gilitschenski, Roland Siegwart and Juan Nieto

Abstract

In the context of future urban automated driving many important problems remain unsolved. A critical one is the analysis and prediction of pedestrian movements around urban roads. Especially the analysis of non-critical situations has not received much attention in the past. This paper focuses on analyzing and predicting movements of pedestrians approaching crosswalks, a very crucial pedestrian-vehicle interaction in urban scenarios. In our previous work, we analyzed the performance of a data-driven Support Vector Machine-based architecture, and the relevance of specific features to infer pedestrian crossing intentions. In this paper, we will use our previous results as baseline to compare against an architecture based on neural networks for time-series classification. In particular we analyze the effectiveness of dense and Long-Short-Term-Memory networks. Furthermore, we will be looking into enhancing our feature vectors by adding LiDAR based images to the classification process. Additionally the evaluation provides an estimate for the temporal prediction horizon. The approaches presented are validated with real world trajectories recorded in Germany. Our results show an average accuracy improvement of $10 - 20\%$ with respect to our previous Support Vector Machine-based approach.

Published in:
IEEE 19th International Conference on Intelligent Transportation Systems (ITSC), 2016
DOI: 10.1109/ITSC.2016.7795975

1 Introduction

Predicting the movement of arbitrary objects is a crucial part of automated driving systems. When considering urban automated driving especially the long-term prediction of pedestrian trajectories represents a major challenge. To illustrate this, consider the example shown in Figure 1 where a car and a pedestrian approach a crosswalk. The car is obliged to stop if the pedestrian intents to cross the street. Timely inference of pedestrian intentions is extremely difficult, and designing a system for this requires important considerations from the vehicle's perspective. First, we do not want to execute an emergency braking maneuver or apply any sudden speed change. These actions would both be uncomfortable for the occupants of the car and highly dangerous for the pedestrian and other vehicles in the area. Second, we do not want to stop when unnecessary, i.e. if the pedestrian does not intent to cross the road. Late and false predictions in such situations will lead to a low system acceptance, apart from deteriorating traffic conditions. Additionally we also have to consider the pedestrians movement possibilities. Although the speed of pedestrians is in general much lower than the one of vehicles, pedestrians are much more agile. A pedestrian can change directions very quickly, for example by doing a sharp (e.g. 90°) turn without reducing the speed. This high agility is what limits current systems to achieve reliable pedestrian movement predictions for only

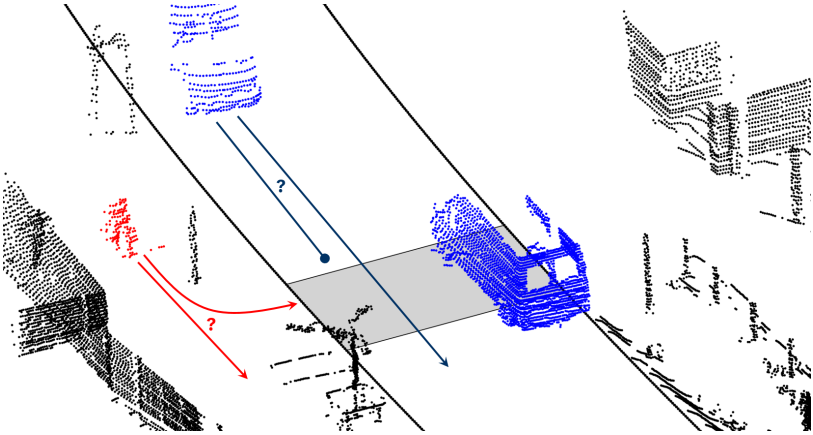


Figure 1: Typical urban scenario: a car (blue) and a pedestrian (red) are approaching a crosswalk (grey box), where the pedestrian has priority. The inference problem involves realizing whether the pedestrian intents to cross the road. For the (e.g. automated) car this information is vital to decide whether it has to stop before the crosswalk or not.

a few hundred milliseconds (e.g. [11]).

Motivated by these fundamental problems, our work aims to develop a system that (i) minimizes false detections and (ii) provides long-term predictions to ensure smooth and safe maneuvers.

One of the main findings of our previous work [107] was the difficulty to build hand-crafted features that generalize well. Deep learning architectures are able to provide end-to-end learning, obtaining therefore the features from the data. This property motivated the idea of utilizing a deep-learning architecture for the inference of pedestrian intentions. In this paper we will focus on the prediction of the pedestrians' intention to cross the street at a given crosswalk. In our previous work, we utilized a Support Vector Machine (SVM) based pipeline with very good results for the given problem. We use this pipeline as a baseline for our comparison of different neural network architectures. In this paper, we will first introduce a dense neural network for a fixed number of time-steps and features to directly classify a pedestrian's intent. For this we will use exactly the same input for both the neural network and the SVM.

In addition to the dense network, a Long-Short-Term-Memory (LSTM) network is created to allow time-series inputs of different sizes. Since LSTMs have been created for learning in time series [39] we expect a higher accuracy. A few optimal features could be created by capturing video feeds of the pedestrians, such that future poses and orientation could be inferred from images. Unfortunately, our current dataset does not contain that information. Therefore we created 2d images from LiDAR data. The Velodyne LiDAR provides a range and an intensity value for every sampled point. These images allow us to gather information from pose and change in pose over time and possibly let us infer information for our problem. For each point, the id of the recording laser, as well as the rotation angle of the LiDAR itself, are known. With these known angular coordinates, it is possible to create 2D images for each spin, for example by coloring by intensity or range [100]. This way, the remarkable image processing classification capabilities of convolutional neural networks may be leveraged. A network is created to classify predictions solely based on images and another one in combination with our hand-crafted features.

The evaluation is performed on pedestrian trajectories recorded in Stuttgart, Germany, and features an evaluation of the temporal prediction horizon.

The specific contributions of this paper are:

- the formulation of a dense and a LSTM network for predicting pedestrian intention near crosswalks,
- a comparison between the different networks and baseline SVM,
- a performance analysis based on LiDAR-based 2D images,
- evaluation of the temporal prediction horizon.

The remainder of the paper is structured as follows: The state-of-the-art on predicting trajectories, behaviors and intentions of pedestrians in urban traffic is reviewed in Section 2. Section 3 introduces different types of neural networks

for classifying time-series of feature vectors. This includes the introduction of a convolutional network for image processing. The evaluation in Section 4 first introduces the dataset, the hand-crafted features and gives an overview on our LiDAR-based 2D images. After that, the different types of networks are evaluated and compared to the SVM baseline. The conclusion is presented in Section 5.

2 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 and are therefore mostly designed to predict whether a pedestrian is going to stop at the curb or not (e.g. [11, 43, 50]).

Most of the vision-based algorithms combine both the detection and prediction of pedestrians. For the scope of this paper we will only analyze the different path prediction techniques and the features employed. An interesting study, that identifies which information human drivers use to decide whether a pedestrian will stop at the curb or not, is presented in [86]. 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 [48] 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 [43]. They show methods based both on the dense optical flow, and a low-dimensional flow-based histogram. They calculate so called motion features, which again capture both the leg 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, e.g. including arm movements, together with a sparse geometrical representation, where every body-part is depicted with a single line, is used in [75]. A common limitation of all discussed algorithms is the prediction horizon. For the given scenario (usually collision avoidance), the prediction accuracy is generally very high for a time horizon of only several hundred milliseconds. This value, however, is not enough for our application. Additionally the shown scenarios only 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 [50]. Here the head orientation is used to determine the pedestrians situational awareness, i.e. if the pedestrian is aware of the approaching car. The paper incorporates this measure into a Dynamic Bayesian Network (DBN) and shows the benefit which a head tracking could add to existing prediction algorithms. They are able to

outperform more complex state-of-the-art algorithms but still have a very limited time horizon.

Apart from these vision-based systems there are other interesting approaches that utilize the pedestrians' trajectory in terms of e.g. Cartesian coordinates in a specified coordinate frame. Again in the context of collision avoidance systems, [11] 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 [24]. These patterns represent the most common paths in this specific environment. A long-term prediction approach is presented in [41]. 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.

This paper aims to provide an approach able to provide predictions with longer time horizon which enables safer interaction between pedestrians and vehicles and is a basic requirement for fully automated driving systems.

3 Neural Network Architectures

This section presents the different architectures that we will evaluate. The demonstrated power of generalization of deep neural network architectures combined with its flexibility in building features are our main motivation to opt for this type of paradigm. In Section 3.1 we introduce a simple dense (feed-forward, fully-connected) network, which we use to create a neural network baseline. It also is the easiest network for initial tests since we can directly use the existing data without any changes. A more sophisticated network structure is presented in Section 3.2. Recurrent networks are designed for learning in time series. Since our database consists of trajectories this matches our scenario perfectly. Furthermore we use convolutional networks (Section 3.3) to learn features from our image source (compare Section 1), these features can be used as either sol or additional input for any of the other networks. All networks are trained for the same classification task (intention recognition) with slightly different properties and inputs. Hyper-parameters were selected by searching within a hand-crafted set of options and then fine-tuning those.

3.1 Dense Neural Networks

Dense neural networks represent the straightforward approach of dealing with the classification of feature vectors. A dense neural network can be divided into several layers. In the case of feed-forward networks, each layer has a predefined number of neurons which are only connected to neurons in the next layer. All dense networks employed in this paper are similar to the depiction in Figure 2.

The input data, our feature vector, leads into a fully connected layer. Rectified linear functions [67] are used as activation function to attain some non-linearity

and training stability. The activation layer is followed by a dropout layer [95] for regularization. This combination of fully-connected, activation and dropout layer is repeated a few times. The final fully-connected layer only has a single output neuron for classification which a sigmoid function transforms to values between -1.0 for not crossing the street with a very high probability and 1.0 for crossing.

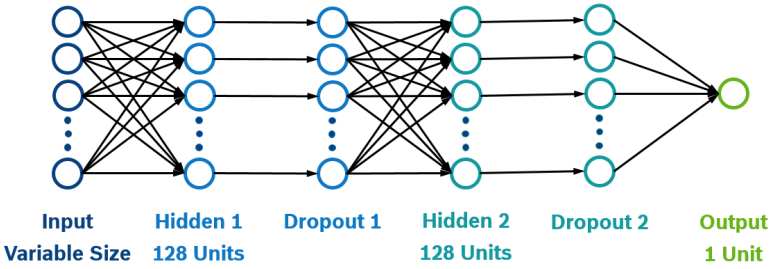


Figure 2: Sample dense neural network with 2 fully connected layers, 2 dropout layers and a decision layer with sigmoid activation.

3.2 Recurrent Neural Networks

Time-series data can often be analyzed more accurately using recurrent neural networks which allow feeding data back into previous layers. One widely employed variant contains Long-Short-Term-Memory (LSTM) units [39]. These networks store state information in their cells which is changed based on new inputs and previous outputs. The output is calculated based on cell state and input values.

LSTM networks are a combination of their cell state and four gate layers. Each gate represents a fully-connected layer with a fitting activation function that takes a concatenation of the current time-steps data and previous output as input. Those gates are then combined by element-wise multiplication and addition to a complete LSTM layer. The forget gate can decrease values in the cell, while the input and cell gates leads to an increase in values. The output is calculated by the output gate which decides which values are being used for classification in this case.

3.3 Convolutional Neural Network

The intent classification may also be possible using LiDAR-based images which can be analyzed using convolutional neural networks [53]. Image features are extracted by convolving trained filters along the image and using those features to classify the respective images. A first approach is done by only using image features and as a second step, the input vectors of our previous networks are added to the input. This feature combination happens at a later stage of the network by simply concatenating image features with the pre-calculated vectors. For regularization purposes, dropout

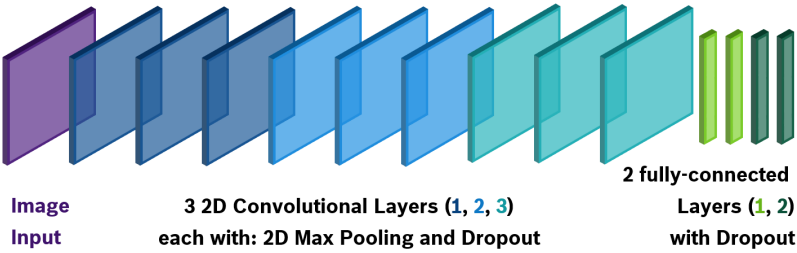


Figure 3: A sample convolutional neural network. The Figure shows three convolutional layers, each followed by a max pooling and a dropout layer. The last convolutional layer is connected to two fully-connected layers.

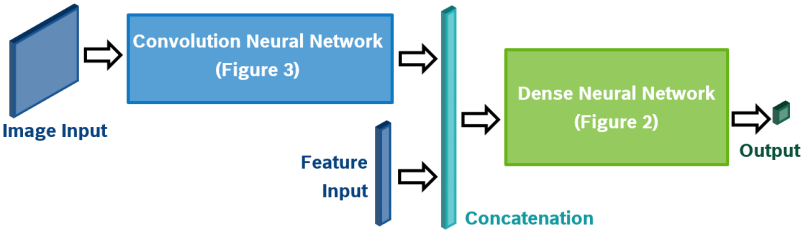


Figure 4: Combination of the networks from Figure 2 and 3. The resulting network uses both features, the ones learned from image data and the hand-crafted features presented in our previous work to solve the given classification problem.

layers are again added to the network. The basic network structures are outlined in Figure 3 and 4.

4 Evaluation

For our evaluation we first provide an overview of our dataset. Afterwards we present a comparison between our previous SVM based classification results and the different neural network architectures from Section 3. All neural networks were implemented in Python using Theano [5] and Lasagne [22]. Training is performed with AdaDelta [119] optimized stochastic gradient descent.

4.1 Dataset

As mentioned in [107], our database contains car and pedestrian tracks recorded with a Velodyne laser scanner. The raw point cloud is processed according to

[101]. 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). Further details can be found in [101]. Figure 5 shows a preprocessed point cloud. Every track is associated with a precise digital map, which describes the static, urban environment, i.e. road boundaries, crosswalks and more. Altogether we use around 2000 trajectories with 100000 data points in this paper.

4.2 Hand-crafted Features and Automatic Labeling

Our previous work [107] presented a feature design and through analysis of them, therefore we will only provide a brief summary here.

Our features can be separated into two main groups. The first group contains all features that only solely relate to the pedestrian. These features are: the velocity both in 2d coordinates and as an absolute value. The distance traveled in the previous time step and two distance measures, which describe the pedestrians' position relative to the road. *dtcurb* describes the orthogonal distance to the closest road boundary (usually a curb). The second distance measure is the minimal

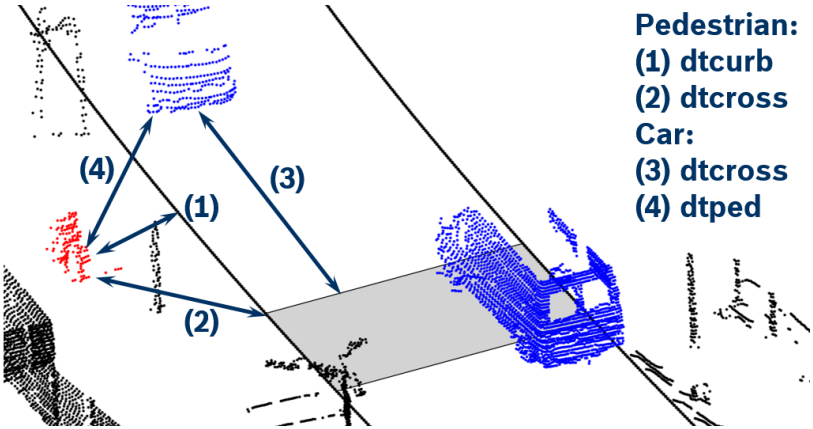
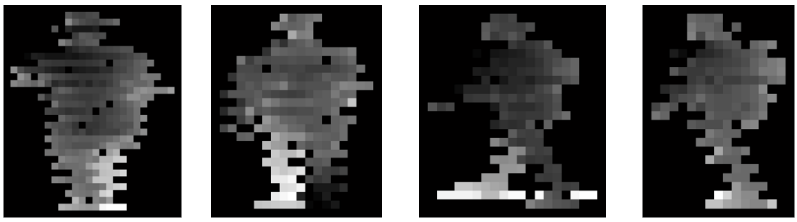


Figure 5: Example of a Velodyne point cloud with an underlying sketch of the street. The two black lines mark the curbs and the grey box symbolizes the position of the crosswalk. The image contains the following Objects: cars (blue), pedestrians (red) and background (black). The image also shows a set of geometrical features which represent the objects movement relative to the crosswalk and relative to each other. Please note that the term "dt" is used as an abbreviation for "distance to".



(a) Standing **(b)** Longitudinal Walking **(c)** Lateral Walking (1) **(d)** Lateral Walking (2)

Figure 6: Velodyne 2D image samples. The images show the Velodyne raw range measures color coded with a gray map (lighter colors correspond to smaller range measurements). The background has been removed from all images.

distance to crosswalk *dtcross*. This value will also be used in this section to provide insight on the prediction horizon. All geometrical features are shown in Figure 5.

The second feature group contains features based on the interaction of the pedestrian and a car. These features describe both the movement and position of the car (e.g. with a velocity and a distance to the crosswalk) and the "true" interaction in terms of a relative velocity and a distance between the pedestrian and the car.

Altogether this sums to 15 single features. These features are only suited to describe a single frame. To encompass temporal information we used the features from the last 4 frames as additional input for our machine learning algorithms. The total number of features sums up to 75.

