Route and Stopping Intent Prediction at Intersections From Car Fleet Data

Florian Gross, Justus Jordan, Felix Weninger, Felix Klanner, *Member, IEEE*, and Björn Schuller, *Senior Member, IEEE*

***Abstract*—In this paper, an approach is presented to predict the route and stopping intent of human-driven vehicles at urban intersections using a selection of distinctive features observed on the vehicle state (position, heading, acceleration, velocity). For potential future advanced driver assistance systems, this can facilitate the situation analysis and risk assessment at road intersections, helping to improve the protection of vulnerable road users. After extracting recorded driving data for nine intersections (featuring over 50 000 crossings) from a database, they are assigned to possible routes and transformed from a time-based representation to a distance-based one. Using random decision forests, the route intent can be predicted with a mean unweighted average recall (UAR) of 0.76 at 30 m before the relevant intersection center, the stopping intent prediction scores a mean UAR of 0.78.**

***Index Terms*—Advanced driver assistance systems, car fleet data, driver state and intent recognition, driving data, machine learning.**

I. Introduction

While the number of registered vehicles in Germany is still on the rise [1], the number of road traffic accidents involving personal injuries is steadily decreasing [2]. One factor contributing to this trend is the evolvement of advanced driver assistance systems' (ADAS) capabilities. To further improve road safety, multiple projects encouraged the research in the field of vehicle-to-x communication (combining vehicle-to-vehicle and vehicle-to-infrastructure). In Germany, recent projects were sim^{TD} [3] and Ko-FAS [4], both initiated by the German automotive branch and funded by the German Ministry of Economics and Technology. Still, the majority of accidents with personal injuries happen in an urban environment [5], [6]. With a rising number of road users and countless possibilities of distraction in a complex environment, there is a need for ADAS in urban areas.

While outside of cities (on highways or freeways), the majority of involved persons are the respective vehicle's passengers, inside of cities also vulnerable road users (VRU), such as pedestrians or cyclists, are likely to be involved in a road traffic accident. Over the last few decades, the protection of vehicle passengers has been improved steadily. However, although there were efforts to improve VRU safety by passive as well as active safety measures, VRU still suffer serious injuries from collisions with vehicles. Hence the most promising way to reduce injuries at accidents with VRU is to prevent them altogether.

One potential use case here is the protection of cyclists when performing a right-turn (see figure 1): Mostly, cyclists travel parallel to cars on separate lanes. Before turning right in countries with right-hand driving, one has to pay special attention to cyclists going straight ahead. A similar situation arises for pedestrians intending to cross the street a driver wants to turn into. In this setting, a potential ADAS could warn drivers if, and *only* if the driver intends to turn right without stopping first. The general *driver intent* in this context comprises the choice of the turn direction (e.g. left, right, straight ahead) as well as the intent to stop at a given point: Mostly stoplines defined through traffic rules, or pedestrians crossing.

Related Work: In the context of robotics, intent prediction is divided into two types of approaches: Descriptive and generative [7]. Descriptive methods aim to characterize patterns through the extraction of low-level features [7]. When labeling these features with the underlying goals, belief and intentions, the "action-effects associations" method allows for intention interpretation. In contrast, generative methods [8] model causes that can produce the observed data through latent (hidden) variables. Another possibility is the mirroring of intentional agents (refer to the "like me" framework describing infant imitation [9]). A general approach to driver intent recognition based on the psychological "Perception-Action" model is described and evaluated in [10].

A more specific case, the prediction of the turn intent is covered in [11]–[18]. By sampling signals from a vehicle's CAN bus and utilizing a Hidden Markov Model (HMM), Sathyanarayana *et al.* [11] manage to predict all right turns, however they did not provide data on how timely the prediction took place. Berndt and Dietmayer employ a similar approach in [12], [13]: Multiple features, such as velocity and steering wheel angle are fed into a left-to-right HMM. It is decided to leave out the turn signal as feature, since it is believed that there is a risk of "overweighing them in well-behaved situations and then failing in critical driving situations" [13]. After evaluating the sensitivity of the individual features, it is found that the steering wheel angle proved most useful for turn recognition. However, when mostly using the steering wheel angle as feature, it can be

F. Gross and J. Jordan are with the BMW Group, Munich 80788, Germany (e-mail: florian.lh.gross@bmw.de; justus.jordan@bmw.de).

F. Weninger was with the Institute for Human-Machine Communication, Technische Universität München, 80333 Munich, Germany. He is now with Nuance Communications, 89077 Ulm, Germany (e-mail: felix@weninger.de).

F. Klanner is with the School of Computer Science and Engineering, Nanyang Technological University, Singapore 0639798 (e-mail: klanner@ntu.edu.sg).

B. Schuller is with the Chair of Complex and Intelligent Systems, University of Passau, Passau 94032, Germany, and also with the Department of Computing, Imperial College London, London SW7 2AZ, U.K. (e-mail: schuller@ieee.org).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TIV.2016.2617625

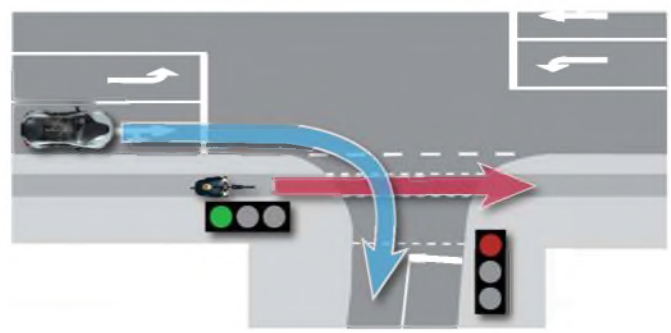

Fig. 1. Right turn with a conflicting cyclist.

suspected that the turns are only recognized after they already started, i.e. a heading change is noticeable.

Using high-precision localization, a digital lane-level map and a Bayesian approach, [14], [15] predict the route choice by only using the entry lane and turn signal. For vehicles showing consistent behavior (i.e. correct lane for the intended turn + turn signal used correctly), the authors report no misclassifications. As soon as vehicles show inconsistent behavior though, the performance decreases significantly (5% correct classifications at the stopline). This approach seems best suited for complex intersections with multiple lanes and all drivers using the turn signals correctly. In 2013, Liebner *et al.* introduced a method [17] for combined turn and stop prediction using an explicit model for a vehicle's velocity profile [18] (the Intelligent Driver Model (IDM) [19]). Similarly, high-precision localization together with a digital lane-level map enables a Bayesian approach. Being the first approach to explicitly model preceding vehicles, the influence of preceding vehicles on the classification performance is evaluated: As for vehicle traces with no preceding vehicles, at a false positive rate (FPR) of 5% a true positive rate (TPR) of 95% is achieved. In the presence of preceding vehicles though, this TPR decreases to 55%. Hence, this approach is able to express uncertainty in the presence of a preceding vehicles, enabling the same FPR for this case. In [20], the performance of a velocity-based stop prediction is evaluated using extensive infrastructure (four radar units, four cameras) at a signaled intersection. A Support Vector Machine (SVM) classifies the stopping intent for all vehicles passing the intersection, and issues warnings when vehicles are predicted to violate a red light. In 2012, Aoude *et al.* introduce an HMM and Bayesian Filtering for the SVM, trained on a naturalistic car dataset for a similar infrastructure [21]. Similarly, [22] surveys the perception-reaction time at signaled intersections for the onset of a yellow phase, whereas [23] focuses on the factors contributing to a stop/go decision and presents a classification tree based method for classifying the same.