In this paper we want to predict the pedestrians' intention to cross the street at a given crosswalk. Since our database contains the whole pedestrian trajectories, and our calculations are performed offline, we are able to automatically infer their intentions based on the observed movement. I.e. a pedestrians' trajectory is marked as *crossing* if we actually saw her crossing the street.

Please note that this method of automatic labeling has some disadvantages, which mainly arise due to sudden or severe motion changes. We have discussed these problems intensively in our previous work [107].

4.3 Image Data

For our first experiments we use the Velodyne [107]. This decision was made mainly because of it's 360° field-of-view and the availability of reliable object detection and tracking algorithms. Accordingly, raw LiDAR data are available for every track. The Velodyne provides for every point both the *id* of the measuring laser and the rotation angle of the sensor itself. Using these two information it is possible to create a 2D image in angular coordinates (Figure 6). Both of the Velodynes raw

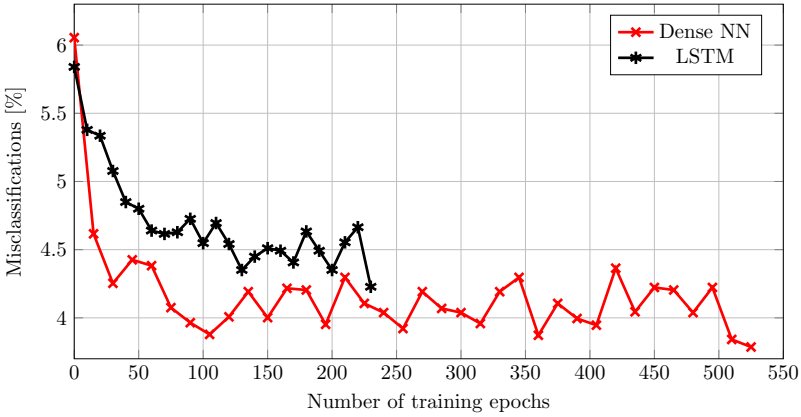


Figure 7: Training progress of the Dense Network and the LSTM are shown over the number of training epochs for one training run of the cross-validation.

measurements (range and intensity) can be used to create gray scale images if plotted with a gray color map. For our purpose we use the previously mentioned object detection to both cut the pedestrians from the gray scale range image and remove the background. Same examples of the resulting images are shown in Figure 6.

4.4 Neural Network Training and Results

For our test we separate the database into a training (80%) and a test (20%) set. In this section we use only the training set for cross validation. Initial tests have shown that using all recorded features (with input dropout) does not improve results compared to the selected, minimal feature set (a subset of all features, from [107]). Most additional features led to fast over-training without improvements on the validation set.

Most hyper-parameters were chosen by training several hundred different networks and selecting and fine-tuning the ones with the highest accuracy. The best performing dense neural network consists of three layers, each with rectified linear functions and dropout of 50%, and achieved a averaged cross-validation accuracy of about 96.21%. The number of units per hidden layer were 32, 64, and 128. Figure 7 displays the training progress over time for one training run within the cross-validation.

The recurrent network did not achieve the same level of accuracy as the simple dense networks. Our best performing LSTM, a two layer LSTM with 64 and 128 hidden units, has a 95.77% cross-validation accuracy. LSTMs outperform when

information has to be stored for a longer period of time. For pedestrians crossing the street, information about orientation and velocity from a few time-steps ago does not seem to be useful anymore. Usually, there is a, more or less, clear point where the pedestrian starts going towards the crosswalk but no prior information in their movements before that point. The advantage of the dense network is that it has simultaneous access to all currently relevant time steps and can make its decision based on all of those at the same time.

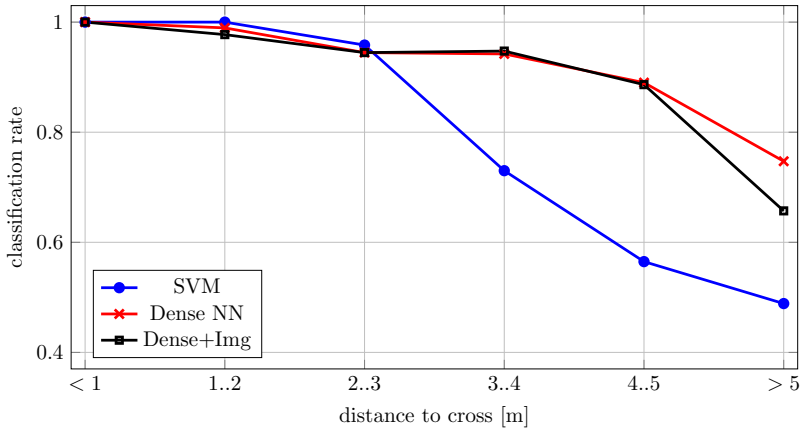
The convolutional networks did not offer additional insights into the pedestrian classification. Without the hand-crafted features, we could only achieve a 3.5% increase in classifying an input of images from 5 time-steps at a time over selecting the bias value. Adding image features to our hand-crafted input vector did not lead to any information gain. A detailed analysis of this will be given in the following section.

4.5 SVM vs. Neural Networks

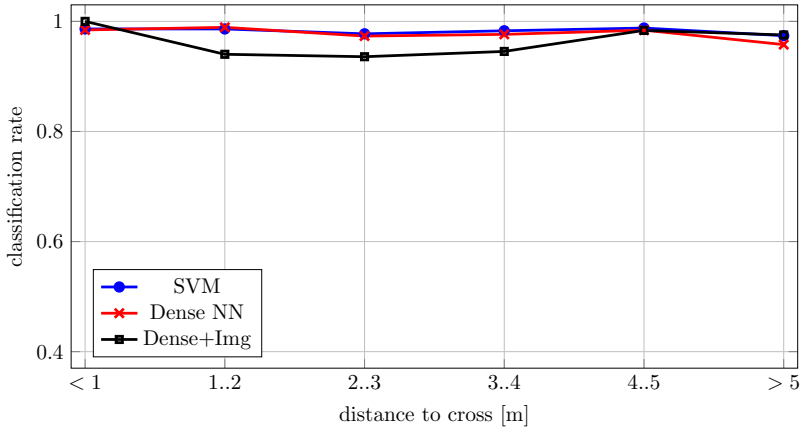
In this section we will analyze the performance of our dense classification network from Section 3 compared to the SVM from our previous work [107]. This evaluation is performed on the test set introduced in the previous section. Image 8a shows the percentage of correctly identified crossing pedestrians as a function of the distance to the crosswalk. All methods show an equally good performance for distance smaller than 3m. For all larger distances the simple dense neural network outperforms the SVM by 10 to 20%. This shows the potential of neural networks for identifying crossing pedestrians at large distances. For the combination of our hand-crafted features and the image-based features we did not obtain the expected improvement in performance. For most cases the performance is either identical or slightly worse than without the images. We assume that the major reason for this is the quality of the images. Although the Velodyne provides a 360° surround view, neither the horizontal nor vertical resolution provide detailed enough information. Usually it is possible to count the single pixels in one of these images (compare Figure 6), and especially at large distances it is possible that a pedestrian only consists of 20-40 points. Since it has been proven that image-based features can be used to improve the performance of state-of-the-art algorithms (e.g. [50]), we assume that we could achieve a better performance with a more detailed image source.

4.6 Evaluation of the Time Horizon

Usually pedestrians predictions around urban street are evaluated in respect to the remaining time-to-cross (e.g. [50]). This makes it possible to specify a temporal prediction horizon. Unfortunately, this procedure cannot be directly applied to non-crossing pedestrians. Their trajectories obviously do not cross the street and in many cases do not come close to it. Therefore it is not possible to estimate a time-to-cross for these pedestrians. Considering this together with the results from Figure 8, we decided to only analyze the crossing trajectories in this section. This means that the model is still trained with the full dataset, but the only the crossing



(a) Crossing pedestrians



(b) Non-crossing pedestrians

Figure 8: Classification results for different network structures compared to the baseline SVM. The accuracy is shown both for crossing (a) and non-crossing (b) pedestrians. The shown neural networks are: the dense network solely with hand-crafted features (Dense NN), and with additional convolution layers for feature extraction from LiDAR images (Dense+Img). For better readability the results are evaluated relative to the discretized distance to cross.

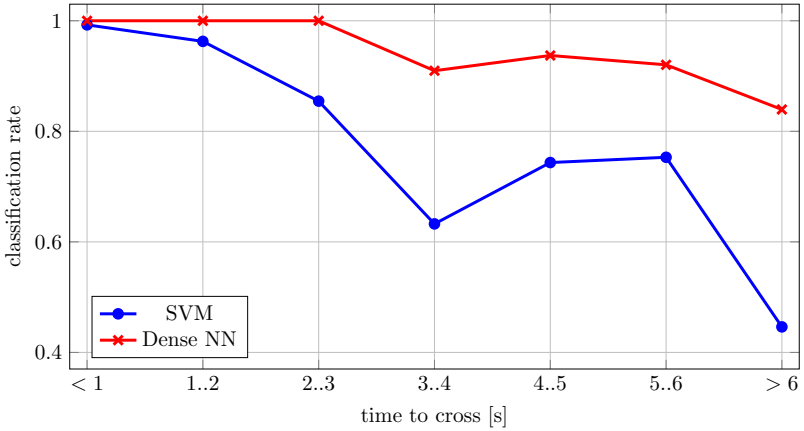


Figure 9: Time-based evaluation. The results of both the SVM and the best dense neural network are shown. The classification accuracy is only evaluated for crossing pedestrians relative to their remaining time-to-cross. For better readability the time-to-cross is evaluated for discretized intervals.

trajectories from our test set are analyzed.

The results for our best dense neural network compared to the SVM are shown in Figure 9. We can see the limitations of our SVM baseline. Mainly due to vast speed changes the classification accuracy drops very fast even for small times (< 3 s). On the other hand we can see a totally different behavior for our dense neural network, where the accuracy is never lower than 80%.

Compared to our previous distance based evaluation (Image 8a) we notice that the shown minimum accuracy is higher. The reason for this is easily explained: The highest observed time-to-cross in the shown portion of the database is 12 s. These high times correspond to a distance-to-cross > 5 m and belong to very slow walking pedestrians. Unfortunately the number of trajectories for such large times is relatively low in our current database. Therefore we decided to not evaluate the accuracy for these times. Hence for this time evaluation the slow walking pedestrians are biased by faster ones.

5 Conclusion

In this paper we proposed the use of deep learning architectures for identifying the pedestrians' intention to cross the street at a given crosswalk. First, we introduced a dense neural network which classifies intention based on features from several timesteps. Second, the time-series features are analyzed using recurrent networks,

namely LSTMs. Third, the influence of image-based features learned from LiDAR images is analyzed. We have shown that all algorithms are able to outperform the baseline SVM. The best results are achieved with the dense network with a hand-crafted feature input. This is especially the case for predicting the pedestrians' intents earlier and further away from the crosswalk. Both the LSTM and the convolutional layers did not lead to the expected improvement. Especially the LSTM suffers from missing clues for significant movement changes in the pedestrians' trajectory. E.g. a head-tracking based on high resolution images could be helpful in this situation.

The evaluation of the temporal prediction horizon showed a very good accuracy for the investigated crossing pedestrians even for large times. For the given dataset the accuracy of the proposed dense neural network never dropped below 80% for the given time horizon of 6 s.

Paper III

Predicting Pedestrian Crossing using Quantile Regression Forests

Benjamin Völz, Holger Mielenz, Roland Siegwart and Juan Nieto

Abstract

Future automated driving systems will require a comprehensive scene understanding. Considering these systems in an urban environment it becomes immediately clear that reasoning about the future behavior and trajectories of pedestrians represents one major challenge. In this paper we focus on predicting the pedestrians' time-to-cross when approaching a crosswalk. Due to the complexity of the underlying model, we propose a data-driven approach that by means of regression models learns the target variable. Instead of utilizing a standard mean regression, we propose the use of *Quantile Regression*. We show that this special type of regression is more suited to describe the variability of real world pedestrian trajectories. We examine and compare two approaches: *Linear Quantile Regression* and *Quantile Regression Forest*, which is an extended version of *Random Forests*. We present evaluations with real data and a detailed analysis emphasizing strengths and weaknesses of quantile regression for the target application.

Published in:
2016 IEEE Intelligent Vehicles Symposium (IV), 2016
DOI: 10.1109/IVS.2016.7535421

1 Introduction

To achieve safe automated driving, a number of longstanding fundamental problems need to be solved. In particular, robust perception in urban environments is still one of the most important challenges. The difficulties mainly arise from the interaction in a common environment between two groups of participants with very different dynamics: vehicles and pedestrians. Due to physical constraints, vehicles have a limited set of possible movements, which makes it somehow simpler to predict their movements, than pedestrians. On the other hand, when considering pedestrians in urban environments, motion prediction presents extra challenges. Although the speed of pedestrians is in general much lower than the ones of vehicles, pedestrians are much more agile, which introduces additional complexity to perform a motion prediction. A pedestrian can change directions very quickly, for example by turning 90° without losing speed. This high agility is what limits current systems to reliable predict pedestrians' movements for only few hundred milliseconds (e.g. [11]).

In the context of automated driving, predictions that span over much longer time horizons are required. To illustrate, consider the example shown in Figure 1 where a car and a pedestrian approach a crosswalk.

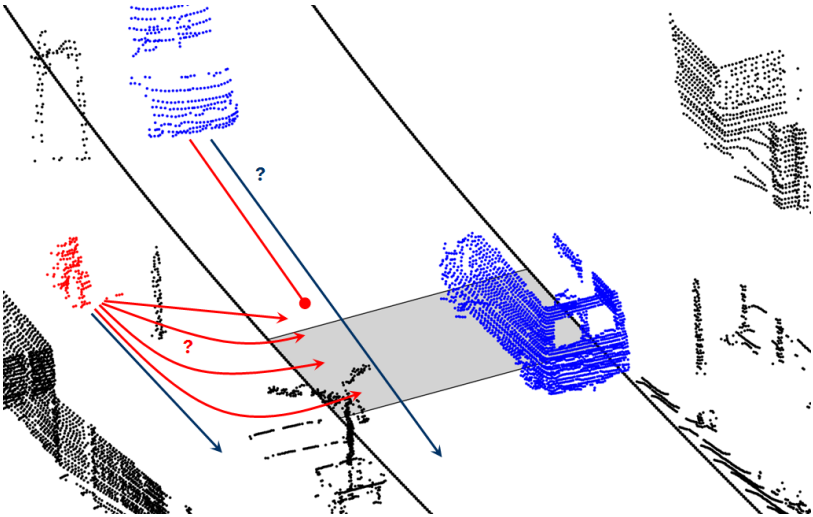


Figure 1: Typical urban scenario: a car (blue) and a pedestrian (red) are approaching a crosswalk (grey box), where the pedestrian has priority. The inference problem involves realizing whether the pedestrian intents to cross the road, and if positive, inferring when this will happen.

The car is obliged to stop if the pedestrian intents to cross the street. Timely inference of pedestrians intention is extremely difficult, and designing a system for this requires important considerations from the vehicle's perspective. First, we do not want to execute an emergency braking maneuver or apply an abrupt speed change every time. These actions would both be uncomfortable for the occupants of the car and highly dangerous for the pedestrian and other vehicles in the area. Second, we do not want to stop when unnecessary, i.e. if the pedestrian does not intent to cross the road. Late and false predictions in such situations will lead to a low system acceptance, apart from deteriorating traffic conditions. Motivated for this problem, our work aims to develop a system that (i) minimizes false detections and (ii) provides timely prediction to ensure smooth and safe maneuvers.

In our previous work we presented a method to infer the pedestrians' crossing intentions with a model learned from real world data [107].

We showed that good predictions of crossing intentions can be obtained with an SVM based pipeline trained with a minimalistic feature set. We also showed that this feature set solely depends on the pedestrians motion relative to the crosswalk.

This paper extends our previous approach by introducing a new model to estimate pedestrians actions as a function of time. Predicting spatial trajectories for pedestrians becomes quickly very uncertain due to the large possibility of movements. Therefore, our strategy is to predict discrete events in time for people's movements rather than the whole trajectory. In the context of our aforementioned example, which also represents the main focus of this paper, the target event is represented by the time-to-cross.

To derive our model we use a data-driven approach. We formulate the prediction task as a regression problem, where the time-to-cross is modeled in dependence of a large feature set (as in [107]). Regression models based on maximum likelihood will estimate the conditional mean of the target variable. This is problematic for our application, because we lose track of uncertainty, which means having no information to evaluate the system's reliability. To overcome this problem, we propose the use of *Quantile Regression* to learn not only the conditional mean but also a full probability distribution from our data. *Quantile Regression* can predict arbitrary conditional quantiles, which means we are now able to predict for example a minimal/maximal time-to-cross and derive a measurement of uncertainty.

The specific contributions of this paper are:

- analysis of pedestrian trajectories at crosswalks,
- utilization of *Quantile Regression Forests* for detailed time-to-cross predictions,
- validation and evaluation with real data and comparison with a standard *Linear Quantile Regression*,
- identification of major challenges for long-time pedestrian path predictions.

This paper has the following structure: Section 2 shows the current state-of-the-art in the field of predicting trajectories, behavior and intentions of road users in urban traffic. Section 3 provides an overview on *Quantile Regression*. In Section 4 we first present our evaluation setup and collected data. Afterwards the evaluation results are shown. The conclusion and some future work is presented in Section 5.

2 State of the Art

The state-of-the-art in intention recognition and path prediction of possible traffic participants can be divided into three categories. The first category covers the special case of highway driving, e.g. [29]. In this paper we focus on urban traffic, therefore we will not discuss this branch of the research further. The second category contains all the scenarios that are relevant for collision avoidance and warning systems [11, 40, 55, 92, 96, 120]. These approaches (Section 2.1) describe systems for anticipatory driving and address either collisions between cars, e.g. left-turn collisions, or head-on collisions with a pedestrian. All of these approaches consider and model the interaction of the two participants and predict their future behavior together. Part of the third category are all the approaches that predict solely the movement of pedestrians without considering any other road users [30, 43, 48, 50, 75]. These approaches, which usually rely only on image processing, will be addressed in Section 2.2.

2.1 Anticipatory Driving

The collision avoidance systems can be divided into two different use cases. The first use case considers only cars at intersections [40, 55, 92, 96, 120]. The main goal is usually the identification of left-turning cars and their related collision scenarios. This information is used to either display a warning or execute an emergency action, like automated braking or evasive steering.

In [40] a database-driven trajectory matching approach is utilized.

They use a labeled database of car trajectories at different intersections. For a car approaching the intersection, they use a probabilistic matching approach to find a matching trajectory in the database and infer the future behavior, e.g. turning left, of the car.

Another line of research uses probabilistic models to infer the future driving behavior. A Hidden Markov Model (HMM) and a large dataset are used in [96] to predict the drivers intention to turn at intersections.

Another important and well used model is the Dynamic Bayesian Network (DBN), which is used by [29], [120] and [55] for both state estimation and prediction of traffic situations.

A different approach that makes use of several time metrics in combination with a predefined conflict area is presented in [92].

The second collision avoidance use case covers pedestrian protection systems.

A pedestrian motion model in combination with a Bayesian Network (BN) is

used in [11] to estimate the collision risk.

Two interesting studies concerning the behavior of pedestrians at the curb are presented in [86].

It analyses: whether a pedestrian will cross the street when a car approaches. Secondly, it determines which information about the pedestrian and surrounding traffic are used by humans in order to decide whether to cross the street or stop at the curb.

2.2 Pedestrian Path Prediction

The interaction of the two road users is a crucial part of all of these systems. In this paragraph we review the state-of-the-art on pedestrian motion prediction. Most of the following papers make use of some type of information from the surrounded infrastructure.

The recognition of pedestrians' action intentions and the prediction of their trajectories are mostly based on image processing.

In [48] the contour of the pedestrians' motion is used to infer their intention to cross the street.

The shown performance is achieved due to the implicit modeling of body language traits, like the body bending and the spread of legs. Two similar approaches are presented in [43]. One is based on dense optical flow, while the other one uses a low-dimensional histogram of the optical flow. This method, which can be seen as a variant of trajectory matching, uses the measured pedestrian's position, together with so-called motion features that capture leg and upper-body movements. A different approach, that also uses body language traits is presented in [75]. The main difference is given by the combination of a sparse representation with a larger variety of used body parts.

One very important feature for pedestrian path predictions is the head orientation. [30] tracks the head with static cameras and uses it to directly estimate the future trajectory. A more sophisticated approach is presented in [50]. Here the head orientation is used to determine the pedestrians situational awareness. I.e. the head orientation is used to analyze, if the pedestrians is aware of the approaching car. The paper incorporates this measure into a DBN and shows the benefit which a head tracking could add to existing prediction algorithms.

Apart from all previously shown approaches [24] uses Gaussian process regression to model pedestrian trajectory patterns. These patterns represent the most common paths in a given environment.

2.3 Limitation of the State of the Art

A common limitation of the approaches discussed above is that, although the accuracy obtained can be in general high, the prediction horizon is typically in the order of hundreds of milliseconds. While this may be adequate for collision warning and collision avoidance systems, we argue that it will not be sufficient for urban automated driving.

To fill this gap, this paper reports on our progress in building a system that can predict pedestrians actions for larger time scales, aiming to progress towards a key need for automated driving in urban environments.

3 Quantile Regression

In the context of machine learning, regression is used to learn a specific function that describes the behavior of a dependent variable y in respect to the independent variable x. During this regression the conditional mean of y is estimated. Due to the large number of possible motions for pedestrians, track will present a varied level of uncertainty, and therefore tracking only the expected conditional mean is not enough. In this paper we present an extended method that is able to not only estimate the conditional mean, but also use the full probability distribution to compute arbitrary conditional quantiles. These quantiles can at the same time provide min and max estimates for the dependent variable y, and allow an estimation of the associated uncertainty.

Quantile Regression is a special type of regression analysis. Instead of estimating the conditional mean of the response variable, the objective is to estimate an arbitrary conditional quantile, e.g. the conditional median (i.e. the 50% quantile). In this section we introduce the two *Quantile Regression* methods that we will evaluate: *Linear Quantile Regression* (LQR) and *Quantile Regression Forests* (QRF).

3.1 Linear Quantile Regression (LQR)

The procedure for LQR [46] is quite similar to the standard regression analysis. Regression analysis tries to find an estimate of the conditional mean $E(Y|X = x)$ by minimizing the expected squared error loss,

$$E(Y|X = x) = \arg\min_z E\{(Y - z)^2|X = x\}.$$

To calculate the α-quantile $Q_\alpha(x)$ from data a slightly different loss function has to be minimized. For $0 < \alpha < 1$ let the loss function L_α be defined as,

$$L_\alpha(y, q) = \begin{cases} \alpha|y - q| & y > q \\ (1 - \alpha)|y - q| & y \leq q \end{cases}.$$

The conditional quantiles minimize the expected loss $E(L_\alpha)$,

$$Q_\alpha(x) = \arg\min_q E\{L_\alpha(Y, q)|X = x\}.$$

3.2 Quantile Regression Forests (QRF)

QRF [62] are an extension of *Random Forests* [13]. 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 and 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. I.e. 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*. 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 [62]. 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 [62]. These weights can be used to compute the estimate of the 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 $0 < \alpha < 1$,

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

4 Evaluation

In this section we will show the results of our evaluation, which was realized using a real world dataset recorded at a crosswalk in a city in southern Germany. The evaluation contains three parts. First the parameters of the QRF are estimated via cross-validation. Afterwards the results of LQR and QRF are compared using a separate test set. At the end we provide a more detailed evaluation concerning the capabilities and limitations of the presented algorithms for pedestrian path prediction.

4.1 Dataset

As mentioned in [107], our database contains car and pedestrian tracks recorded with a Velodyne laser scanner. The raw point cloud is processed according to

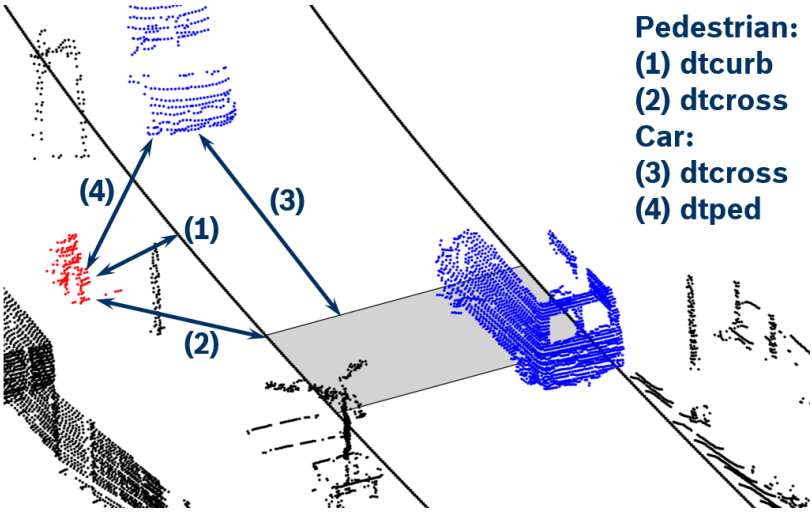


Figure 2: Example of a Velodyne Point Cloud with an underlying sketch of the street. The two black lines mark the curbs and the grey box symbolizes the position of the crosswalk. The image contains the following Objects: cars (blue), pedestrians (red) and background (black). The image also shows a set of geometrical features which represent the objects movement relative to the crosswalk and relative to each other. Please note the the term "dt" is used as an abbreviation for "distance to".

[101]. 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). Further details can be found in [101]. Figure 2 shows a preprocessed point cloud.

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

4.2 Features

Our previous work [107] included an extensive feature definition, therefore we will only provide here a brief summary.

Our features can be separated into two main groups. The first group contains all features that only solely relate to the pedestrian. These features are: the velocity both in 2d coordinates and as an absolute value. The distance traveled in the previous time step and two distance measures, which describe the pedestrians'

position relative to the road. *dtcurb* describes the orthogonal distance to the closest road boundary (usually a curb). The second distance measure is the minimal distance to crosswalk *dtcross*. This value will also be used in this section to provide insight on the prediction horizon. All geometrical features are shown in Figure 2.

The second feature group contains features based on the interaction of the pedestrian and a car. These features describe both the movement and position of the car (e.g. with a velocity and a distance to the crosswalk) and the "true" interaction in terms of a relative velocity and a distance between the pedestrian and the car.

Altogether this sums to 15 single features. These features are only suited to describe a single frame. Because of that we use the features from the last 4 frames as additional input for our machine learning algorithms. The total number of features sums up to 75.

Please note that the feature selection algorithm from our previous work is not used in this paper. Therein we found that a small feature subset, solely depending on the pedestrians trajectory, is required for a successful intention recognition. We stated that this feature set will most likely not be sufficient for further, detailed trajectory predictions. Especially the presence of cars is expected to have a significant influence on these trajectories. Therefore we use the full feature set for our evaluation.

4.3 Problem Definition and Labels

In this paper we want to predict the pedestrians time-to-cross. This time is defined as the time the pedestrian needs to reach the point where he will step on the road. Because of this aim, we will only evaluate the trajectories of actually crossing pedestrians. Since our database contains the whole trajectory for the pedestrians', we are able to calculate the time-to-cross individually for every time step of the trajectories. This method of automatic labeling has some disadvantages which will be discussed in Section 4.6.

4.4 Cross Validation

In this section the results of a 5-fold cross validation for the *Quantile Regression Forests* are presented. The QRF has three parameters:

(1) *mtry* – number of (random) features to try in each tree (feature bagging).

(2) *nodesize* – minimal size of terminal nodes. A larger value results in smaller trees and vice versa.

(3) *ntree* – number of trees to grow.

According to [62] the algorithm is usually stable for a wide range of these parameters. Our results support this statement.

The results are calculated for the 10% and 90% quantiles. Therefore a time interval is defined as the difference between the prediction of these two quantiles. $\Delta t = Q_{0.9}(x) - Q_{0.1}(x)$. Furthermore a prediction is marked as correct, if the true

observation lies within the predicted quantiles. The best parameter set will provide the largest number of correct predictions with the smallest time interval. In this context a small time interval is equal to a small prediction uncertainty. The time intervals are averaged over all observations.

Depending on the case the results are either shown as one value over all features or relative to the distance to the crosswalk *dtcross*. This detailed examination provides a more meaningful representation of the results than the simple calculation of an overall value for all features, but it is only shown when a significant difference between two values was found.

The result for *mtry*, the number of features to use in each tree, is shown in Table 1. If we would only consider the percentage of correct predictions a small value of *mtry* would provide the best results. But this high percentage comes at the price of higher time intervals. A good compromise for *mtry* could be to use 20 because it combines a high prediction performance with low time intervals.

If we consider the minimal size of the terminal nodes, shown in Table 2, one can see easily see that the best result is obtained for size 1. The percentage of correct predictions is high (93.12%) and all time intervals (for all distances) are similar. The only downside of this choice is that a small *nodesize* results in large trees. This could be computationally demanding, if combined with a larger number of trees.

Table 3 shows the overall prediction performance relative to the number of grown trees. The differences are not significant but a small peak can be found for 1000 trees. In this case the resulting time predictions are not shown because they are almost equal in all cases.

4.5 Comparing LQR and QRF

This section compares the results of LQR and QRF. Both models are trained on the same training set and evaluated with a separate test set that was not part of the cross validation in Section 4.4. This test set has some peculiarities which definitely lead to poorer performance. These peculiarities will be further discussed and analyzed in Section 4.6. The results for the LQR and QRF are shown in Image 3a and 3b. Table 4 supplements the figures with numerical values. In general it can be seen, that both methods provide similar results, however observing the numerical values it can be seen that QRF is in average more accurate. If the predicted time intervals are compared, one can see that the QRF provides much narrower intervals for small values of *dtcross*, but overestimates the intervals for larger *dtcross* values.

The LQR on the other hand shows a different behavior. The predicted intervals do not grow very large due to its linearity constraints, but this also results in very large intervals for small values of *dtcross*. Normally we would assume that there are less errors if the algorithm produces larger intervals (compare Section 4.4). Image 3a shows a different picture. The LQR produces also a large number of errors for small times (and therefore small values of *dtcross*). This includes also the prediction of negative crossing times. Please note that especially all red dots which are below the x-y-line are particularly bad, because the pedestrian moves

Table 1: The performance relative to the number of random features used in each tree is shown. Both the percentage of correct predictions and the predicted time intervals relative to the distance to the crosswalk are shown.

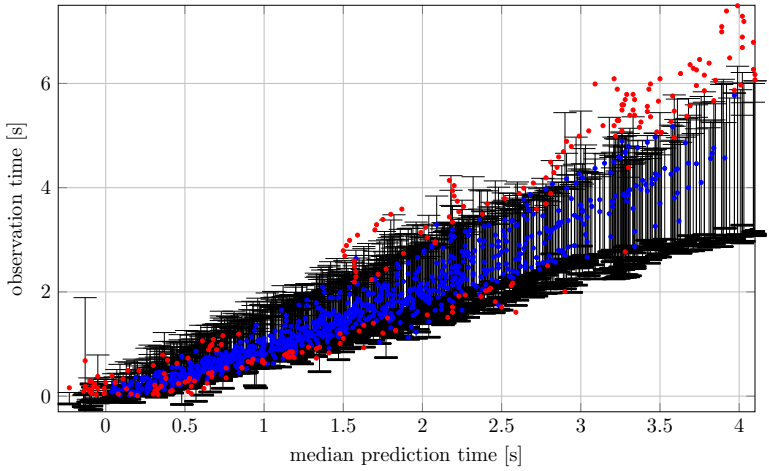
mtry	5	10	20	30	50	70
correct [%]	95.49	93.61	91.78	90.55	89.28	88.62
dtcross [m]	time interval [s]					
$0-1$	0.45	0.36	0.32	0.31	0.29	0.29
$1-2$	1.06	0.96	0.91	0.90	0.88	0.88
$2-3$	1.31	1.18	1.12	1.10	1.07	1.05
$3-4$	1.57	1.46	1.41	1.38	1.35	1.34
$4-5$	2.58	2.28	2.21	2.22	2.24	2.27

Table 2: Impact of the size of the terminal nodes on the performance of the QRF. Both the percentage of correct predictions and the predicted time intervals relative to the distance to the crosswalk are shown.

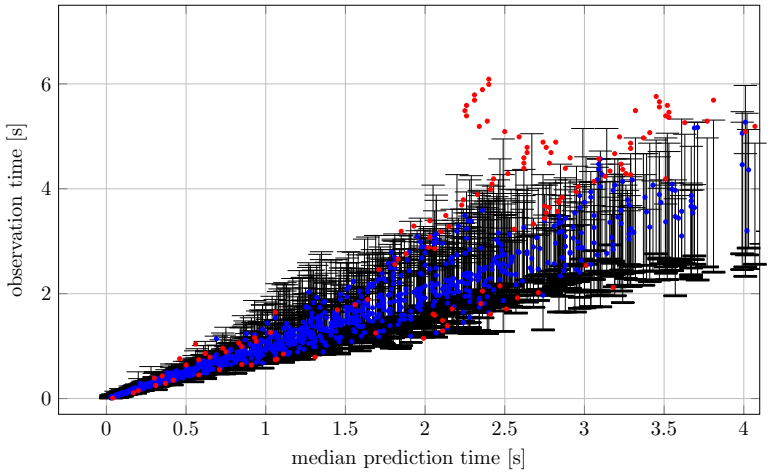
nodesize	1	3	5	10	15	20
correct [%]	93.12	92.62	92.33	91.04	90.80	90.75
dtcross [m]	time interval [s]					
$0-1$	0.32	0.32	0.32	0.31	0.32	0.33
$1-2$	0.92	0.92	0.92	0.90	0.94	0.98
$2-3$	1.09	1.10	1.10	1.11	1.16	1.21
$3-4$	1.30	1.33	1.35	1.39	1.48	1.56
$4-5$	1.90	1.96	2.02	2.22	2.46	2.67

Table 3: The table shows the overall performance relative to the number of grown trees. The predicted time intervals are stable over all trees and therefore not shown.

ntree	50	100	500	1000	1500	2000
correct [%]	90.45	90.52	91.01	91.06	90.93	90.99



(a) *Linear Quantile Regression.*



(b) *Quantile Regression Forests.*

Figure 3: LQR and QRF result: every dot represents one measurement and is either blue, if the observed value lies within the predicted interval, or red otherwise. Every dot is associated with a black line. This line represents the predicted interval, which is here given by the difference between the 10% and 90% quantile. (Best viewed in color.)

Table 4: Comparison of the LQR and QRF results. Both the performance over all features and the associated time intervals for certain intervals of the distance to the crosswalk are shown.