[24], [25] pursue a different approach: The two-dimensional trajectory is predicted by either comparing the current trajectory to previously recorded ones [24] or using a Gaussian Mixture Model [25]. In [26], the behavior of human beings in dangerous situations is analyzed and a detection approach using brake pedal force and speech features is presented. Most approaches use the vehicle dynamics such as velocity, acceleration and heading and the change of these as primarily modeled input. However, recent research efforts such as [27], [28] investigate the connectedness of a driver's gaze direction to the driver's intention. Doshi *et al.* infer the driver's gaze patterns and the vehicle's surroundings by inside-mounted and outside-mounted cameras. It is further investigated, in which direction the drivers gaze for what reason. Although it is focused on lane change maneuvers, characteristic gaze patterns for left or right turns are discovered. Jain *et al.* demonstrate a promising approach to fuse multi-modal sensor cues, including the driver's gaze direction [28]. Still, the use of a driver-facing camera limits the use of this approach to the ego vehicle only, and one cannot predict the driver activity for e.g. crossing or preceding vehicles.

All in all, although there are already numerous studies concerning *driver intent recognition*, most of them focus on lane-change detection on freeways, the recognition of maneuvers after they already started (i.e. for a significant heading change) or are only feasible in highly constrained and demanding environments, such as the ones using high-precision localization. This paper introduces a two-fold approach for driver intent prediction based on a naturalistic car fleet dataset retrieved from production vehicles. The performance of classification via Random Decision Forests (RDF) as well as a Support Vector Machine (SVM) is investigated and it is demonstrated that in most cases a correct prediction can be achieved more than three seconds before reaching the navigation link reference node. The remainder of this paper is structured as follows: Section II introduces the dataset and the necessary steps before training of the machine learning algorithms described in Section III. Section IV shows experimental prediction results, whereas Section V concludes this paper.

II. Data Collection and Pre-Processing

As this paper utilizes naturalistic driving data, recorded from different drivers with different car models, first the process of data collection and pre-processing is described. The process of recording was already dealt with in the research project, more details can be found in [3]. For sim^{TD}, more than 100 vehicles were equipped with off-the-shelf hardware and data transmission units, sending data from the vehicle's CAN bus to a backend every second. They were then instructed to travel specific routes in the area of Frankfurt, continuously sending driving data to a central server. In addition to applying this data to new ADAS, it was used to test the performance of potential vehicle to vehicle and vehicle to infrastructure communication systems. A part of this paper's contribution is to prepare the raw driving data for usage with machine learning algorithms. As it can be seen in figure 2, the four major pre-processing steps are extracting trace points from the database, route association, transformation to a local reference system and the selection and extraction of applicable features.

A. Data Source Description

In order to generate a better understanding of the underlying sim^{TD} dataset, several terms have to be defined and explained. The smallest unit of the underlying dataset is a *trace point*.

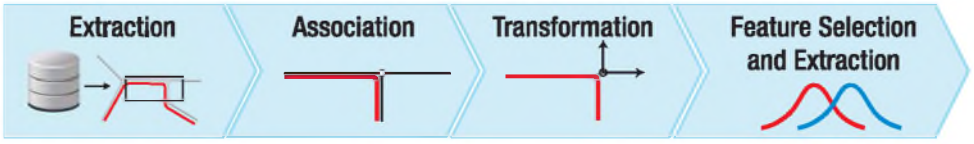

Fig. 2. Steps for data collection and pre-processing.

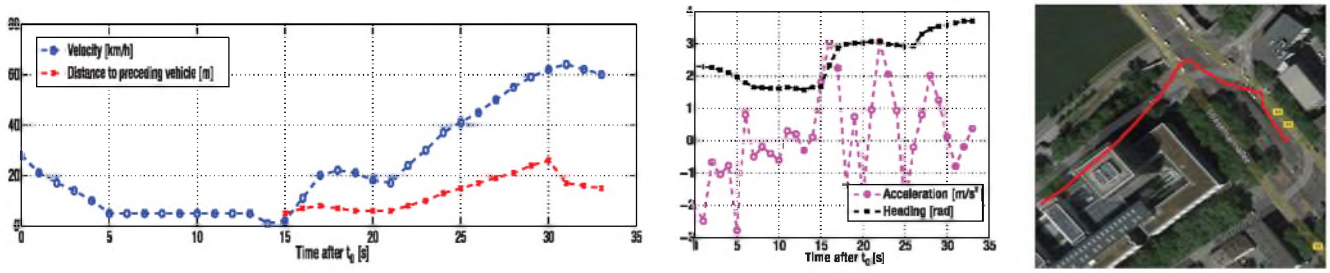

Fig. 3. Exemplary trace crossing intersection 5 with preceding vehicle. (a) Velocity and distance to preceding vehicle. (b) Heading and acceleration. (c) Position; Image data: Aerowest, Google; Map data: GeoBasis- DE/BKG, Google.

A trace point is a state vector of the driving dynamics and position at a given time. To be able to create a time series out of these state vectors, each of them are associated with a session ID (SID $\in \mathbb{N}$) and a timestamp (t_s):

$$\mathbf{y}(\mathrm{SID}, t_s) = \begin{bmatrix} \phi & \hat{=} & \text{WGS-84 latitude} \\ \lambda & \hat{=} & \text{WGS-84 longitude} \\ v & \hat{=} & \text{velocity} \\ \dot{v} & \hat{=} & \text{acceleration} \\ \theta & \hat{=} & \text{compass heading} \\ d_{\mathrm{ACC}} & \hat{=} & \text{distance to preceding vehicle} \end{bmatrix}. \tag{1}$$

After concatenation of all state vectors of an individual SID, one retrieves a *trace* or *trajectory*

$$T_{\mathrm{SID}} = [\mathbf{y}(\mathrm{SID}, t_0), \mathbf{y}(\mathrm{SID}, t_0 + \Delta t), ..., \mathbf{y}(\mathrm{SID}, t_0 + n\Delta t)] \tag{2}$$

For the sim$^{\mathrm{TD}}$ dataset, data were recorded and wirelessly transmitted to a database server with a sample rate of 1 Hz. Detailed information about the project can be found at [3]. Figure 3 shows an exemplary representation of such a time series of a vehicle passing an intersection behind another vehicle. When the vehicle approaches the turn (i.e. the significant heading change) at $t = 15$ s, it decelerates until it reaches the vertex of the turn. This demonstrates the physical properties of road traffic: To ensure safe and controlled travel around corners, drivers need to pass the turn with a certain maximum speed, dependent on the vehicle dynamics and curvature of the road (see [17] for details on a possible model). Further, a preceding vehicle is detected. The distance is measured via the radar of the ego vehicle.