	LQR	QRF
correct [%]	77.72	84.42
dtcross [m]	time interval [s]	
$0-1$	0.69	0.37
$1-2$	1.15	1.13
$2-3$	1.60	1.38
$3-4$	2.06	2.29
$4-5$	2.96	3.45

actually faster than predicted. In an automated driving system this could result in the necessity to trigger an emergency stop (if the difference between prediction and observed value is large).

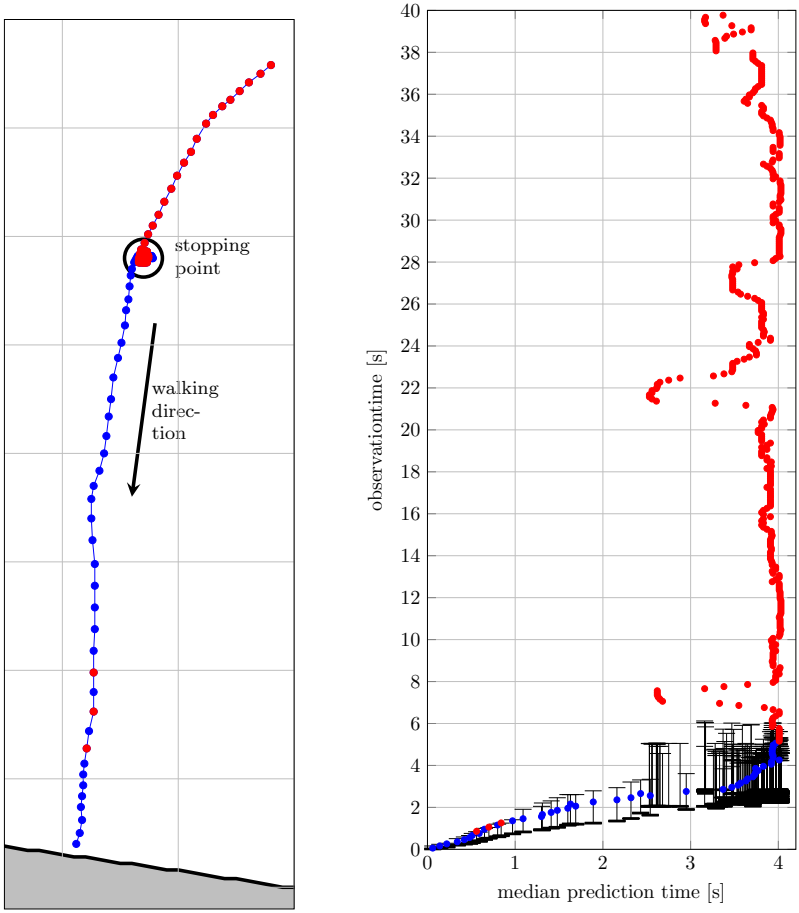
Image 3b shows that the QRF is much better suited for predictions of small times. Additional to the narrow intervals the number of errors is lower than with LQR.

4.6 Analysis of the results

This section shows an analysis of the cases the system fails to provide good results. For the analysis we will use the QRF prediction and compare against the actual trajectories. Please note that all shown trajectories show the behavior of real humans, neither actors nor remote controlled dummy's where involved.

As mentioned in Section 4.1: since we know the whole track of all objects beforehand and therefore can calculate the point in time when the pedestrian first step on the crosswalk, it is feasible to calculate the remaining time-to-cross for every time step of every pedestrian track. However, this can lead to systematic errors in all cases in which a pedestrian changes his movement drastically. The resulting errors can be divided into three major error types.

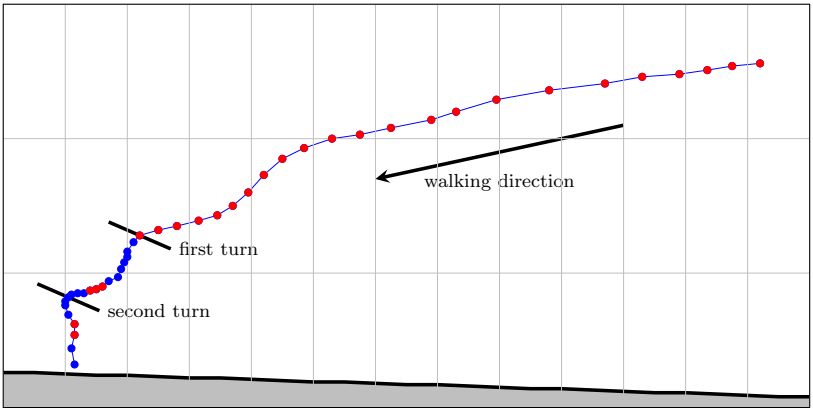
First, a pedestrian can come to a complete stop and wait at the roadside for several seconds (Figure 4). This leads to a theoretical wrong prediction (up to the point in time where the pedestrian starts to walk again), because the stopping and standing maneuvers are not recognized (and intentionally not part of the training data). However if we take a look on the associated prediction (Image 4b), we can see that the prediction is intuitively correct because it predicts a possible time-to-cross of $\sim$ 4s seconds which equals to the real time to cross after the pedestrian starts walking again. The second error type belongs to pedestrians who change their movement velocity drastically, e.g. start running. This case represents a potentially serious security risk for every automated driving system. The third and



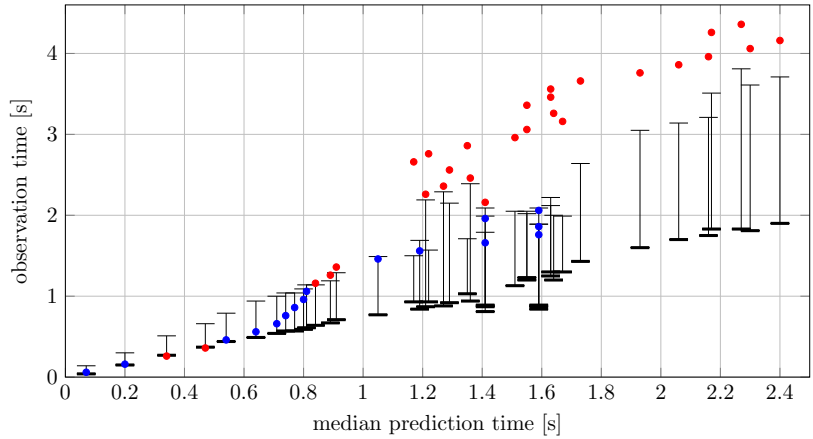
(a) Trajectory: the stopping location is marked with a circle. Every dot represents one time step of the trajectory and is marked either in blue for a correct prediction or red otherwise. The road is depicted at the bottom of the image.

(b) QRF prediction: the red dots represent the globally wrong measurements (observed time $\gg$ predicted time). 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.

Figure 4: 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. Large gaps between the single steps (dots) represent high velocities and vice versa.



(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.

Figure 5: The shown pedestrian runs first, then slows down and reaches the crosswalk after doing several sharp turn. The figure shows both the trajectory and the corresponding QRF prediction.

last error type represents a similarly dangerous situation. These error type contains pedestrians who are doing sharp (e.g. 90°) turns. The associated prediction before a turn is usually very different from the real remaining time-to-cross. Figure 5 shows an example of a pedestrian who both does several sharp turns and changes his velocity repeatedly.

In summary our analysis shows that almost all errors belong to humans who considerably change their velocity or walking direction. The predicted time-to-cross is therefore only locally but not globally correct. In this context the largest challenge is the fast detection of these changes. Only with this fast detection it is possible to provide a reliable prediction with low uncertainty. In all other cases it is possible to predict a reasonable time interval using *Quantile Regression*. The prediction of this time interval considers the possible variabilities in the current situation and enables a more sophisticated decision making for any automated driving system.

5 Conclusion

In this paper, we introduced a method for detailed predictions of pedestrian crossing behavior at urban crosswalks in terms of the remaining time-to-cross. We proposed to use *Quantile Regression* to supplement and extend the prediction of a standard conditional mean regression. With *Quantile Regression* it is possible to depict the complexity and variability of typical pedestrian behaviors in the prediction. We are able to predict the conditional mean together with arbitrary conditional quantiles. These quantiles can be used to both provide a time interval for the possible crossing and an estimate for the associated uncertainty, through the size of this time interval. We implemented and compared two *Quantile Regression* methods, the *Linear Quantile Regression* (LQR) and the *Quantile Regression Forests*. Our results showed that the QRF produces better results than LQR when the time-to-cross is less than three seconds. On the other hand, the LQR in general provides a poor approximation of the underlying complexity and variability of the pedestrians' movements due to its linearity. We have shown that the performance of the algorithm for larger time-to-cross values is limited due to both the large variability of possible pedestrians motions (compare Figure 1) and the ability of the underlying tracking algorithm to fast detections of motion changes (e.g. sharp turns).

Paper IV

Inferring Pedestrian Motions at Urban Crosswalks

Benjamin Völz, Holger Mielenz, Igor Gilitschenski, Roland Siegwart and Juan Nieto

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 them. 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 data set 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.

Published in:
IEEE Transactions on Intelligent Transportation Systems, 2019
DOI: 10.1109/TITS.2018.2827956

1 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, 56]. 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. [11]) and are usually used for pedestrian protection systems.

In this paper we want to address the problem of pedestrian intention prediction.

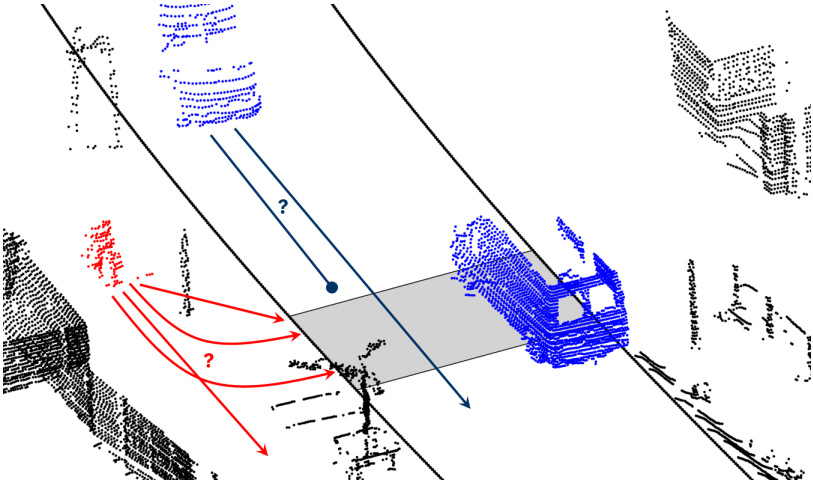


Figure 1: Point cloud depicting a typical urban scenario: a car (blue) and a pedestrian (red) are approaching a crosswalk (grey box), where the pedestrian has priority. In this scenario we want to infer the pedestrians future motion. Some possible motions are depicted as red arrows. This information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

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 [107, 109] 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 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 Figure 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* [46]. 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 2 shows the current state-of-the-art in the field of predicting trajectories, behavior and intentions of road users in urban traffic. Section 3 introduces the hierarchical prediction system and the extended features set. An overview on the pedestrian intention recognition algorithms will be presented in Section 4. Section 5 comprises the theoretical foundation of the *Quantile Regression* and the corresponding prediction of the time-to-cross and distance-to-cross. Section 6 provides an overview of our dataset and the evaluation. Conclusions are presented in Section 6.

2 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 [11, 43, 50]). approaching the road orthogonally

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 [86]. 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 [48] 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 [43]. 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 [75], 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 [50]. 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) [66] 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. non-vision 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 [88]. We will use such a IMM filter based prediction as basis of comparison for our evaluation in Section 6. Again in the context of collision avoidance systems, [11] 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 [24]. These patterns represent the most common paths in this specific environment. In [51] 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 [7]. Here, both global, and 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 [3]. 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 [41]. 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.

3 System Description

Predicting pedestrian movements is a highly complex task. As stated in Section 1 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 3.1. Additionally we will describe the feature set used within our entire inference processes in Section 3.2.

3.1 Hierarchical Prediction System

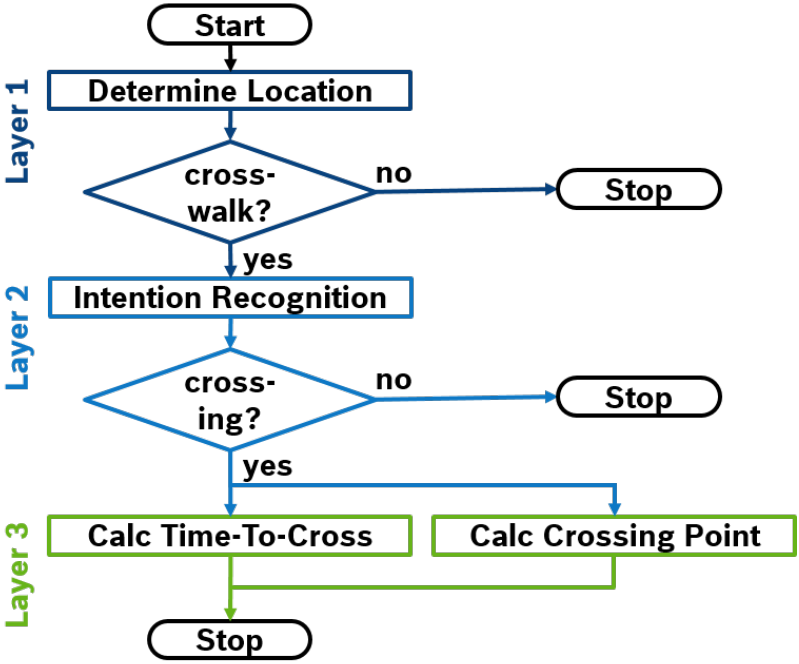


Figure 2: Flowchart of the proposed hierarchical system architecture. First a geographic area, e.g. a crosswalk, is chosen. The second layer identifies the pedestrians intention to cross the street in the given area. Afterwards the third layer calculates relevant detailed predictions, e.g. the remaining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

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 [29] 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 4). The third and last main layer contains all the inferences of continuous variables which are approached with regression methods (Section 5). 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.

3.2 Features

For the machine learning algorithms in the following sections a meaningful set of features is necessary. Based on our previous work [107] 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 [58]. Each filter represents a different motion model, typical models can be found in [90].

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 6.5) 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.

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

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 6.2.

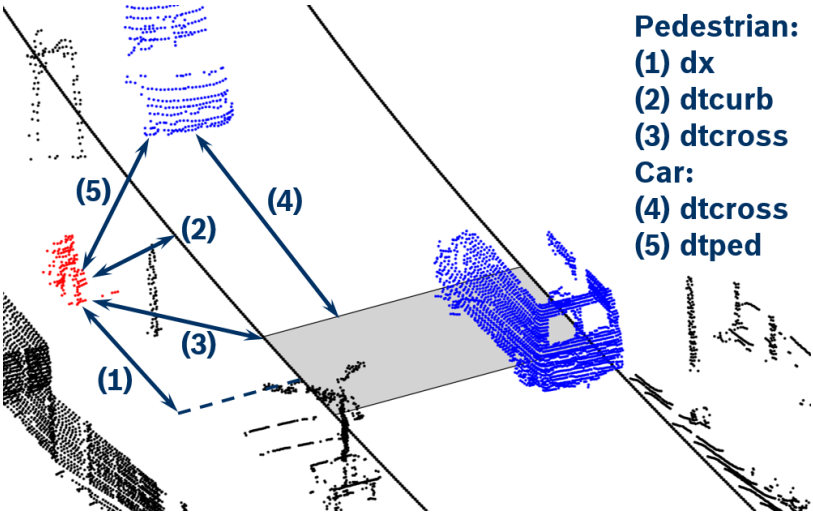


Figure 3: All relevant distance measures for the interaction of all relevant dynamic objects both with the map and each other are shown. The underlying image shows a Velodyne Point Cloud with an sketch of the street. The two black lines mark the curbs and the grey box symbolizes the position of the crosswalk.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

Track history

Within our previous work [107] 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 $dx(t-4)$.

4 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 [107]. 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 [107] we analyzed the most relevant features for identifying pedestrians' intention to cross the street. We use an algorithm called Recursive Feature Elimination (RFE) [36]. 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 [107]. 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.

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

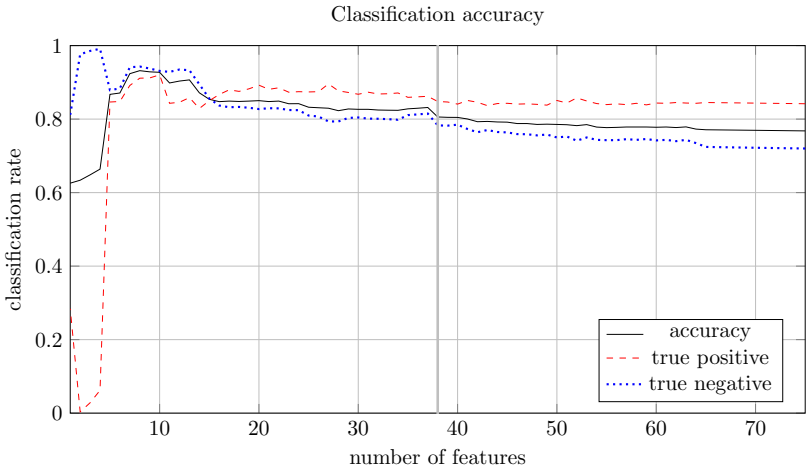


Figure 4: Result of the single feature elimination. The classification accuracies are plotted over the number of used features. The Evaluation is carried out on a time step basis, therefore the results show the perecentage of correct classified time steps of all trajectories. Please note that this is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

Another possible implementation of the feature relevance estimation is the so called group elimination [36]. 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.