The position plot with an added satellite image (figure 3(c)) illustrates one peculiarity of the dataset: As the uncorrected Global Navigation Satellite System (GNSS) localization from production vehicles is used, the position inside cities and other occluded areas could be inaccurate by several meters. In this particular example, the GNSS positions are visible. The vehicle's exact path can only be speculated about with the help of the satellite image. However, for the later approach only the heading and lateral displacement to a reference path (e.g. the geometry of the navigation link leading to the reference node) are considered, eliminating the need for a high-precision localization (e.g. using WAAS/EGNOS). In the spirit of only using data which can be inferred for other vehicles, the turn signal feature is omitted as well. Additionally, this allows the intent prediction for drivers setting the turn signal too late or not at all.

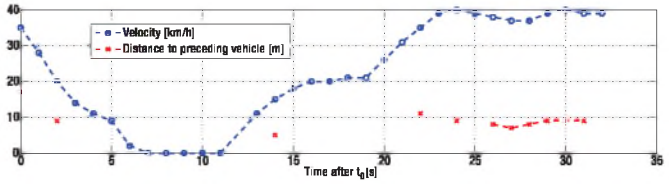

Fig. 4. Exemplary trace with signal drop-outs.

Another challenge arises from the drop-outs of the radar signal in the sim$^{\mathrm{TD}}$ dataset. In some cases (see figure 4), preceding vehicles seem to disappear and appear for several time steps, whereas–based on the behavior of the host vehicle–it can be speculated that another vehicle is present and cannot be detected all the time. Consequently, the chosen machine learning algorithms have to be robust with regard to signal noise and drop-outs of the front-facing radar data and the uncorrected GNSS localization.

B. Intersection-Wise Data Extraction

This paper focuses on creating machine learning models for individual navigation options at intersections. However, since the driving data are stored session-wise–covering a whole trip–only relevant data points in the vicinity to an intersection are extracted. The challenge of finding these *relevant* points is best explained by a short walkthrough of the first two steps of the

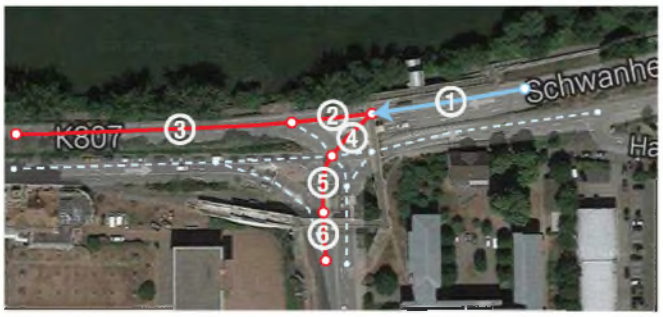

Fig. 5. Logical representation of streets through navigation links (intersection 7); Image data: Aerowest, Google; Map data: GeoBasis-DE/BKG, Google.

intersection-wise data extraction for a real-world example: Figure 5 shows the satellite image of an intersection in Frankfurt. The street network is automatically extracted in a graph-like structure from a navigation map, with *nodes* being vertices, connected by *links* (edges). The links can be polylines to allow a more precise street mapping. For this example, it is decided to collect data for the route decision following the node from the *reference link* ①in the direction indicated by the arrow. Starting at this node, two route options are obvious: Following link ②or link ④. These options occuring directly after the reference node are called the *initial route options*. However, if these routes are not expanded further, problems might arise when associating each trace with one of these routes: As seen in the previous section, the GNSS localization is affected by heavy noise. To still enable robust route association, the links are expanded further to ensure an adequate distance between the route endpoints. The two expanded route options are in this case: Route 1 with ①→ ②→ ③, route 2 with ①→ ④→ ⑤→ ⑥. Hence, the spatial size of the bounding box of the inscribed routes for extracting trace points is found.

1) Expansion of Possible Routes: As the logical representation of street networks are graphs, the task of expanding possible routes can be expressed as a graph traversal task. For this paper, it is decided to ignore routes that involve a 180 degree turn or travel the same link twice. This further reduces the problem to a tree traversal task. In general, the route decision tree, starting at the reference link, is continuously traversed breadth-first and expanded dynamically until certain criteria are met. These are:

1) Never travel the same navigation link twice
2) Expand to a maximum of 1 km
3) Only consider legal routes
4) If expanding backwards, expand at least 5 s (determined by the maximum speed associated with the link)
5) If expanding forwards, ensure that the endpoints of the initial route options have a minimum distance of $2r_c$ (the *corridor radius*, see step "Route Association")

The last constraint only applies to expanded routes that represent the initial route options, i.e. even when two routes are less than $2\,r_c$ apart, but belong to the same initial route option, they're not expanded further. This prevents pointless expansion, when the resulting classification label for the route decision evaluated at the reference node would remain the same either way.

2) Trace Extraction: After the respective initial route options have been expanded to an appropriate distance from each other, the trace points in that area are extracted. For the data sources, a query is set for all points inside a specific coordinate rectangle. However, before the trace points can be connected to traces based on their session ID, they have to be split up to individual intersection crossings: As the session IDs are only issued once at the beginning of a journey, one vehicle can cross the same intersection multiple times with the same session ID. Hence, the traces are split for each intersection crossing and the traces are now transformed from a WGS-84 datum to a local, cartesian Gauss-Krüger coordinate system [29], with the reference node being at the origin. All following computations are thus executed in a two-dimensional euclidean space.

3) Route Association: The dataset for each intersection contains not only the desired traces, but rather a great amount of either physically impossible traces (e.g. crossing a river) or traces going in the wrong direction. The next step–the route association–is responsible for two major tasks: filtering and associating the traces to the expanded routes. In the end, this step will also yield the route label for the respective traces. The process of route association is divided in three steps: Corridor matching, gate activation and trace cropping. For each trace, these steps are executed for each route option. If one trace matches more than one route, this ambiguity is resolved at the last step.

The first step–corridor matching–represents the coarsest filtering step. For each trace point, the perpendicular distance d_c to the nearest route segment is computed. If this distance is smaller than the corridor radius r_c, the respective trace point is marked as valid. The first and last valid trace points define the trace point interval for further computations. If one point inside this interval does not fulfill the corridor criterion ($d_c < r_c$), the whole trace is discarded for the route at hand. The corridor criterion filters vast outliers, but is still ignoring the fact that some parts of the routes are shared (e.g. the reference link). Thus, by introducing *gate activation*, the direction as well as completeness of a route match is evaluated. As previously mentioned, links can be given as polylines of multiple support points to allow a detailed route geometry. These points are used to create *gates*, by creating a perpendicular line on the route, spanning one corridor radius to the left and to the right. If a trace fails to activate all gates–traversing in the right direction within a maximum distance–it is not further considered for this route. These criteria ensure the removal of traces that either stop prematurely, travel the wrong direction or belong to other routes. Figure 6 shows examples for acceptable (green) and unacceptable (red) traces.