5 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 [109] we compared different *Quantile Regression* algorithms and decided to use *Quantile Regression Forests*, which will be introduced in the next section.

5.1 Quantile Regression Forests (QRF)

QRF [62] is an extension of *Random Forests* [13]. 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* [13]. 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 [62]. 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 [62]. 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 $0 < \alpha < 1$,

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

5.2 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 6.2, 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 [109], 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 6).

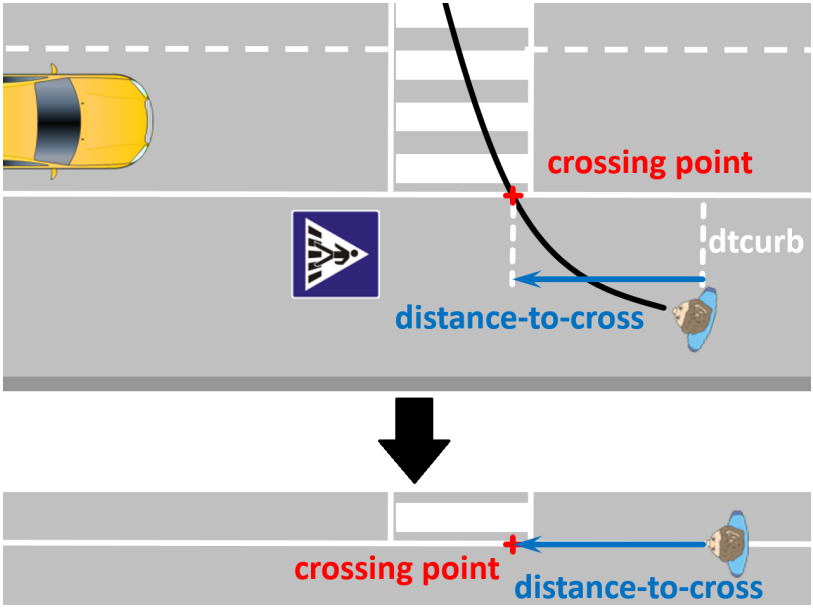


Figure 5: Definition of the crossing distance label. The 2D problem in the global coordinate frame can be projected into a 1D representation, because the distance to the curb *dtcurb* is known. If the distance-to-cross is known, it is easily possible to calculate the corresponding *crossing point* in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

5.3 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 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.

6 Evaluation

Our evaluation is composed by two main parts. Before we start with the evaluation itself, the metrics employed will be discussed in Section 6.1, followed by a description of the datasets in Section 6.2.

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 6.4). This section will especially analyze the level to which a model generalization might be possible.

Finally in Section 6.5 we discuss the overall remaining challenges, which limit the performance in general.

6.1 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 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 3.2 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.

6.2 Dataset

Our database contains car and pedestrian tracks recorded with a Velodyne laser scanner. The raw point cloud is processed according to [101]. 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 6.4 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 6.5.

6.3 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 [109].

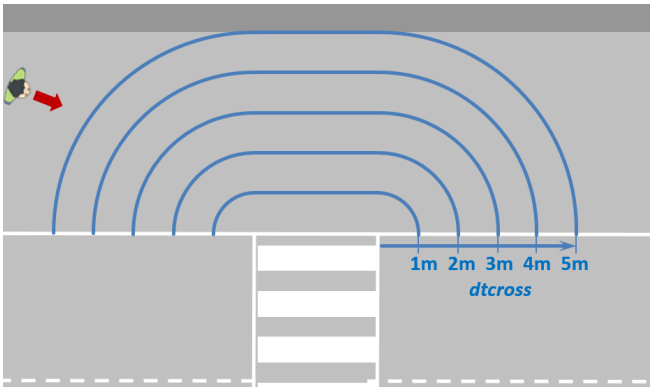
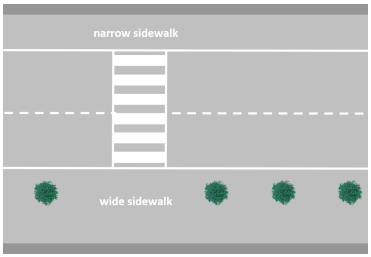
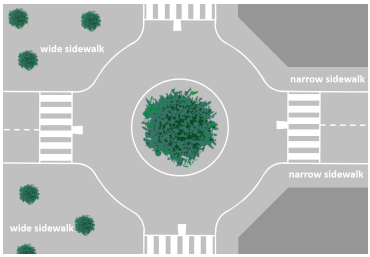


Figure 6: The distance-based evaluation principle is shown. All further evaluations will provide performance measures relative to the pedestrians distance to the crosswalk *dtcross* as a measure for the prediction horizon.



(a) Straight road.



(b) Roundabout.

Figure 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).

Intention Recognition
Compared with our previous work [107], 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 6.5.

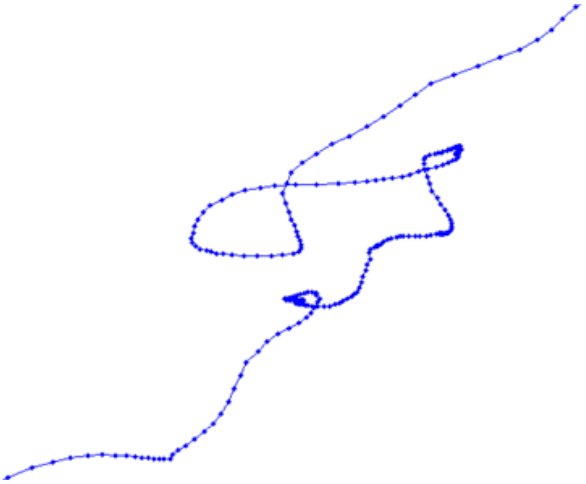
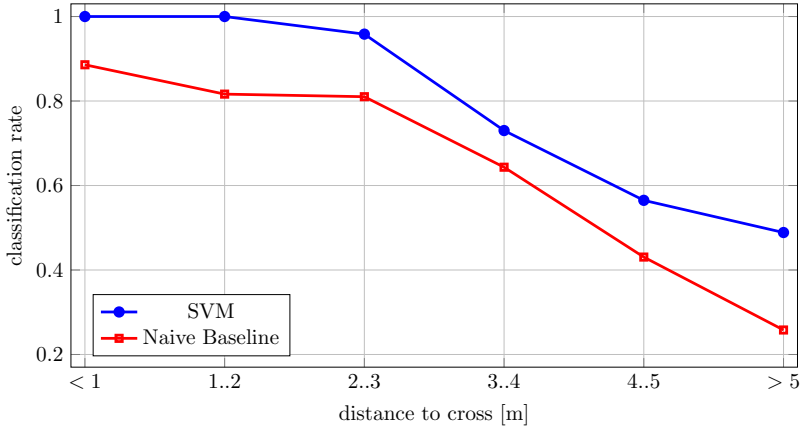


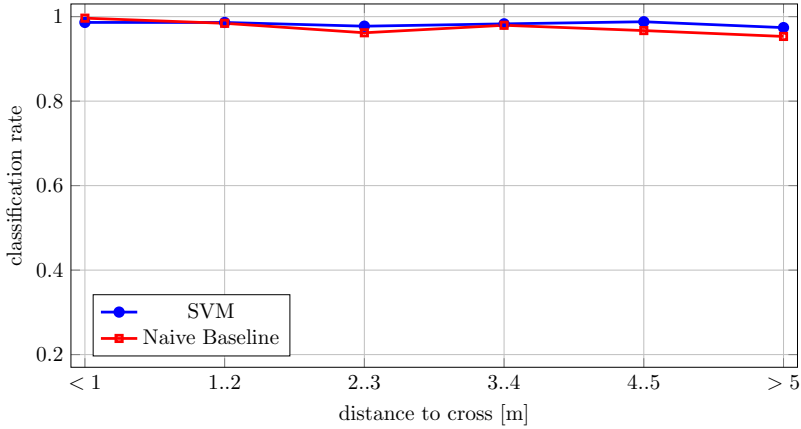
Figure 8: Trajectory of a pedestrian moving in several circles before moving towards the road (road not shown). The trajectory starts in the lower left corner and visualizes each measurement as one blue dot. Such trajectories show confusing behavior that is almost impossible to label properly and could 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 6.5.

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



(a) Crossing pedestrians (True Positive)



(b) Non-crossing pedestrians (True Negative)

Figure 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.

ans will walk parallel to the street before doing a late turn towards the crosswalk. These two behaviors are inseparable for most large distance measures, which results in a poorer performance accuracy.

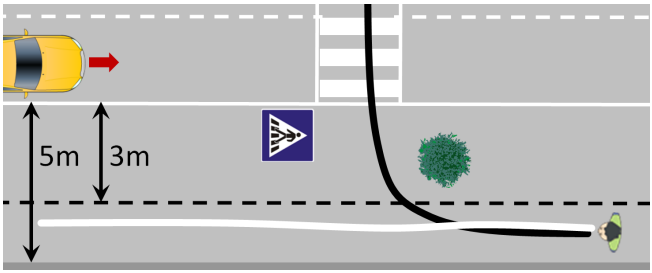


Figure 10: Typical trajectories at a crosswalk. White: Trajectory of a pedestrian, who passes the crosswalk with a constant *dtcurb* of $3-5$m. Black: Crossing pedestrian, who is walking parallel to the street for a long time before turning towards the crosswalk. Since both trajectories are more or less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

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 5.3, Table 1 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.

Table 1: 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

6.4 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 6.3 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 2 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 2, 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.

The last column of Table 2 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.

6.5 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. Although some of the problems may be unique to our combination

Table 2: Intention recognition generalization test for different crosswalks geometries. The results from Section 6.3 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%

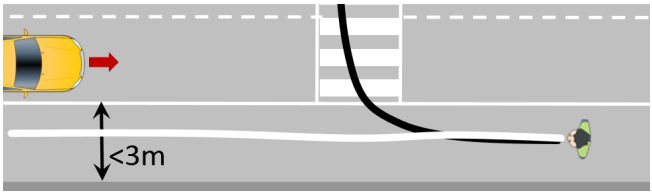


Figure 11: Typical trajectories at a *narrow* crosswalk. White: Trajectory of a pedestrian, who passes the crosswalk with a constant *dtcurb* of $1 - 2$m. Black: Crossing pedestrian, who is walking parallel to the street for a long time before turning towards the crosswalk. Since both trajectories are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

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:

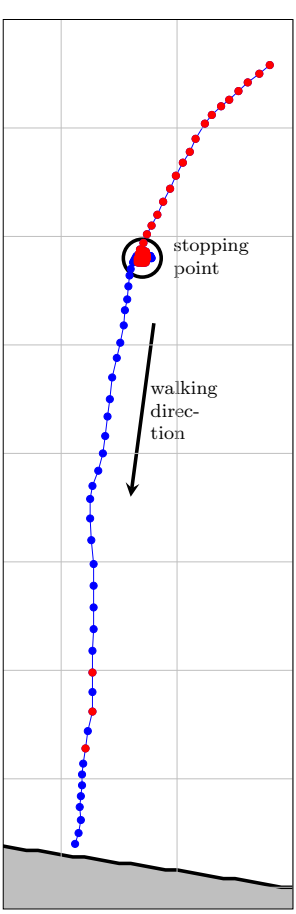
- high accelerations (or decelerations),
- sharp turns,
- stopping, usually combined with some movement on the spot.

To explain the problems, we first should recall the previously described automatic labeling procedure (Section 6.2). 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.

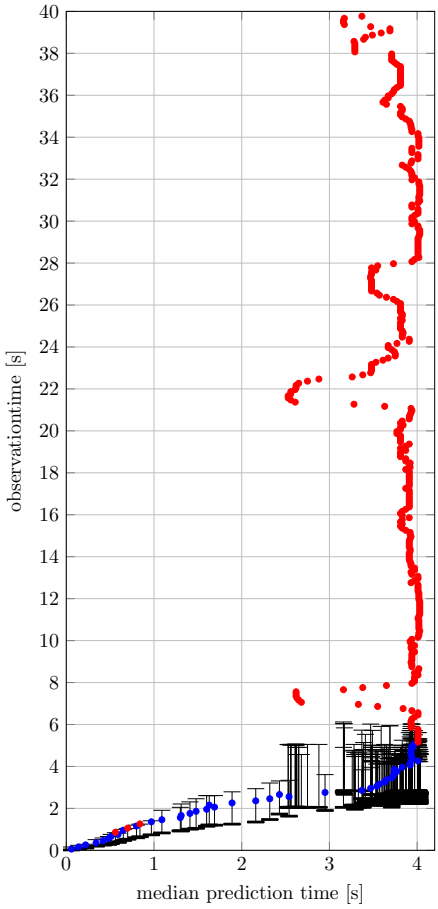
Figure 12 depicts a pedestrian who will cross the street, but suddenly stops and waits at the roadside for several seconds. Since our automatic labeling framework is not able to detect this stop, our algorithm provides a theoretically wrong prediction (Image 12b). However, if we take a closer look at 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 3.2), 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 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 Image 13a. 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 (Image 13b), we can again see that, although labeled as wrong, we got exactly the prediction which

[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.

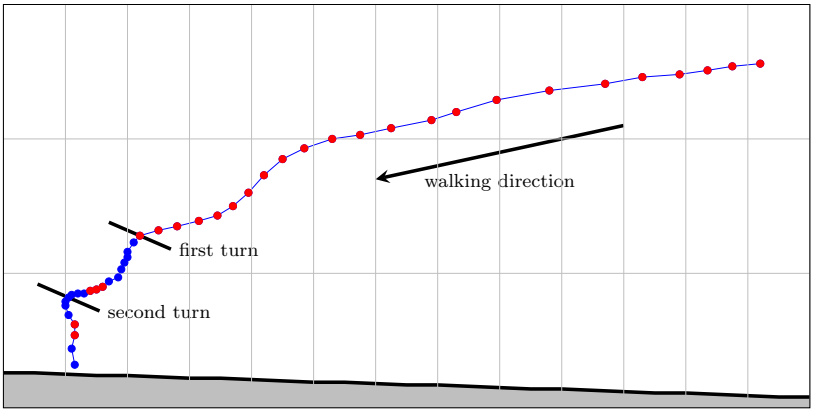


(a) Trajectory: the stopping location is marked with a circle. Every dot represents one time step of the trajectory and is marked either in blue for a correct prediction or red otherwise. The road is depicted at the bottom of the image.

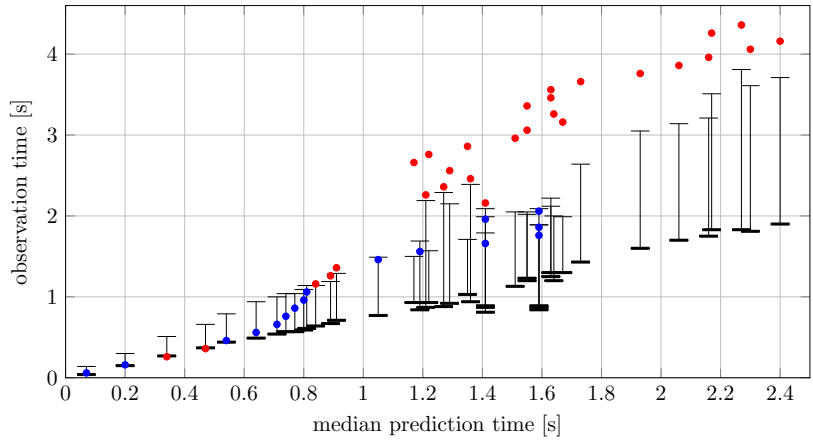


(b) QRF prediction: the red dots represent the globally wrong measurements (observed time $\gg$ predicted time) and blue the correct ones. The prediction represents the time a pedestrian would need to cross the street if he would continue walking in similar manner. The prediction can therefore be seen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

Figure 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.

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

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 [50].

6.6 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 3. 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 5) 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 3: 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.

7 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.

Paper V

Towards Infrastructure-Supported Planning for Urban Automated Driving

Benjamin Völz, Axel Stamm, Matthias Maier, Rüdiger-Walter Henn, Roland Siegwart and Juan Nieto

Abstract

Visual obstructions in urban areas are a major challenge for automated vehicles. Safely handling these obstructions requires usually a very defensive driving style, because the automated vehicle has to slowly advance e.g. into an unobserved intersection until the sensors are able to observe a priority road. To overcome this problem, that essentially impairs both traffic flow and passenger safety, we propose to utilize additional sensor information provided by road side sensors via vehicle-2-x communication. We shortly introduce the publicly funded project MEC-View that provides the infrastructure for these experiments. We provide an overview on both how an automated vehicle can incorporate the additional infrastructure information into its processing chain in general, and provide detailed insight on how these additional informations are handled in our vehicles behavior and motion planning. We present an evaluation based on simulated data visualizing the potential benefit regarding time savings and passenger comfort for multiple traffic scenes depicting different traffic densities.

Published in:
Robotics: Science and Systems (RSS). Workshop on Scene and Situation Understanding for Autonomous Driving, 2019

1 Introduction

Highly automated driving is more and more approaching actual real world applications in urban areas. Although most automated test vehicles are equipped with a 360-degree panoramic vision based on multiple redundant and complementary sensor technologies, a major problem remains: visual obstructions. Usually none of the typical sensor technologies (radar, lidar, camera) is able to see through solid objects. Only the radar is sometimes able to observe concealed objects via reflections on the road surface. Apart from that, any visual obstruction, like a parked truck or a wall, produces a especially challenging scenario for automated vehicles. Because of safety requirements, where the automated vehicle is only allowed to drive into observable areas, these scenarios remain either infeasible or only feasible with massive performance and comfort degradation.

To solve this problem we propose to use vehicle-2-x communication (V2X). Cooperative information can enable new functionality and even improve traffic flow [34]. In this paper we want to focus on one specific aspect: vehicle-2-infrastructure

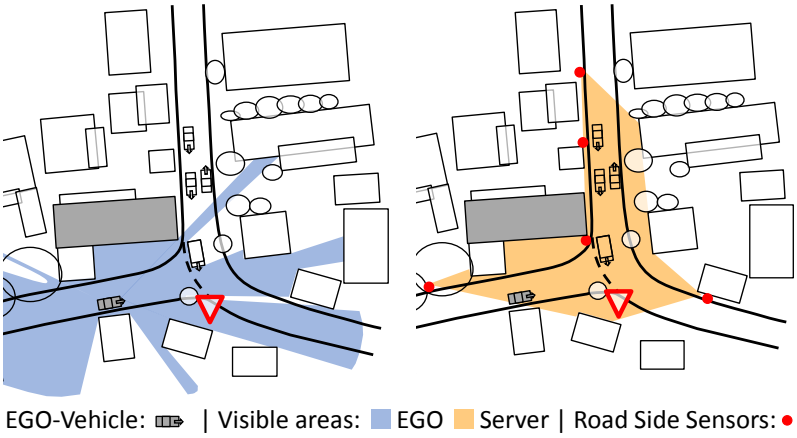


Figure 1: MEC-View test area in Ulm, Germany. The images show the intersection with the minor road approaching from the lower left corner. The intersection features a building (large grey box) placed directly at the priority road leaving a pedestrian walkway of less then two meters. This results in a major visual obstruction for the EGO-vehicle (grey) approaching on the minor road. The visible area of the EGO-vehicle is shown in blue on the left side. The road side sensors are marked with red dots and the combined visible area of the server is shown in orange (right image).

(V2I) communication. Within the project MEC-View [61] a pilot intersection is equipped with multiple sensors. These sensors can be used to effectively extend the field-of-view of our automated vehicle. Figure 1 shows the target intersection together with an impression of possible visible areas for both an automated vehicle (left) and the infrastructure (right). The special feature of the selected intersection is a challenging layout containing a massive visual obstruction due to a building placed directly at the intersection (compare large grey box in Figure 1).

Within MEC-View the data collected by the road side sensors are aggregated in a local Mobile Edge Computing (MEC [33]) server. The server aggregates, fuses and tracks all infrastructure data and provides, among other things, an object list to any interested automated vehicle. This enables the vehicle to do a more profound and safe planning for it's own actions and manoeuvres. During the project we want to analyze the potential benefit of such a system for urban highly automated driving. We provide on overview on how we can use these additional information within our planning environment to drive safe, as well as, time and jerk optimal. Our evaluation compares two cases, driving with and without additional infrastructure information while yielding and merging to a priority road. This initial evaluation is conducted based on simulated traffic at the target intersection.

The remainder of the paper is structured as follows. Section 2 summarizes the state-of-the-art regarding both connectivity based automated driving and related planning algorithms. An overview of our general planning architecture is introduced in Section 3, and the specific algorithms for generating safe and comfortable trajectories for merging onto a priority road are presented in Section 4. Section 5 introduces our evaluation simulation setup, as well as the evaluation itself. Finally the conclusions are presented in Section 6.

2 Related Work

Analyzing and optimizing the traffic at intersections has been a major research areas in recent years. A general survey on different cooperative intersection management systems, including both signalized and non-signalized intersections, can be found in [17]. The survey contains a well structured overview on different topics including traffic control through optimisation of traffic lights [115] and cooperative methods like: negotiation for green lights [16], space and time slot reservations [23], cooperative trajectory planning [57] and much more. Our work focuses on mixed traffic at non-signalized intersections, therefore we will not further look into traffic lights and related algorithms here.

In [72] an overall planning system for both highway platooning and urban intersection handling is presented. The system contains a layered architecture including a strategic, tactical and operational layer. The tactical layer coordinates the cooperative manoeuvres based on defined interaction protocols. In [68] a motion planning approach is presented that uses a parallelized cooperative and safety planner. The cooperative planner tries to find a global optimal solution with a Multi agent Markov Decision Process (MMDP). Considering only partially

cooperative vehicles, [73] introduced a priority-based approach where automated vehicles would cross the intersection without traffic lights or stop signs, while all non-cooperative traffic would be regulated with traffic lights. Here the automated, cooperative vehicles would be prioritized over other traffic, which could potentially adversely affect social acceptance.

Handling visibility information is essential for urban driving. The basic effect of visibility on planning is addressed in [12]. They use a Partially Observable Markov Decision Process (POMDP) for tactical decision making under uncertainty. [69] computes the possible visibility on the priority road from given road side geometry. Based on this they are able to infer when and where they have full visibility at the priority road and how fast they can approach this point, to still be able to come to a full stop, if required.

Automated vehicles usually utilize a complex system architecture including many different layers. In [70] a set of different architectures is summarized and compared regarding robustness, e.g. against measurement errors. Our two most important planning components are: a behavior planner and a motion planner. Behavior planning usually focusses on tactical decisions, like target lane or velocity selection, while the motion planner is responsible for calculating safe and comfortable trajectories. A general overview on different motion and trajectory planning algorithms can be found in [31].

3 Planning Environment

Our planner is separated into three base layers, route planning, strategic behavior planning and detailed motion planning (Figure 2).

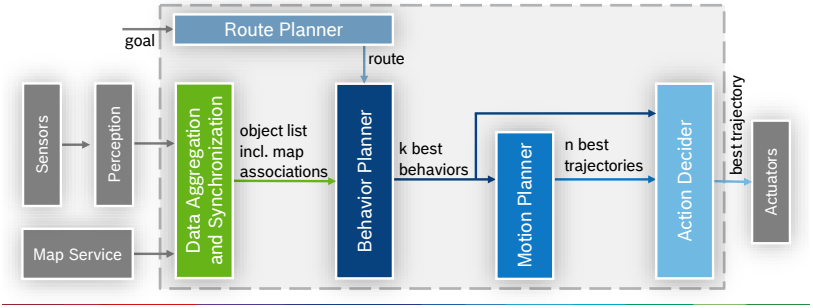


Figure 2: Simplified overview of our planner structure containing a route, behavior and motion planning.

For this use case our input combines a list of dynamic objects together with a precise map. The map contains both geometric information (lane boundaries, static

obstacles), and semantic information (traffic signs and rules, like speed limits and lane priority information).

The behavior planner analyzes the overall situation together with the route goal to find the current best possible action. Each action contains at least a target lane and max speed. Additionally there may be further information encoded, like position of next relevant stop points and lead follow vehicles. The behavior planner always provides multiple actions for further evaluation. One possible combination would be: a action driving through an intersection, a fallback option for stopping in front of the intersection, and a emergency action, which could e.g. trigger an immediate full brake.

The motion planner has the responsibility to calculate the best possible trajectory for any given behavior plan. We use a sampling based motion planner that samples and evalutes trajectory bundles for different low level tasks. Each of the tasks has a different sampling strategy and a specific cost function. Our main tasks are:

- Free Driving: samples to reach a target velocity at different times.
- Vehicle Following: samples to match both a target follow point and velocity, both defined by the prediction of our lead vehicle.
- Stopping: samples to reach a specified point in space at different times.
- Synchronizing: will be introduced in Section 4.2.

The cost functions are also task specific and weight different terms for target achievement, safety and comfort.

4 Synchronized Merge

Considering a traffic scene where our automated vehicle has to yield to priority traffic. If the overall scene is known, we are able to infer the best possible behavior and trajectory long before reaching the intersection area. This section introduces a generic concept for inferring possible gaps between objects and compute trajectories to merge into or cross through one of these gaps.

4.1 Merge Behavior

Based on our framework structure presented in Section 3 we first need to estimate a tactical behavior which essentially evaluates and provides the sampling strategy and boundaries for the underlying motion planner task. The *merge* behavior introduced in this section tries to estimate a time window when our automated vehicle has to be at the intersection area. This calculation has to take both the dynamics of the crossing objects and the kinodynamic constraints of our own vehicle into account. Additionally there are some rules imposed by legal, safety and comfort requirements, i.e. defining minimum time gaps other objects These time gap requirements may both depend on the EGO task (crossing or merging) and the target objects type,

i.e. due to the potentially high velocity difference pedestrians may require larger time gaps than cars, that move at similar speeds.

Given our pilot intersection [61] we will now illustrate this for an example situation of merging to a priority lane (Figure 3). Based on the traffic rules encoded in our

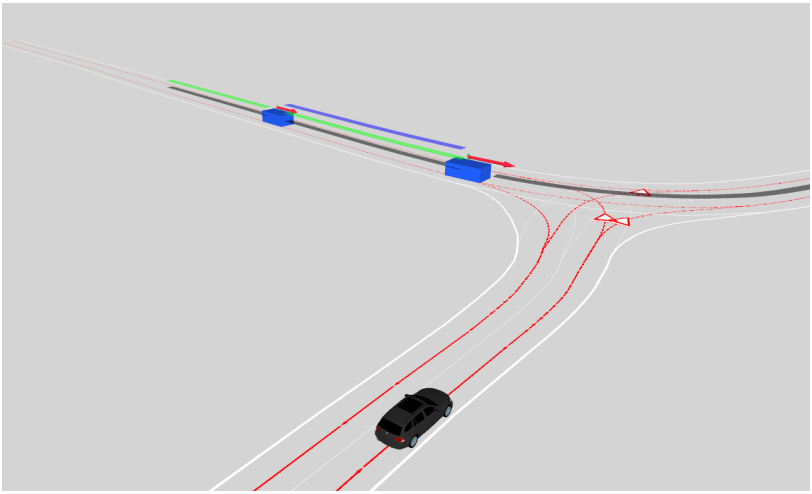


Figure 3: *Merge* behavior situation with all available current gaps (black), a subset of valid gaps (green) and the best gap which is currently selected (blue)

map, we are able to infer priority roads and whether we are going to cross or merge with them. Also we are able to identify all relevant objects on the priority road and estimate the current gaps[1] between them. Based on the objects dynamic state we are able to predict the gaps along the map until reaching the intersection, which allows us to analyze if the gap will be big enough for us to merge into it. For vehicles we do a simple constant velocity prediction along our lane centerlines. For other traffic participants, like pedestrians, a more sophisticated prediction approach may be required to reach proper real-life performance. Essential requirement for such a prediciton would only be that it allows inferring time gaps between objects, one matching algorithm can be found in [110].

A gap is considered big enough if it is able to fit our car and a given desired safety gap both to the front and back, based on legal requirements this safety gap has to be at least 1s each, we are currently testing with gaps in the range 1.5 to 2s. For safety reasons the gaps can also be enlarged based on different uncertainties, from e.g. perception or prediction.

[1] Gaps are considered as spatial intervals along our lane map

After we have enumerated all gaps and excluded all unsafe options, we are able to select a gap. Since our major concern is a combination of time savings and jerk optimization, we first eliminate all gaps that violate our kinodynamic boundaries and choose the first remaining one. Figure 3 illustrates an example: available gaps (black), feasable gaps (green) and our choosen one (blue).

If after all no valid gap could be found, the behavior planner will fall back to stopping at the intersections yield sign position.

4.2 Synchronizing Trajectories

The gaps search from the previous section defines the boundaries for this specific task of our sampling based motion planner introduced in Section 3. The boundaries are: a fixed position along our lane, a target velocity (given by the predicted future lead vehicle) and a time interval, when to be at the point.

The distance and time to the interscetion can be used to calculate a required average speed. Our sampling space is defined by a speed interval around the average speed, hereafter termed as approaching speed, and an acceleration interval contained in the EGO vehicle's dynamic constraints. With this sampling parameter we generate trajectories which accelerate or decelerate smoothly towards an approaching speed, hold this speed as long as needed to accelerate or decelerate to the target speed at the intersection. Once we generated a bundle of trajectories we apply a cost function to all of them regarding e.g. feasibility for our vehicle's dynamic constraints, collision probability and longitudinal jerk for comfortable driving. So based on our cost function evaluation we select the best trajectory with lowest cost which is collision free and feasible to drive. An exemplary trajectory bundle is shown in Figure 4.

5 Evaluation

The infrastructure at out test area in Ulm is currently set up, i.e. there are no real world data available yet. Because of this we instead focus our initial development on a basic simulation. In this section we will first give an overview about this simulation system and its capabilities. Afterwards the evaluated situations and the corresponding data will be introduced. At the end we will show the results of the comparison between driving under visibility constraints and driving with the additional MEC-Server information for the same situations.

5.1 Simulation Setup

Our simulation system heavily relies on our precise map, which contains geometrical informations, like centerlines and boundaries, as well as lane information. Lanes are defined as a combination of a centerline and at least two boundaries (left/right). Additionally there are information about traffic rules, like speed limits and priority rules.

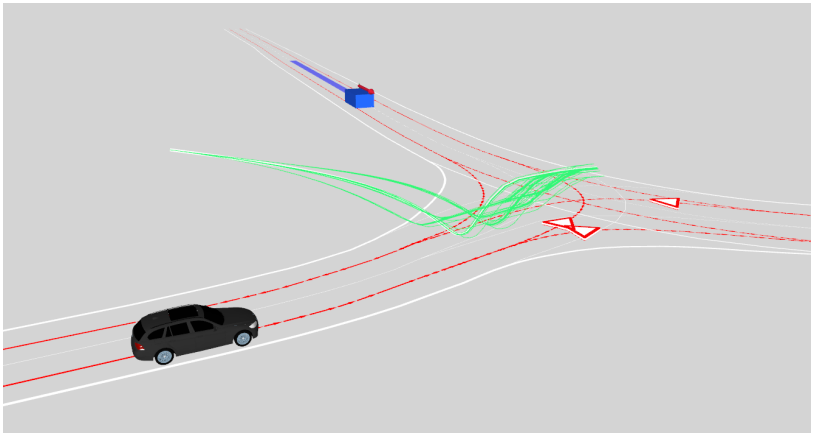


Figure 4: Generated trajectory bundle (green) from our vehicle (black) towards an intersection with one crossing vehicle (blue) with a slightly highlighted selected trajectory (bold white). The z-axis of the trajectories corresponds to speed. Also the intended gap to merge into (blue, behind crossing vehicle) is shown behind the crossing vehicle.

Based on these map information, vehicles can be arbitrarily placed on lanes with both a target velocity and driving intent (can be seen as a hidden target). This placement can either be done by hand, through loading a preset from a recording, or using a automatic traffic generator, which places and removes objects based on a given set of rules. Vehicles automatically accelerate and brake, if required by, e.g. new speed limits or slower lead vehicles. They also have a basic reasoning that allows for collision free yielding, merging and crossing at intersecions. Our EGO vehicle can either be simulated as well or integrated as hardware in the loop.

With this setting we can easily test our planning algorithms before running them in real traffic scenarios.

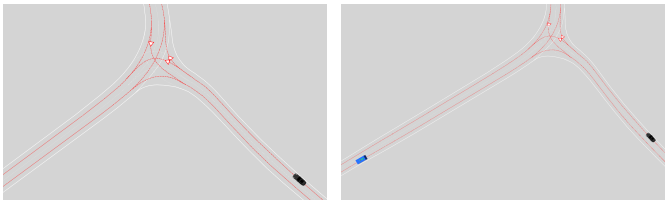
5.2 Scenarios and Metrics

With our simulation setup we want to compare the following two situations.

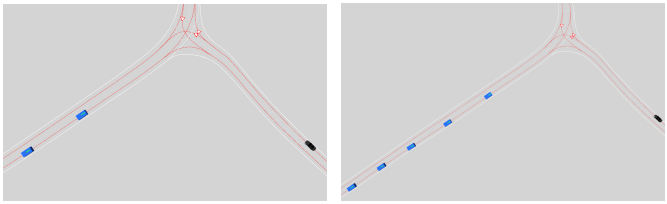
1. *EGO* (driving only with the perception of our vehicle): Our vehicle needs to slow down while approaching the intersection until our sensors see enough to decide if we can merge into the intersection. If there is no possibility at that time due to crossing traffic it will stop at the intersection.
2. *V2I* (driving with additional V2I information): The EGO vehicle now knows

about crossing vehicles and is able to switch into a *merge* behavior and act as described in Section 4.

As evaluation setup we focus on different crossing traffic situations while starting at the same point of the map for the EGO vehicle, the situations are depicted in Figure 5. The situations vary in terms of number and placement of dynamic objects, ranging from no traffic to dense traffic. The starting position of EGO is fixed in all situations.



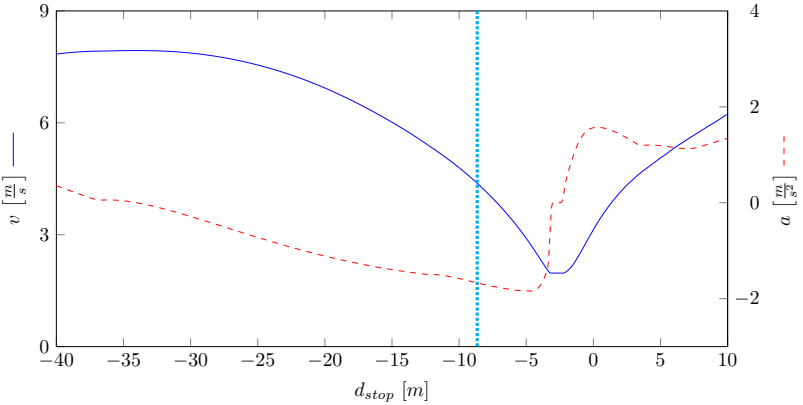
(a) *0*: No crossing traffic. **(b)** *1*: One crossing vehicle.