To reduce the storage consumption for the extracted traces, they are now cropped to the start and end point of the route. Although an adequate route expansion should prevent one trace being assigned to multiple routes, for some cases (e.g. when the corridor radius is chosen too small and thus the routes are not expanded far enough) this ambiguity needs to be resolved. As there are multiple concrete choices available for the route assignment, this problem is reduced to simply comparing the trace to the available routes and choosing the best fit. For comparing two trajectories in 2D space it is decided to use the *weak*

Fig. 6. Gate activation. One trace is invalid (red) since it fails to activate gates ⑥ and ⑦. Another red trace activates all gates but gate ③ inside the corridor radius (dashed blue line). The valid, green trace activates all gates inside the corridor radius. Image data: Aerowest, Google; Map data: GeoBasis-DE/BKG, Google.

Fig. 7. Matched routes for two urban intersections; Map data: GeoBasis-DE/BKG, Google. (a) Intersection 1. (b) Intersection 2.

Fréchet distance [30] which was already utilized in [31] for route matching. The route with the smallest weak Fréchet distance to the trace is then chosen as the final route assignment. Figure 7 shows the final route matching for two intersections.

C. Intersection-Centered Reference System

Until now, all trace points were primarily represented by their timestamps and their corresponding latitudes and longitudes on a map. For the following processing though, a one-dimensional representation based on the distance to the reference node of the navigation links (in most cases the intersection center) is advantageous for several reasons:

1) The trace information becomes independent from the geographic location of the intersection
2) A distance-based reference system is a more intuitive representation of trajectories
3) The trajectories become more comparable and are invariant to the duration of stops

To allow for a precise representation of curved road trajectories, it is decided to introduce a Frenet coordinate system along the sequence of navigation links for each route with its origin at the reference node. For the actual transformation, all trace points are matched to their nearest route segment. This results in an explicit representation by the longitudinal position on the sequence of navigation links (s) and the lateral displacement (n).

D. Feature Extraction & Labeling

After all traces for the respective intersections have been converted into the same format, the feature extraction creates an adequate numerical representation of the traces in order to apply the machine learning methods. Additionally, the existing traces have to be marked with their actual intent, this process is also known as labeling. For this paper, the vehicle dynamics as well as its position are chosen to be extracted, since they feature a high predictive power for driver intent prediction tasks (see Section I, [13], [15], [17]).

When building a machine learning model intended for live prediction, it has to be made sure that all features that are extracted from the dataset can also be extracted during an online drive. Specifically, when building a feature vector, only points from the past can be included and reference systems equal for all intents have to be chosen. In detail, the chosen features are extracted as follows:

Distance to Reference Node: Distance s to the reference node, computed via the navigation link geometry.
Velocity: The current velocity v in ms^{-1}, either retrieved from the vehicle odometry (recommended) or computed from the GNSS position (as used im $\mathrm{sim}^{\mathrm{TD}}$).
Acceleration: The current acceleration a in ms^{-2}, retrieved from the vehicle odometry.
Heading Difference: To provide the ability to generalize to multiple intersections in the future, only the heading difference $\Delta\varphi$ (computed from the GNSS positions) in rad of the current compass heading compared to the navigation link geometry is used.
Distance to Preceding Vehicle: The distance d to the vehicle in front of the own vehicle, in meters. Measured by the front-facing far range radar.
Flag for Presence of Preceding Vehicle: Boolean flag (false $=$ 0 or true $=$ 1) to indicate the presence of a preceding vehicle, output from radar.
Lateral Displacement: Lateral distance n to the navigation link geometry.

This results in the feature vector $\mathbf{x}(s)$ sampled at a distance s. Additionally, these points are not only sampled at the current distance (s_0), but also at previous distances, resulting in the final feature vector:

$$\mathbf{x} = [\mathbf{x}(s_0 - n_h \cdot s_h), \ldots, \mathbf{x}(s_0 - s_h), \mathbf{x}(s_0)], \qquad (3)$$

with a number of n_h points sampled every s_h meters. As for this approach, these parameters were chosen empirically to sample 4 points every 10 meters, resulting in a feature vector length of 35.

After all features have been extracted and converted to a numerical format, one last step has to be executed: The labeling of the trace, i.e. assigning it the actually driven intent. As a reminder, the overall driver intent consists of the route intent and stopping intent. The route intent is given by one of the initial navigation options and is labeled for all feature vectors created from the trajectory. The stopping intention however requires another approach: Before analyzing the data for an intersection,

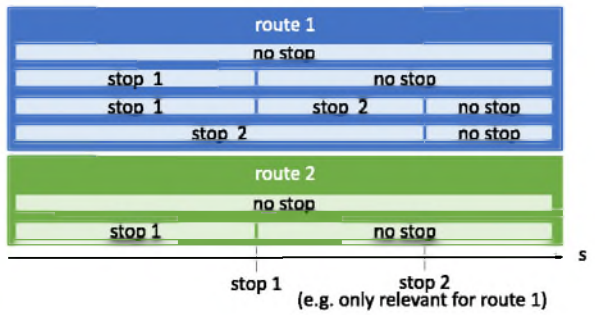

Fig. 8. Possible route and stopping intents, depending on the distance s to the reference node.

TABLE I
MEAN UAR AND TP@5FP OVER NINE INTERSECTIONS, EVALUATED AT FOUR TTI. BEST AVERAGE MEASURES PER INTENT AND TTI ARE SET IN BOLD FONT.

		Route Intent		Stopping Intent	
TTI [s]	Measure	SVM	RDF	SVM	RDF
2.88	UAR	0.67	**0.75**	0.73	**0.77**
	TP@5FP	0.42	**0.59**	0.44	**0.58**
2.16	UAR	0.67	**0.76**	0.70	**0.78**
	TP@5FP	0.44	**0.57**	0.45	**0.57**
1.44	UAR	0.67	**0.79**	0.67	**0.80**
	TP@5FP	0.46	**0.58**	0.48	**0.61**
0.72	UAR	0.75	**0.79**	0.70	**0.80**
	TP@5FP	0.50	**0.61**	0.52	**0.71**

all stoplines are labeled manually, including the information for which routes the stoplines are valid. For the future it is considered to crowdsource these stoplines from the dataset [32]. From each recorded vehicle trace the executed stop maneuvers are known, i.e. vehicle velocity approaches zero between 15 m before and 5 m after a route relevant stopline. Therefore all feature vectors prior to a particular stop are associated with the respective stopping intent. If there are multiple stops on a route the intent of the successive stop is not anticipated before the preceding stop has been performed and thus is only labeled starting from the preceding stop. After the last stop the vehicle trace is labeled with 'no stop' (see figure 8). An example is given for an intersection with two initial route options and two identified stoplines:

Route Intent: { Route 1, Route 2 }
Stopping Intent: {Stopline 1, Stopline 2, No Stop}

The possible class labels for the overall intent (route and stopping combined) are hence the power set of the these two intent sets. In order to ease computation with the probabilities produced for each label by a later classifier, the power set of labels is translated into a matrix (rows: route intent; columns: stopping intent). By then computing either the row or column sum, the overall probability for the routes or stoplines can be inferred. The choice of the power set method is motivated by the increased insight and use of synergies: By laying the classification results out in a matrix, one can easily identify common misclassifications and parametrize an ADAS function accordingly. Moreover, the classification of route and stopping intent is closely related, e.g. identifying a stopping intent can hint towards taking a specific navigation option.

III. MACHINE-LEARNING APPROACHES

Following the definition and extraction of features, the implementation of the machine learning algorithm used–Random Decision Forest–is covered. For this paper, the implementation of Leo Breiman [35] is used. RDF is an ensemble learning method: By creating and combining a multitude of weak learners (decision trees in this case), one strong classifier is built. To allow an evaluation of the individual feature information gains and an incremental training of the RDF in the future with minimal data pre-processing, it is decided to perform no PCA or standardization on the training data.

IV. RESULTS

For nine different intersections (with over 50,000 crossings), the performance is retrieved for each of the two machine learning algorithms and averaged for an overall representation of the classifier performance (see table I). Each evaluation is carried out using 10-fold cross validation. For this, the dataset is divided into two parts: 90% are used to train the machine learning model as described previously, the remaining 10% are utilized to assess the algorithm's ability to generalize to unknown data. This is repeated 10 times in a round-robin manner, so that each trace will be part of the evaluation once.

For each of the methods, a general overview on the performance is given by providing two major performance measures (Unweighted Average Recall (UAR) and true positive rate at 5% false positive rate (TP@5FP)) for four different worst-case times to the reference node (TTI), assuming the vehicle travels constantly with the maximum allowed speed inside cities: 2.88 s ($s = -40$ m), 2.16 s ($s = -30$ m), 1.44 s ($s = -20$ m) and 0.72 s ($s = -10$ m). TP@5FP is only defined for binary decision. If more than two classes exist, this measure is computed for one-vs-all classifications (i.e. creating as many binary decisions as there are class power sets) and averaged. UAR is the unweighted average of the class-wise recalls and is better suited to imbalanced class distributions than accuracy. With imbalanced class distribution as given by the dataset studied here, a high accuracy could be obtained trivially, by always predicting the majority class.

A. Route Intent Prediction

Throughout all TTI, multi-lane intersections stand out as the intersections with the best performance, with the lateral displacement contributing heavily to the performance. Whereas multi-lane intersections provide an adequate prediction even at TTI = 2.88 s and gain little performance when getting nearer to the navigation link reference node, the performance for single-lane intersections increases remarkably with decreasing TTI. For these intersections, it can be observed that the classifier relies heavily on the velocity feature.

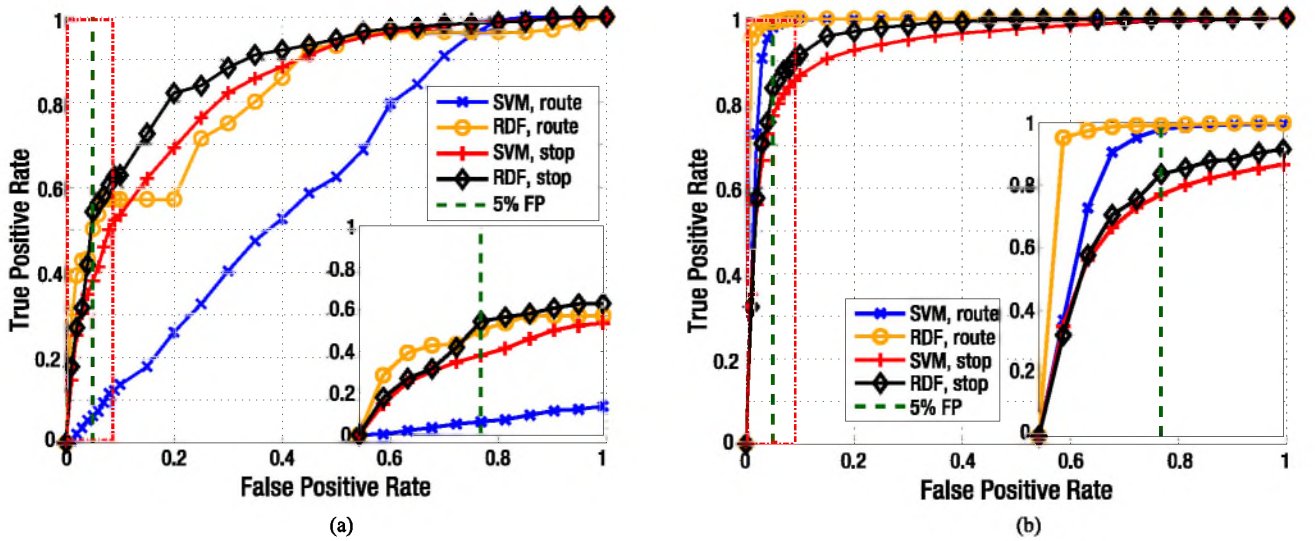

Fig. 9. ROC curves for the SVM and RDF predictions (route and stopping intent) at two different intersections at $\mathrm{TTI} = 2.16\mathrm{s}$. Additionally, insets show the area around 5% false positives with more details. (a) Intersection 1, single-lane. (b) Intersection 2, multi-lane.

B. *Stopping Intent Prediction*

Using the combined route and stopping intent labeling introduced in II-D, the RDF classifier performs reasonably well on the $\mathrm{sim}^{\mathrm{TD}}$ dataset (table I). As for the route intent prediction, the same observation of a performance gain towards the center of the intersection can be made. Remarkably, the overall performance is comparable or even better than for the route prediction and not dependent on the topology of the intersection (mostly, the number of lanes). This is mostly attributable to the fact that stopping prediction relies on the velocity and acceleration features the most.

At 20 m to the reference node (i.e. $TTC = 1.44$ s, assuming the vehicle is driven at the maximum allowed speed), our overall numbers are similar to the ones obtained by [20], which is remarkable since in their study a much more constrained environment was used for the evaluation (multi-lane intersection with high-precision map and localization, high sampling rate).

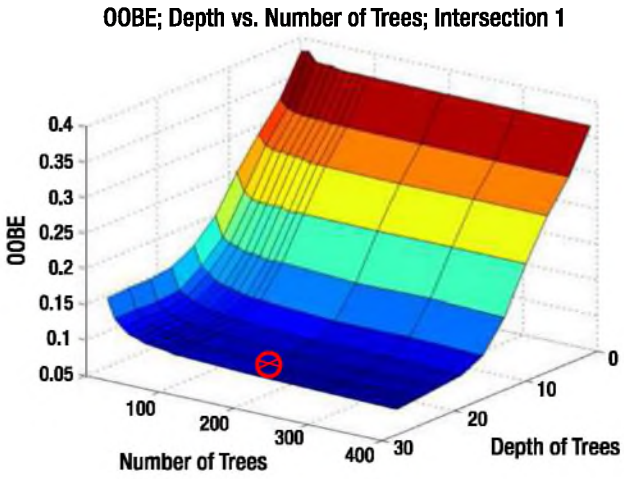

Fig. 10. Influence of RDF parameters on out-of-bag error: Depth versus Number of Trees with 6 features selected for node splitting.