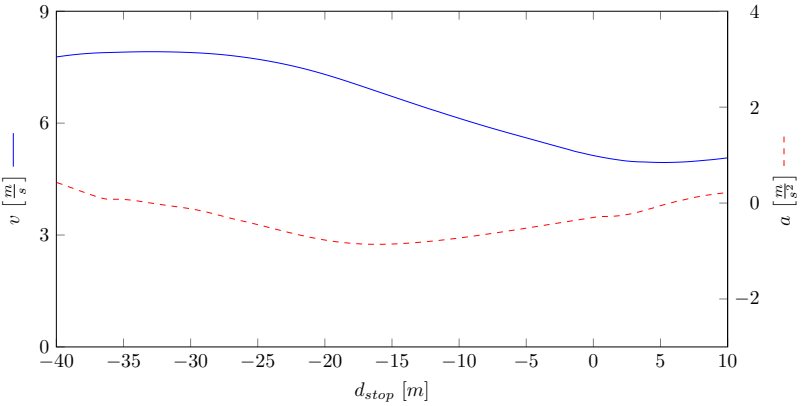
(c) *2*: Two crossing vehicles. **(d)** *3*: Five crossing vehicles.

Figure 5: Evaluated *situations* with different crossing traffic densities. Planning goal is to turn right at the intersection while merging into crossing traffic. The starting position of the EGO vehicle will be maintained throughout all *situations*.

The *situation 0* in Image 5a will be seen as baseline for our comparison. During the scenario we capture the usual velocity v and acceleration a profiles over the driven distance d, which are shown exemplary in Image 6a for the *EGO* and in Image 6b for the *merge* case. The shown distance on the x-axis d_{stop} in both plots is the relative distance to the stop point, i.e. measured from vehicle front. This distance is negative, while approaching the stop point. The vertical line in Image 6a refers to the point where our EGO vehicle sensors are initially able to observe the priority road, i.e. where crossing traffic is initially visible. This is of course only relevant if no V2I information are available.

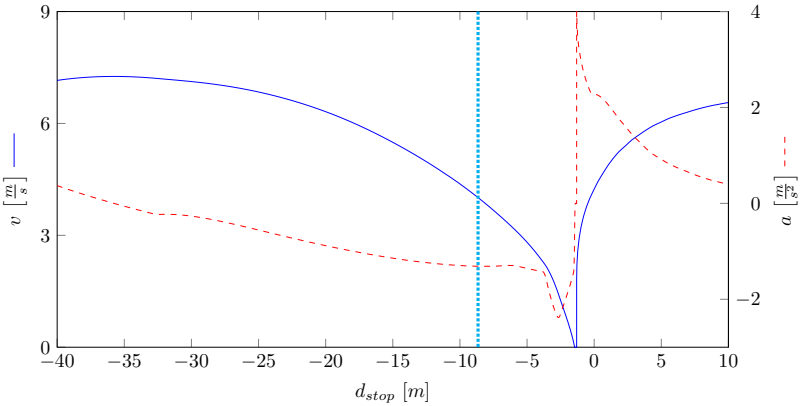


(a) *EGO* case: driving profiles. The vertical line refers to the point where EGO vehicle sensors are able to initially observe the priority road.

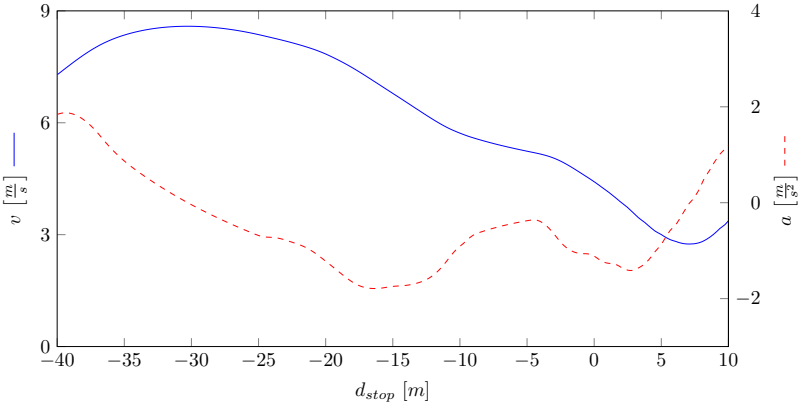


(b) *Merge* case: driving profiles.

Figure 6: Evaluation of *situation 0*: No crossing vehicles, comparable to free driving. *EGO* driving lead to slowing down towards the stop line. *Merge* behavior is crossing the intersection while only slowing down to fulfil lateral acceleration constraints. (a) and (b) show the driving profiles for velocity v, acceleration a and time t over the distance to stop d_{stop}.

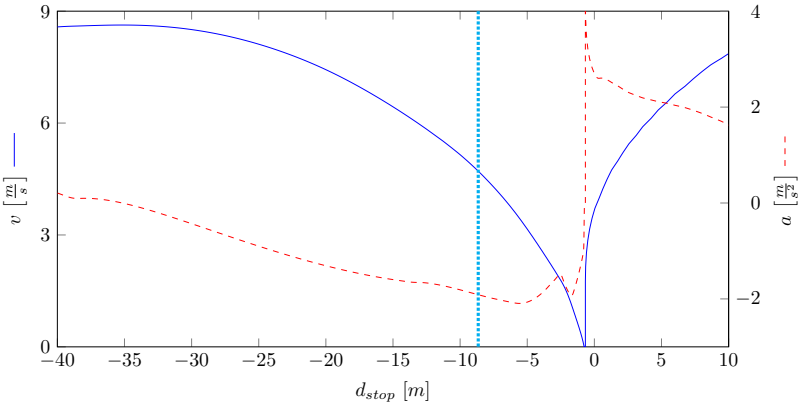


(a) *EGO* case: driving profiles. The vertical line refers to the point where EGO vehicle sensors are able to initially observe the priority road.

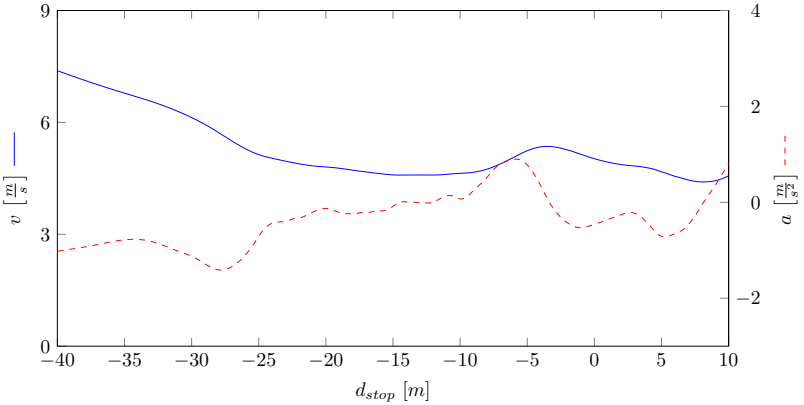


(b) *V2I* case: driving profiles.

Figure 7: Evaluation of *situation 1*: For *EGO* driving the EGO vehicle merged behind the crossing vehicle. For our *merge* behavior the EGO vehicle merged in front of the crossing vehicle. (a) and (b) show the driving profiles for velocity v, acceleration a and time t over the distance to stop d_{stop}.

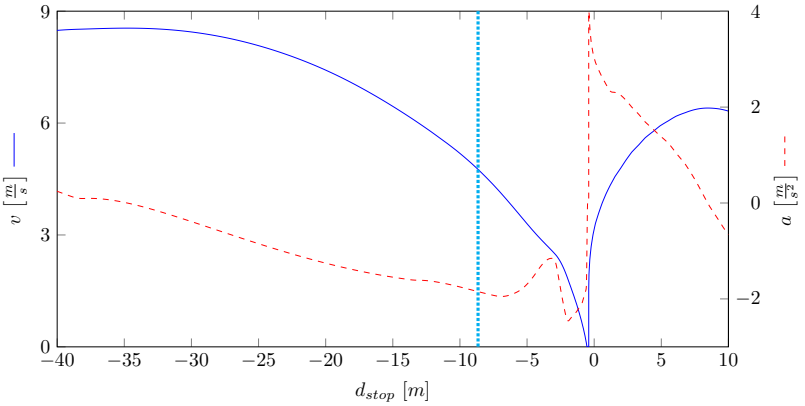


(a) *EGO* case: driving profiles. The vertical line refers to the point where EGO vehicle sensors are able to initially observe the priority road.

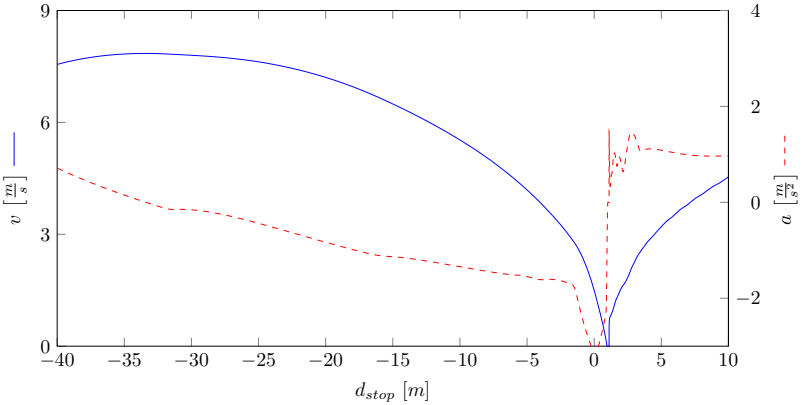


(b) *V2I* case: driving profiles.

Figure 8: Evaluation of *situation 2*: Merging behind the second vehicle. (a) and (b) show the driving profiles for velocity v, acceleration a and time t over the distance to stop d_{stop}.



(a) *EGO* case: driving profiles. The vertical line refers to the point where EGO vehicle sensors are able to initially observe the priority road.



(b) *V2I* case: driving profiles.

Figure 9: Evaluation of *situation 3*: Merging behind the fifth vehicle. (a) and (b) show the driving profiles for velocity v, acceleration a and time t over the distance to stop d_{stop}.

5.3 Results

We now will discuss the results for each scenario in direct comparison between the *EGO* case (driving with vehicle sensors) and the *V2I* case (driving with infrastructure support).

The Figures 7a, 8a and 9a show *EGO* driving in such *situations* leads to a stop at the intersection. Depending on the number of crossing vehicles, the main difference between those *situations* in the driving profile is the time our EGO vehile has to wait at the stop line, as indicated by the time jump around $d_{stop} = 0$.

The *merge* behavior allows the EGO vehicle to slow down while approaching the intersection to let the crossing vehicles pass and then continues merging into the intersection within the time interval of the selected gap. A example for such a profile is shown in Image 8b. Image 7b depict a exceptional case. In this *situation* our *merge* behavior leads to approaching the intersection clearly faster to merge in front of the crossing vehicle. This leads to a notable saving of driving time in such *situations*, where the space in front of crossing traffic is large enough. Another special *situation* is shown in Image 9b. Because of the large amount of vehicles and the resulting high traffic density, our vehicles is blocked from merging in between any of the crossing vehicles. This shows that even when the *merge* behavior is unable to merge into a gap, the overall system behavior defaults back to the baseline performance (Image 9a), i.e. the new *merge* behavior never performs worse than the *EGO* baseline.

An important measure regarding passenger comfort is the jerk. Table 1 shows a reduction of both the maximal and minimal jerk for all situations. The major reason for the jerk reduction is that we don't have to come to a full stop anymore. Please note, that in general the deceleration jerk j^- is expected to have a much higher negative impact. Because of this j^- is subject to much higher limitations during trajectory planning, i.e. resulting in highly comfortable brake trajectories. The acceleration jerk j^+ on the other hand has, by design, much less constraints

Table 1: Maximal (acceleration) and minimal (deceleration) jerk, and the overall duration for all evaluated scenarios.

Situation	**Max j^+** $\left[\frac{m}{s^3}\right]$	**Min j^-** $\left[\frac{m}{s^3}\right]$	**Duration** $[s]$
0-*EGO*	12.69	−0.81	9.6
0-*V2I*	0.44	−0.80	7.6
1-*EGO*	37.42	−3.39	12.3
1-*V2I*	2.36	−2.44	8.4
2-*EGO*	37.42	−3.47	11.3
2-*V2I*	2.39	−2.76	9.3
3-*EGO*	37.42	−4.48	17.7
3-*V2I*	21.30	−4.35	17.8

to allow faster merging while accelerating from standstill. Table 1 also shows that our system can reduce the time required to pass the intersection area by up to 4s. These 4s can be achieved in *situation 1*, where our vehicle was able to merge in front of all crossing traffic.

6 Conclusion

In this paper we provided an overview on our current work regarding analyzing the potential benefit of infrastructure supported urban automated driving. The infrastructure for this project is provided by the German publicly funded project MEC-View. We showed details on how a planning framework in an automated vehicle can effectively use an extended field of view on V2X basis. We introduced a concept for synchronizing our trajectory with arbitrary dynamic objects, targeting safe, time saving and jerk minimal EGO trajectories.

We evaluated our system based on multiple scenario simulations, that varied between no traffic and dense traffic on the priority road. Using a base scenario (without infrastructure information) for comparison, we found: In the majority of the scenes the automated vehicle could both significantly improve the passenger comfort, due to massive jerk reduction, and reduce the time required to cross the intersection area by up to 4s. Both these benefits arise mainly from the elimination of the necessity to (fully) stop at the intersection. The last scene showed that in very dense traffic (containing multiple vehicles without merging gaps between them) the system falls back to the base performance. To solve these cases either additional information, like turning intentions (that open up new gaps), or some kind of cooperation between the vehicles may be required to enhance these cases. In our future work we will look further into handling these dense traffic scenarios and provide an analysis of the real world driving performance, as soon as the infrastructure is fully set up.

Bibliography

[1] G. Agamennoni, J. I. Nieto, and E. M. Nebot. Estimation of multivehicle dynamics by considering contextual information. *IEEE Transactions on Robotics*, 28(4):855–870, 2012.

[2] A. Alahi, K. Goel, V. Ramanathan, A. Robicquet, L. Fei-Fei, and S. Savarese. Social lstm: Human trajectory prediction in crowded spaces. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 961–971, 2016.

[3] 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.

[4] F. Bartoli, G. Lisanti, L. Ballan, and A. Del Bimbo. Context-aware trajectory prediction. In *24th International Conference on Pattern Recognition (ICPR)*, pages 1941–1946, 2018.

[5] F. Bastien, P. Lamblin, R. Pascanu, J. Bergstra, I. J. Goodfellow, A. Bergeron, N. Bouchard, and Y. Bengio. Theano: new features and speed improvements. Deep Learning and Unsupervised Feature Learning NIPS 2012 Workshop, 2012.

[6] A. Bera, S. Kim, T. Randhavane, S. Pratapa, and D. Manocha. Glmp-realtime pedestrian path prediction using global and local movement patterns. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 5528–5535, 2016.

[7] 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.

[8] A. Bera, T. Randhavane, and D. Manocha. Aggressive, tense or shy? identifying personality traits from crowd videos. In *Proc. of the 26th International Joint Conference on Artificial Intelligence (IJCAI)*, pages 112–118, 2017.

[9] M. Bertozzi, L. Castangia, S. Cattani, A. Prioletti, and P. Versari. 360° detection and tracking algorithm of both pedestrian and vehicle using fisheye images. In *Proc. of the IEEE Intelligent Vehicles Symposium (IV)*, 2015.

[10] N. Bisagno, B. Zhang, and N. Conci. Group lstm: Group trajectory prediction in crowded scenarios. In *European Conference on Computer Vision Workshops*, pages 213–225. Springer, 2019.

[11] 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.

[12] S. Brechtel, T. Gindele, and R. Dillmann. Probabilistic decision-making under uncertainty for autonomous driving using continuous pomdps. In *17th International IEEE Conference on Intelligent Transportation Systems (ITSC)*, 2014.

[13] L. Breiman. Random forests. *Machine Learning*, 45:5–32, 2001.

[14] N. Brouwer, H. Kloeden, and C. Stiller. Comparison and evaluation of pedestrian motion models for vehicle safety systems. In *IEEE 19th International Conference on Intelligent Transportation Systems (ITSC)*, pages 2207–2212, 2016.

[15] C. J. Burges. A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery*, 2:121–167, 1998.

[16] H.-J. Chang and G.-T. Park. A study on traffic signal control at signalized intersections in vehicular ad hoc networks. *Ad Hoc Networks*, 11(7):2115–2124, 2013. Theory, Algorithms and Applications of Wireless Networked Robotics Recent Advances in Vehicular Communications and Networking.

[17] L. Chen and C. Englund. Cooperative intersection management: A survey. *IEEE Transactions on Intelligent Transportation Systems*, 17(2):570–586, 2016.

[18] Y. F. Chen, M. Liu, M. Everett, and J. P. How. Decentralized non-communicating multiagent collision avoidance with deep reinforcement learning. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 285–292, 2017.

[19] S. Chung and H. Huang. Incremental learning of human social behaviors with feature-based spatial effects. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2417–2422, 2012.

[20] M. Clamann, M. Aubert, and M. Cummings. Evaluation of vehicle-to-pedestrian communication displays for autonomous vehicles. In *Tech. Rep.*, 2017.