C. *Influence of RDF Parameters*

From literature [36], [37] it is well-known that the RDF classifier is sensitive to several parameters. Among these, the influence of the three most important are evaluated: Number of trees, depth of trees and number of features selected for a node split. As the first two parameters represent the topology of the RDF, they are evaluated together. As a metric, the out-of-bag error (OOBE) [35] is chosen, as it represents an intrinsic measure for the ability to generalize to unknown data, produced during training. This eliminates the need for time-consuming cross validation but still provides a common basis for comparison. The OOBE is retrieved as follows: Each tree in the Random Forest is trained on two thirds of the dataset. After training, the other third ("out-of-bag") is evaluated by the tree. The OOBE represents the average of the classification error on the out-of-bag samples over all trees. Figure 10 shows the OOBE for the number and depth of the trees for one intersection. One can see that the RDF is highly sensitive to the depth of trees and mildly sensitive to the number of trees. For both parameters the intuitive principle "the more, the better" holds, since random forests are robust against overfitting [35]. For this RDF approach, the number of trees was chosen to be 200, with each tree being 20 levels deep at maximum (Figure 10), being a compromise between performance and RDF model complexity. As the evaluation for the number of selected features showed that the decision trees are not very sensitive to the number of features selected for node splits, in accordance with the recommendation by Breiman [35], the number of selected features is chosen to be the first integer less than $\log_2(M) + 1 = 6$, with $M = 35$ denoting the total number of features.

D. *Feature Information Gain*

To be able to perform a sensitivity analysis, the *information gain* (IG) [38] is computed for each feature. The IG measures

the predictive power for a feature based on entropies. For this, the weighted average of all entropies for the class distribution assuming all possible feature values is subtracted from the entropy of the overall class distribution ($H(c)$):

$$\mathrm{IG} = H(c) - \sum_{\forall v \in f} P(c \mid f = v) \cdot H(c \mid f = v), \qquad (4)$$

with $\sum_{\forall v \in f} P(c \mid f = v) = 1$ and $P(c \mid f = v)$ being the likelihood for the feature f assuming the value v (i.e. the relative frequency taken from the histogram). The entropy $H(X)$ for finite samples from an arbitrary random variable X can be calculated as follows:

$$H(X) = -\sum_{i} p(x_i) \log\left(p(x_i)\right), \qquad (5)$$

where $p(x_i)$ denotes the relative frequency of the value x_i for the random variable X. For c describing a two-class distribution (with classes a and b), the entropy would result to $H(c) = -p(a) \log\left(p(a)\right) - p(b) \log\left(p(b)\right)$. The unit of the entropy, and thus also of the information gain as difference of entropies is Shannon (Sh).

A feature with an information gain of 0.0 would contribute nothing to the prediction, whereas a feature with 1.0 information gain would most likely be the label itself. The data mining workbench *Weka* [39] provides an information gain-based attribute evaluation, which is used for this paper. For both plots, the *meta-features*, such as distance to the reference node as well as the flag, whether a preceding vehicle exists, are left out, as they should only be auxiliary to the other features.

The information gain for features of two different intersections can be seen in Figure 11, computed at the four distances corresponding to the known TTI. The evaluation displays once again the importance of the lateral displacement and heading difference for multi-lane intersections (Figure 11(b)). Furthermore, for intersections where turns begin before the reference node, the heading difference becomes increasingly important respectively (Figure 11(a)). All in all, in comparison the features have a higher predictive power on intersection 1 than on intersection 2, as can be seen from the sum of the IGs.

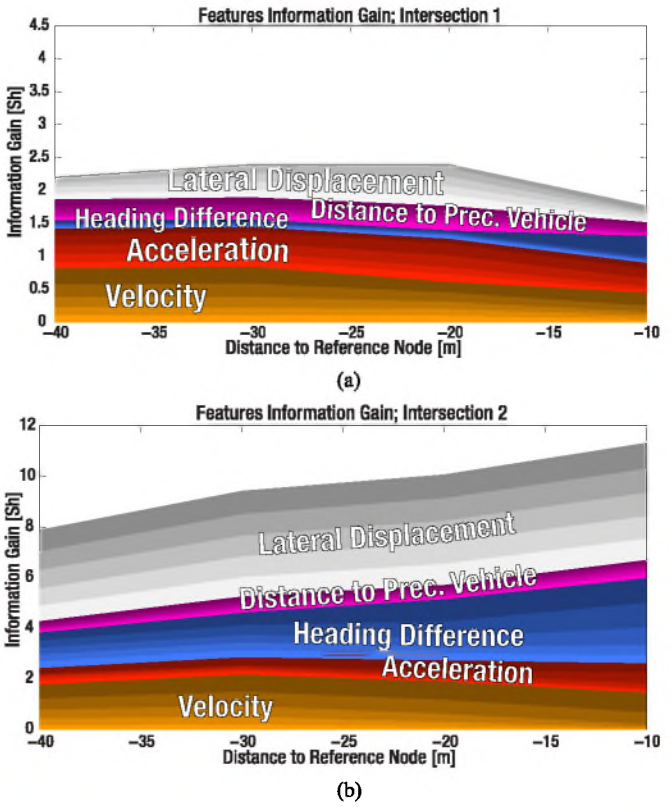

Fig. 11. Information gain for RDF features for two different intersections. The more recent the feature, the darker the shaded area. (a) Intersection 1. (b) Intersection 2.

E. Comparison to Baseline Approach

As a state-of-the-art baseline, linear SVMs [33] are selected. They represent a popular machine learning algorithm and usually perform well in classification tasks due to their maximum margin property, as demonstrated in [21]. In order to reduce the dimensionality and thus speed up training, the data are standardized to zero-mean and unit-variance and a Principal Component Analysis (PCA) [34] is performed, retaining 95% of the original variance.

Table I compares the average measures for both classifiers for the prediction of the route and stopping intent, while figure 9 provides the ROC curves for two intersections. Throughout all TTI and performance measures, the RDF manages to outperform the chosen SVM (PCA pre-processing, linear kernel, unweighted classes) under the given specific experimental conditions. It can be suspected that the RDF utilizes its strength to deal with data sets with great variation and many outliers best. In detail, there are contributing effects: At some intersections, the SVM classifies all instances as the majority class, i.e. the route taken by most vehicles. For highly imbalanced route distributions (e.g. popular routes versus merely taken routes), this poses serious problems. Also, in some cases the SVM's doesn't follow the general principle that the performance increases when moving towards the reference node: Single TTI evaluations exhibit deteriorated performance. All in all, the maximum margin principle could interfere with the classification of the diverse dataset.

V. Conclusion and Future Work

The results allow concluding that this concept of driver intent prediction is suitable for the route and stopping intent prediction at urban intersections. The proposed method integrates a dataset with recorded car fleet driving data containing the position, velocity, acceleration, heading and distance to the preceding vehicle. The recorded vehicle traces are then robustly assigned to routes from a navigation database, keeping the inaccuracy of the GNSS localization of today's production vehicles in mind.

To enable the machine learning methods to generalize to multiple intersections in the future, the features are selected to be independent of the direction or location of the intersection: The position is represented by the distance to the reference node and the lateral displacement to the navigation link, and the vehicle direction being described by the heading difference to the

navigation link direction. Using these features and a Random Decision Forest, the route intent prediction achieves a TP@5FP of 59% at a TTI of 2.88 s. The stopping intent is predicted with TP@5FP = 58% at the same TTI. The SVM as baseline classifier scores a TP@5FP of 42% for route and 44% for the stopping intent 2.88 s from the navigation link reference node.

This, together with the UAR allows the conclusion that the proposed RDF performs better than the SVM throughout all TTI for the given experimental conditions. As a rule of thumb, for the range of UAR encountered in this paper, an absolute UAR difference of 1% absolute is sufficient to yield statistical significance at the 0.1% level, according to a z-test. The analysis of the information gain for all features reveals that for road intersections with only one lane towards the reference node, velocity and acceleration possess the highest predictive power. For intersections with clearly differing headings for the respective initial route options, the heading difference to the reference navigation link gains high predictive power towards the beginning of the turn. For multi-lane intersection (i.e. with separate lanes for each route option), the lateral displacement to the reference link contributes heavily to the route prediction performance.

Finally, an outlook is given on possible areas of improvement. It can be considered to transfer the learned models via wireless communication and in turn collect data on a larger car fleet. This eliminates the shortcoming of the current evaluation: The models are trained and evaluated using the sim^{TD} dataset, possibly overfitting on the driving behavior in this particular dataset. Furthermore, integrating data sources other than sim^{TD}, where drivers were instructed to travel a specific route, might result in more balanced class distributions. To further improve the classification performance, more data from on-board ADAS sensors (camera, radar, lidar) should be extracted, utilizing the perception of the static and dynamic environment.

In the end, training one model for a group of intersections seems beneficial: By clustering together similar intersection topologies, the data from all grouped intersections are used for the training of one model, vastly reducing the number of required machine learning models and thus the occupied disk space. Sensible intersection parametrization comprises the number of lanes, maximum allowed speed, number of outgoing links and heading changes at the decision point. The intersections are then assigned to clusters by an arbitrary clustering algorithm, where each cluster has its own intent prediction model.

ACKNOWLEDGEMENTS

The authors wish to thank M. Kleinsteuber for the fruitful discussions and comments.

REFERENCES

[1] Statistisches Bundesamt, "Statistik des Kraftfahrzeug- und Anhängerbestandes (German)," 2014 (accessed May 2, 2014). [Online]. Available: https://www-genesis.destatis.de/genesis/online/logon?sequenz=tabelleErgebnis&selectionname=46251-0001

[2] Statistisches Bundesamt, "Statistik der Straßenverkehrsunfälle mit Personenschaden (German)," 2014 (accessed May 2, 2014). [Online]. Available: https://www-genesis.destatis.de/genesis/online/logon?sequenz=tabelleErgebnis&selectionname=46241-0003

[3] H. Stubing *et al.*, "simTD: A car-to-x system architecture for field operational tests [Topics in Automotive Networking]," *IEEE Commun. Mag.*, vol. 48, no. 5, pp. 148–154, May 2010.

[4] "Forschungsinitiative Ko-FAS," 2014 (accessed May 2, 2014). [Online]. Available: http://ko-fas.de

[5] F. Valent, F. Schiava, C. Savonitto, T. Gallo, S. Brusaferro, and F. Barbone, "Risk factors for fatal road traffic accidents in Udine, Italy," *Accident Anal. Prevention*, vol. 34, no. 1, pp. 71–84, 2002.

[6] Statistisches Bundesamt, "Statistik der Straßenverkehrsunfälle (German)," 2014 (accessed May 3, 2014). [Online]. Available: https://www-genesis.destatis.de/genesis/online/logon?sequenz=tabelleErgebnis&selectionname=46241-0001

[7] Y. Demiris, "Prediction of intent in robotics and multi-agent systems," *Cogn. Process.*, vol. 8, no. 3, pp. 151–158, 2007.

[8] G. Csibra and G. Gergely, "'Obsessed with goals': Functions and mechanisms of teleological interpretation of actions in humans," *Acta Psychologica*, vol. 124, no. 1, pp. 60–78, 2007.

[9] A. N. Meltzoff, "The like me framework for recognizing and becoming an intentional agent," *Acta Psychologica*, vol. 124, no. 1, pp. 26–43, 2007.

[10] D. Windridge, A. Shaukat, and E. Hollnagel, "Characterizing driver intention via hierarchical perception–action modeling," *IEEE Trans. Human–Mach. Syst.*, vol. 43, no. 1, pp. 17–31, Jan. 2013.

[11] A. Sathyanarayana, P. Boyraz, and J. H. Hansen, "Driver behavior analysis and route recognition by hidden Markov models," in *Proc. IEEE Int. Conf. Veh. Electron. Safety*, Sep. 2008, pp. 276–281.

[12] H. Berndt, J. Emmert, and K. Dietmayer, "Continuous driver intention recognition with hidden Markov models," in *Proc. 11th IEEE Int. Conf. Intell. Transp. Syst.*, 2008, pp. 1189–1194.

[13] H. Berndt and K. Dietmayer, "Driver intention inference with vehicle onboard sensors," in *Proc. IEEE Int. Conf. Veh. Electron. Safety*, 2009, pp. 102–107.

[14] S. Lefevre, J. Ibanez-Guzman, and C. Laugier, "Context-based estimation of driver intent at road intersections," in *Proc. IEEE Symp. Comput. Intell. Veh. Transp. Syst.*, 2011, pp. 67–72.

[15] S. Lefevre, C. Laugier, and J. Ibanez-Guzman, "Exploiting map information for driver intention estimation at road intersections," in *Proc. IEEE Intell. Veh. Symp.*, 2011, pp. 583–588.

[16] M. Liebner, M. Baumann, F. Klanner, and C. Stiller, "Driver intent inference at urban intersections using the intelligent driver model," in *Proc. IEEE Intell. Veh. Symp.*, 2012, pp. 1162–1167.

[17] M. Liebner, F. Klanner, M. Baumann, and C. Ruhhammer, "Velocity-based driver intent inference at urban intersections in the presence of preceding vehicles," *IEEE Intell. Transp. Syst. Mag.*, vol. 5, no. 2, pp. 10–21, Apr. 2013.

[18] M. Liebner, C. Ruhhammer, F. Klanner, and C. Stiller, "Generic driver intent inference based on parametric models," in *Proc. 16th IEEE Int. Conf. Intell. Transp. Syst.*, Oct. 2013, pp. 268–275.

[19] M. Treiber, A. Hennecke, and D. Helbing, "Congested traffic states in empirical observations and microscopic simulations," *Phys. Rev. E*, vol. 62, pp. 1805–1824, Aug. 2000.

[20] G. S. Aoude, B. D. Luders, K. K. H. Lee, D. S. Levine, and J. P. How, "Threat assessment design for driver assistance system at intersections," in *Proc. 13th IEEE Int. Conf. Intell. Transp. Syst.*, Sep. 2010, vol. 1, pp. 1855–1862.