[21] Daimler. Overview: Mercedes-Benz F 015 luxury in motion. https://media.daimler.com/marsMediaSite/en/instance/ko.xhtml?oid=9904624, 2017. Accessed: 2019-09-28.

[22] S. Dieleman, J. Schlüter, C. Raffel, E. Olson, S. K. Sønderby, D. Nouri, D. Maturana, M. Thoma, E. Battenberg, J. Kelly, J. D. Fauw, M. Heilman, diogo149, B. McFee, H. Weideman, takacsg84, peterderivaz, Jon, instagibbs, D. K. Rasul, CongLiu, Britefury, and J. Degrave. Lasagne: First release., 2015. URL http://dx.doi.org/10.5281/zenodo.27878.

[23] K. Dresner and P. Stone. A multiagent approach to autonomous intersection management. *Journal of Artificial Intelligence Research*, 2:1593–1596, 2008.

[24] 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.

[25] Euro NCAP. AEB pedestrian. https://www.euroncap.com/en/vehicle-safety/the-ratings-explained/vulnerable-road-user-vru-protection/aeb-pedestrian/. Accessed: 2019-09-28.

[26] Z. Fang, D. Vázquez, and A. M. López. On-board detection of pedestrian intentions. In *Sensors*, volume 17, 2017.

[27] A. F. Foka and P. E. Trahanias. Probabilistic autonomous robot navigation in dynamic environments with human motion prediction. *International Journal of Social Robotics*, 2(1):79–94, 2010.

[28] GeneSys. ADMA family GPS/ Inertial System Automotive/ Railway. https://www.genesys-offenburg.de/en/products/adma-family-gpsinertial-system-automotiverailway/. Accessed: 2019-09-28.

[29] 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.

[30] M. Goldhammer, M. Gerhard, S. Zernetsch, K. Doll, and U. Brunsmann. Early prediction of a pedestrian's trajectory at intersections. In *IEEE Int. Conf. on Intelligent Transportation Systems (ITSC)*, 2013.

[31] D. González, J. Pérez, V. Milanés, and F. Nashashibi. A review of motion planning techniques for automated vehicles. *IEEE Transactions on Intelligent Transportation Systems*, 17(4):1135–1145, 2016.

[32] A. Gorrini, G. Vizzari, and S. Bandini. Age and group-driven pedestrian behaviour: from observations to simulations. *Collective Dynamics*, 1:1–16, 2016.

[33] E. I. S. Group. Gs mec 001: Multi-access edge computing (mec); terminology. Technical report, European Telecommunications Standards Institute (ETSI), 2018.

[34] H.-J. Günther, R. Riebl, L. Wolf, and C. Facchi. Collective perception and decentralized congestion control in vehicular ad-hoc networks. In *Proc. of the Vehicular Networking Conference*, 2016.

[35] A. Gupta, J. Johnson, L. Fei-Fei, S. Savarese, and A. Alahi. Social gan: Socially acceptable trajectories with generative adversarial networks. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2255–2264, 2018.

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

[37] K. Hao. The Three Challenges Keeping Cars from Being Fully Autonomous. *MIT Technology Review*, 2019.

[38] T. J. Hastie, R. J. Tibshirani, and J. H. Friedman. *The elements of statistical learning : data mining, inference, and prediction.* Springer series in statistics. Springer, 2009.

[39] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural Comput.*, 9(8):1735–1780, 1997.

[40] E. Käfer, C. Hermes, C. Wöhler, H. Ritter, and F. Kummert. Recognition of situation classes at road intersections. In *IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2010.

[41] 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.

[42] 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)*, pages 2543–2549, 2016.

[43] C. G. Keller and D. M. Gavrila. Will the pedestrian cross? a study on pedestrian path prediction. *IEEE Transactions on Intelligent Transportation Systems*, 15(2):494–506, 2014.

[44] Kersten Heineke and Philipp Kampshoff and Armen Mkrtchyan and Emily Shao. Self-Driving Car Technology: When Will the Robots Hit the Road? *McKinsey & Company*, 2017.

[45] S. Kim, S. J. Guy, W. Liu, D. Wilkie, R. W. Lau, M. C. Lin, and D. Manocha. Brvo: Predicting pedestrian trajectories using velocity-space reasoning. *The International Journal of Robotics Research*, 34(2):201–217, 2015.

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

[47] S. Köhler, M. Goldhammer, S. Bauer, K. Doll, U. Brunsmann, and K. Dietmayer. Early detection of the pedestrian's intention to cross the street. In *IEEE Int. Conf. on Intelligent Transportation Systems (ITSC)*, 2012.

[48] 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 Intelligent Transportation Systems Magazine*, 2013.

[49] S. Köhler, M. Goldhammer, K. Zindler, K. Doll, and K. Dietmeyer. Stereo-vision-based pedestrian's intention detection in a moving vehicle. In *IEEE 18th International Conference on Intelligent Transportation Systems (ITSC)*, pages 2317–2322, 2015.

[50] J. Kooij, N. Schneider, F. Flohr, and D. Gavrila. Context-based pedestrian path prediction. In *Proc. of the European Conference on Computer Vision (ECCV)*, pages 618–633. Springer, 2014.

[51] 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*, 38:322–334, 2016.

[52] J. F. P. Kooij, F. Flohr, E. A. I. Pool, and D. M. Gavrila. Context-based path prediction for targets with switching dynamics. *International Journal of Computer Vision*, 127(3):239–262, 2019.

[53] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems 25: 26th Annual Conference on Neural Information Processing Systems 2012. Proceedings of a meeting held December 3-6, 2012, Lake Tahoe, Nevada, United States.*, pages 1106–1114, 2012.

[54] N. Lang, M. Rüßmann, A. Mei-Pochtler, T. Dauner, S. Komiya, X. Mosquet, and X. Doubara. Self-Driving Vehicles, Robo-Taxis, and the Urban Mobility Revolution. *Boston Consulting Group*, 2016.

[55] S. Lefevre, C. Laugier, and J. Ibanez-Guzman. Evaluating risk at road intersections by detecting conflicting intentions. In *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, 2012.

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

[57] L. Li and F. Wang. Cooperative driving at blind crossings using intervehicle communication. *IEEE Transactions on Vehicular Technology*, 55(6):1712–1724, 2006.

[58] X. R. Li and V. P. Jilkov. Survey of maneuvering target tracking. part v: Multiple-model methods. *IEEE Transactions on Aerospace and Electronic Systems*, 41(4):1255–1321, 2005.

[59] M. Luber, L. Spinello, J. Silva, and K. O. Arras. Socially-aware robot navigation: A learning approach. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 902–907, 2012.

[60] A. Maurya and P. Bokare. Study of deceleration behaviour of different vehicle types. *International Journal for Traffic and Transport Engineering*, 2:253–270, 2012.

[61] MEC-View. Mobile edge computing based object detection for automated driving. www.mec-view.de. Accessed: 2019-01-15.

[62] N. Meinshausen. Quantile regression forests. *Journal of Machine Learning Research*, 7:983–999, 2006.

[63] J. Mingle. A New Lens on Suburbia. Center for Advanced Urbanism Conference Explores the Suburbs? Sustainable Future. *MIT News*, 2016.

[64] Mitsubishi. Mitsubishi electric introduces road-illuminating directional indicators. https://www.mitsubishielectric.com/news/2015/1023.html?cid=rss. Accessed: 2019-09-28.

[65] A. Møgelmose, M. M. Trivedi, and T. B. Moeslund. Trajectory analysis and prediction for improved pedestrian safety: Integrated framework and evaluations. In *IEEE Intelligent Vehicles Symposium (IV)*, pages 330–335, 2015.

[66] K. P. Murphy. *Dynamic Bayesian Networks: Representation, Inference and Learning*. PhD thesis, University of California, Berkeley, USA, 2002.

[67] V. Nair and G. E. Hinton. Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, pages 807–814, 2010.

[68] M. Naumann, M. Lauer, and C. Stiller. Generating comfortable, safe and comprehensible trajectories for automated vehicles in mixed traffic. In *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, 2018.

[69] ö. Ş. Taş and C. Stiller. Limited visibility and uncertainty aware motion planning for automated driving. In *2018 IEEE Intelligent Vehicles Symposium (IV)*, 2018.

[70] ö. Ş. Taş, F. Kuhnt, J. M. Zöllner, and C. Stiller. Functional system architectures towards fully automated driving. In *2016 IEEE Intelligent Vehicles Symposium (IV)*, 2016.

[71] M. Pfeiffer, U. Schwesinger, H. Sommer, E. Galceran, and R. Siegwart. Predicting actions to act predictably: Cooperative partial motion planning with maximum entropy models. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2096–2101, 2016.

[72] J. Ploeg, E. Semsar-Kazerooni, A. I. M. Medina, J. F. C. M. de Jongh, J. van de Sluis, A. Voronov, C. Englund, R. J. Bril, H. Salunkhe, Á. Arrúe, A. Ruano, L. García-Sol, E. van Nunen, and N. van de Wouw. Cooperative automated maneuvering at the 2016 grand cooperative driving challenge. *IEEE Transactions on Intelligent Transportation Systems*, 19(4):1213–1226, 2018.

[73] X. Qian, J. Gregoire, F. Moutarde, and A. D. L. Fortelle. Priority-based coordination of autonomous and legacy vehicles at intersection. In *17th International IEEE Conference on Intelligent Transportation Systems (ITSC)*, 2014.

[74] R. Quintero, J. Almeida, D. F. Llorca, and M. Á. Sotelo. Pedestrian path prediction using body language traits. In *IEEE Intelligent Vehicles Symposium Proceedings (IV)*, pages 317–323, 2014.

[75] 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.

[76] R. Quintero Mínguez, I. Parra Alonso, D. Fernández-Llorca, and M. Á. Sotelo. Pedestrian path, pose, and intention prediction through gaussian process dynamical models and pedestrian activity recognition. *IEEE Transactions on Intelligent Transportation Systems*, 20(5):1803–1814, 2019.

[77] A. Rasouli and J. K. Tsotsos. Autonomous vehicles that interact with pedestrians: A survey of theory and practice. *IEEE Transactions on Intelligent Transportation Systems*, pages 1–19, 2019.

[78] E. Rehder and H. Kloeden. Goal-directed pedestrian prediction. In *IEEE International Conference on Computer Vision Workshop (ICCVW)*, pages 139–147, 2015.

[79] Robert Bosch GmbH. Stereo Video Camera. https://www.bosch-mobility-solutions.com/en/products-and-services/passenger-cars-and-light-commercial-vehicles/driver-assistance-systems/lane-departure-warning/stereo-video-camera/. Accessed: 2019-09-28.

[80] A. Robicquet, A. Sadeghian, A. Alahi, and S. Savarese. Learning social etiquette: Human trajectory understanding in crowded scenes. In *Proc. of the European Conference on Computer Vision (ECCV)*, pages 549–565. Springer, 2016.

[81] J. Rohde, B. Völz, H. Mielenz, and J. M. Zöllner. Precise vehicle localization in dense urban environments. In *2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC)*, pages 853–858, 2016.

[82] M. Roth, F. Flohr, and D. M. Gavrila. Driver and pedestrian awareness-based collision risk analysis. In *IEEE Intelligent Vehicles Symposium (IV)*, pages 454–459, 2016.

[83] N. Roy, P. Newman, and S. Srinivasa. Modeling and prediction of pedestrian behavior based on the sub-goal concept. In *Robotics: Science and Systems VIII*, 2013.

[84] A. Rudenko, L. Palmieri, L. Palmieri, M. Herman, K. M. Kitani, D. M. Gavrila, and K. O. Arras. Human motion trajectory prediction: A survey. *arXiv:1905.06113*, 2019. Submitted to the International Journal of Robotics Research.

[85] K. Saleh, M. Hossny, and S. Nahavandi. Intent prediction of pedestrians via motion trajectories using stacked recurrent neural networks. *IEEE Transactions on Intelligent Vehicles*, 3(4):414–424, 2018.

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

[87] F. Schneemann and P. Heinemann. Context-based detection of pedestrian crossing intention for autonomous driving in urban environments. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2243–2248, 2016.

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

[89] C. Schöller, V. Aravantinos, F. Lay, and A. Knoll. The simpler the better: Constant velocity for pedestrian motion prediction. *arXiv:1903.07933*, 2019.

[90] 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.

[91] A. T. Schulz and R. Stiefelhagen. A controlled interactive multiple model filter for combined pedestrian intention recognition and path prediction. In *IEEE 18th International Conference on Intelligent Transportation Systems (ITSC)*, pages 173–178, 2015.

[92] F. Seeliger, G. Weidl, D. Petrich, F. Naujoks, G. Breuel, A. Neukum, and K. Dietmayer. Advisory warnings based on cooperative perception. In *Proc. of the IEEE Intelligent Vehicles Symposium (IV)*, 2014.

[93] X. Shi, X. Shao, Z. Gua, G. Wu, H. Zhang, and R. Shibasaki. Pedestrian trajectory prediction in extremely crowded scenarios. In *Sensors*, volume 19, 2019.

[94] Shu-Yun Chung and Han-Pang Huang. A mobile robot that understands pedestrian spatial behaviors. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 5861–5866, 2010.

[95] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15:1929–1958, 2014.

[96] T. Streubel and K. H. Hoffmann. Prediction of driver intended path at intersections. In *Proc. of the IEEE Intelligent Vehicles Symposium (IV)*, 2014.

[97] H. Su, J. Zhu, Y. Dong, and B. Zhang. Forecast the plausible paths in crowd scenes. In *Proc. of the 26th International Joint Conference on Artificial Intelligence (IJCAI)*, pages 2772–2778, 2017.

[98] Y. Sugiyama, M. Fukui, M. Kikuchi, K. Hasebe, A. Nakayama, K. Nishinari, S. ichi Tadaki, and S. Yukawa. Traffic jams without bottlenecks? experimental evidence for the physical mechanism of the formation of a jam. *New J. Phys.*, 10(3), 2008. 033001.

[99] L. Sun, Z. Yan, S. M. Mellado, M. Hanheide, and T. Duckett. 3dof pedestrian trajectory prediction learned from long-term autonomous mobile robot deployment data. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 1–7, 2018.

[100] Z. Taylor, J. Nieto, and D. Johnson. Multi-modal sensor calibration using a gradient orientation measure. *Journal of Field Robotics*, 32:675–695, 2015.

[101] 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.

[102] The Economist. Reinventing Wheels: A Special Report on Autonomous Driving. *The Economist Newspaper*, 2018.

[103] P. Trautman and A. Krause. Unfreezing the robot: Navigation in dense, interacting crowds. In *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 797–803, 2010.

[104] P. Trautman, J. Ma, R. M. Murray, and A. Krause. Robot navigation in dense human crowds: the case for cooperation. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 2153–2160, 2013.

[105] Velodyne Lidar. HDL-64E. https://velodynelidar.com/hdl-64e.html. Accessed: 2019-09-28.

[106] A. Vemula, K. Muelling, and J. Oh. Modeling cooperative navigation in dense human crowds. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 1685–1692, 2017.

[107] 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.

[108] B. Völz, K. Behrendt, H. Mielenz, I. Gilitschenski, R. Siegwart, and J. Nieto. A data-driven approach for pedestrian intention estimation. In *2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC)*, pages 2607–2612, 2016.

[109] 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.

[110] B. Völz, H. Mielenz, I. Gilitschenski, R. Siegwart, and J. Nieto. Inferring pedestrian motions at urban crosswalks. *IEEE Transactions on Intelligent Transportation Systems*, 20(2):544–555, 2019.

[111] B. Völz, A. Stamm, M. Maier, R.-W. Henn, R. Siegwart, and J. Nieto. Towards infrastructure-supported planning for urban automated driving. In *Robotics: Science and Systems (RSS); Workshop on Scene and Situation Understanding for Autonomous Driving*, 2019. URL https://sites.google.com/view/uad2019/.

[112] M. Won, T. Park, and S. H. Son. Toward mitigating phantom jam using vehicle-to-vehicle communication. *IEEE Transactions on Intelligent Transportation Systems*, 18(5):1313–1324, 2017.

[113] World Health Organization. Global Status Report on Road Safety. *World Health Organization*, 2015.

[114] World Health Organization. Ambient Air Pollution: A Global Assessment of Exposure and Burden of Disease. *World Health Organization*, 2016.

[115] R. Wunderlich, C. Liu, I. Elhanany, and T. U. II. A novel signal-scheduling algorithm with quality-of-service provisioning for an isolated intersection. *IEEE Transactions on Intelligent Transportation Systems*, 9(3):536–547, 2008.

[116] H. Xue, D. Q. Huynh, and M. Reynolds. Bi-prediction: Pedestrian trajectory prediction based on bidirectional lstm classification. In *International Conference on Digital Image Computing: Techniques and Applications (DICTA)*, pages 1–8, 2017.

[117] H. Xue, D. Huynh, and M. Reynolds. Location-velocity attention for pedestrian trajectory prediction. In *IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 2038–2047, 2019.

[118] K. Yamaguchi, A. C. Berg, L. E. Ortiz, and T. L. Berg. Who are you with and where are you going? In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1345–1352, 2011.

[119] M. D. Zeiler. ADADELTA: an adaptive learning rate method. *arXiv:1212.5701*, 2012.

[120] J. Zhang and B. Roessler. Situation analysis and adaptive risk assessment for intersection safety systems in advanced assisted driving. In *Proc. of the 21th Fachgespräch Autonome Mobile Systeme*, pages 249–258, 2009.