[21] G. S. Aoude, V. R. Desaraju, L. H. Stephens, and J. P. How, "Driver behavior classification at intersections and validation on large naturalistic data set," *IEEE Trans. Intell. Transp. Syst.*, vol. 13, no. 2, pp. 724–736, Jun. 2012.

[22] H. Rakha, I. El-Shawarby, and J. R. Setti, "Characterizing driver behavior on signalized intersection approaches at the onset of a yellow-phase trigger," *IEEE Trans. Intell. Transp. Syst.*, vol. 8, no. 4, pp. 630–640, Dec. 2007.

[23] N. Elmitiny, X. Yan, E. Radwan, C. Russo, and D. Nashar, "Classification analysis of driver's stop/go decision and red-light running violation," *Accident Anal. Prevention*, vol. 42, no. 1, pp. 101–111, 2010.

[24] C. Hermes, C. Wöhler, K. Schenk, and F. Kummert, "Long-term vehicle motion prediction," in *Proc. IEEE Intell. Veh. Symp.*, 2009, pp. 652–657.

[25] J. Wiest, M. Höffken, U. Kreßel, and K. Dietmayer, "Probabilistic trajectory prediction with gaussian mixture models," in *Proc. IEEE Intell. Veh. Symp.*, Jun. 2012, pp. 141–146.

[26] L. Malta, C. Miyajima, and K. Takeda, "A study of driver behavior under potential threats in vehicle traffic," *IEEE Trans. Intell. Transp. Syst.*, vol. 10, no. 2, pp. 201–210, Jun. 2009.

[27] A. Doshi, B. Morris, and M. Trivedi, "On-road prediction of driver's intent with multimodal sensory cues," *IEEE Pervasive Comput.*, vol. 10, no. 3, pp. 22–34, Jul. 2011.

[28] A. Jain, A. Singh, H. S. Koppula, S. Soh, and A. Saxena, "Recurrent neural networks for driver activity anticipation via sensory-fusion architecture," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2016, pp. 3118–3125.
[29] W. Großmann, "Geodätische Rechnungen und Abbildungen in der Landesvermessung," Stuttgart, Germany: Wittwer, 1976.
[30] H. Alt and M. Godau, "Computing the fréchet distance between two polygonal curves," *Int. J. Comput. Geometry Appl.*, vol. 5, no. 1–2, pp. 75–91, 1995.
[31] C. Wenk, R. Salas, and D. Pfoser, "Addressing the need for map-matching speed: Localizing global curve-matching algorithms," in *Proc. 18th Int. Conf. Sci. Statist. Database Manage.*, 2006, pp. 379–388.
[32] C. Ruhhammer, N. Hirsenkorn, F. Klanner, and C. Stiller, "Crowdsourced intersection parameters: A generic approach for extraction and confidence estimation," in *Proc. IEEE Intell. Veh. Symp.*, Jun. 2014, pp. 581–587.
[33] B. Schölkopf and A. Smola, *Learning With Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*, ser. Adaptive Computation and Machine Learning. Cambridge, MA, USA: MIT Press, 2002.
[34] I. Jolliffe, *Principal Component Analysis*. New York, NY, USA: Wiley, 2002.
[35] L. Breiman, "Random forests," *Mach. Learning*, vol. 45, no. 1, pp. 5–32, 2001.
[36] C. Strobl, A.-L. Boulesteix, T. Kneib, T. Augustin, and A. Zeileis, "Conditional variable importance for random forests," *BMC Bioinformatics*, vol. 9, no. 1, 2008, Art. no. 307.
[37] R. Diaz-Uriarte and S. Alvarez de Andres, "Gene selection and classification of microarray data using random forest," *BMC Bioinformatics*, vol. 7, no. 1, 2006, Art. no. 3.
[38] J. T. Kent, "Information gain and a general measure of correlation," *Biometrika*, vol. 70, no. 1, pp. 163–173, 1983.
[39] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten, "The weka data mining software: An update," *SIGKDD Explor. Newslett.*, vol. 11, no. 1, pp. 10–18, Nov. 2009.

Florian Gross received the M.Sc. degree in electrical engineering and information technology from Technische Universität München, Munich, Germany, in 2014.

His master thesis covered the driver intent prediction at urban intersections using car fleet datasets.

He is currently a Development Specialist at BMW Group, Munich. His research interests include sensors and applied machine learning for ADAS, highly automated, and autonomous driving.

Justus Jordan received the Diploma degree in mechatronics and information technology from Technische Universität München (TUM), Munich, Germany, in 2013. He is working toward the Ph.D. degree at BMW Group in cooperation with the Research Group for Geometric Optimization and Machine Learning, TUM Department of Electrical, Electronic, and Computer Engineering.

His interest focuses on research and development of autonomous driving systems, especially the analysis and prediction of driving scenes based on extracted knowledge from vehicle fleet data.

Felix Weninger received the Diploma and Doctoral degrees, both in computer science, from Technische Universität München, Munich, Germany, in 2009 and 2015.

He is currently a Senior Research Scientist at Nuance Communications, Ulm, Germany.

During 2013–2014, he was an Intern at Mitsubishi Electric Research Laboratories, Cambridge, MA, USA. His primary research interest includes machine learning applied to signal processing tasks, in particular tensor factorization and deep learning methods.

Dr. Weninger serves as a Reviewer for several IEEE journals including the IEEE TRANSACTIONS ON SIGNAL PROCESSING and the IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS. He has published more than 80 peer-reviewed papers in books, journals, and conference proceedings.

Felix Klanner received the Diploma degree in mechanical engineering from Technische Universität München, Munich, Germany, in 2005 and the Ph.D. degree in engineering from Technische Universität Darmstadt, Darmstadt, Germany, in 2008.

He joined BMW in 2007 and is currently the Director of Future Mobility Research, Southeast Asia, a joint initiative of BMW Group and Nanyang Technological University (NTU), Singapore. His main research interest include development of advanced driver assistance systems, applied machine learning for signal processing, and extraction of knowledge from car fleet data.

Dr. Klanner is a Reviewer for several IEEE journals and conferences, and Visiting Associate Professor at the School of Computer Science and Engineering, NTU. He published more than 50 patents and several peer-reviewed papers in books, journals, and conference proceedings.

Björn Schuller received the Diploma degree in 1999, the Doctoral degree in 2006, and the Habilitation and Adjunct Teaching Professorship in the subject area of signal processing and machine intelligence in 2012, all in electrical engineering and information technology from Technische Universität München, Munich, Germany.

He is a Tenured Full Professor heading the Chair of Complex and Intelligent Systems at the University of Passau/Germany and a Reader in machine learning in the Department of Computing, Imperial College London, London, U.K.

Dr. Schuller is the President-Emeritus of the Association for the Advancement of Affective Computing, Elected Member of the IEEE Speech and Language Processing Technical Committee, and a Member of the ACM and ISCA. He (co-)authored 5 books and more than 500 publications in peer-reviewed books, journals, and conference proceedings leading to more than 12 000 citations.