Honda Research Institute EU
Honda Research Institute Europe GmbH
https://www.honda-ri.de/

General Behavior Prediction By A Combination Of Situation Specific Models

Sarah Bonnin, Thomas H Weisswange, Franz Kummert, Jens Schmüdderich

2014

Preprint:

This is an accepted article published in IEEE Transactions on Intelligent Transportation Systems. The final authenticated version is available online at: https://doi.org/[DOI not available]

General Behavior Prediction by a Combination of Scenario Specific Models

Sarah Bonnin, Thomas H. Weisswange, Franz Kummert, *Member, IEEE,* and Jens Schmuedderich

***Abstract*—Before taking a decision, a driver anticipates the future behavior of other traffic participants. However, if a driver is inattentive or overloaded he may fail to consider relevant information. This can lead to bad decisions and potentially result in an accident. A computational system designed to anticipate other traffic participants' behaviors could assist the driver in his decision making by sending him an early warning when a risk of collision is predicted. Existing research in this area usually focuses on only one of two aspects: quality or scope. Quality refers to the ability to warn a driver early before a dangerous situation happens. Scope is the diversity of scenarios in which the approach can work. In general we see methods targeting broad scope but showing low quality and others having narrow scope but high quality. Our goal is to create a system with high quality and high scope. To achieve this, we propose an architecture that combines classifiers to predict behaviors for many scenarios. In this paper we will first introduce the generic concept of such a system applicable to highway scenarios as well as inner-city scenarios. We will show that a combination of general and specific classifiers is a solution to improve quality and scope based on a concrete implementation for lane change prediction in highway scenarios.**

I. INTRODUCTION

DRIVING is a complex task. Worldwide, on average 1.2 million people are killed in road crashes each year and as many as 50 million are injured [1]. According to some estimates [2], 93% of accidents are caused by driver errors, 80% of them by a driver's inattentiveness within three seconds of an accident.

Some investigations [3] show that an accident would need to be anticipated at least two seconds before a collision to ensure that the driver has time to react. Recent studies show the benefit of Advanced Driver Assistance Systems (ADAS) in simple situations (such as Autonomous Emergency Braking (AEB) to targets on straight roads), but emphasize the shortcomings of existing ADAS in complex situations, such as at junctions or intersections [4].

When driving, a human is able to perceive his environment to understand the state of his surroundings and make conclusions about their future actions. Nowadays, technology is advanced enough to precisely detect static infrastructure and other traffic participants. To use this context for behavior prediction we propose designing a set of scenario models: each model represents expert scene knowledge using a set of specific features. Classifiers are trained to recognize future behavior from this set of features. By doing so, behavior prediction can be considered as a classification problem. In our current work, classifiers are trained to predict whether a surrounding entity will come into the way of the ego-vehicle and imposes a risk of collision.

We propose a system with the goal of sending early warnings to the ego-vehicle driver and assisting the driver in complex situations based on the prediction of surrounding traffic participants' behaviors. To cope with the large variety of scenarios, the scenario-specific classifiers are organized in a tree. By activating only those classifiers fitting the present scenario, a generic system applicable to different scenarios is achieved.

The paper will be organised as follows: Section II gives an overview on related work. Section III presents the formal concept of the system. Its application to highway and inner-city scenarios is explained in sections IV and V. We will demonstrate the benefits of the proposed methods in section VI by applying it to cut-in prediction on a highway. We also show that the combination of specific classifiers in the system outperforms the simple classifier. Finally, section VII gives a conclusion and presents the future work.

II. RELATED WORK

Behavior prediction has been a main topic of research in the last 10 years but, as we will explain in this section, the understanding of meaning and purpose varies.

Our approach aims to increase the anticipation time of the driver by using behavior prediction. Several studies about the prediction of behaviors exist, focusing on scope or quality to varying extends. Quality refers to the accuracy of warning a driver of a dangerous situation. Scope refers to the diversity of scenarios in which an approach can work.

A. Behavior Recognition and Short Term Prediction

Nowadays, cars are equipped with numerous sensors that supply a variety of information about the driver and the driving environment. The final objective of such equipment is to make driving more enjoyable and safe. In order to be able to offer assistance beyond crash mitigation, it would be beneficial to know what the driver is currently doing. This is what behavior recognition is used for. The most widely used method to recognize such driver maneuvers is the Bayesian Network (BN)[5][6]. Recognition of a behavior can only be done while it is performed.

S. Bonnin is with the Research Institute for Cognition and Robotics, Bielefeld University, and Honda Research Institute Europe GmbH, Offenbach am Main, Germany sarah.bonnin@gmail.com

F. Kummert is with the Faculty of Technology, Bielefeld University, Bielefeld, Germany Franz@techfak.uni-bielefeld.de

T.H. Weisswange and J. Schmuedderich are with the Honda Research Institute Europe GmbH, Offenbach am Main, Germany FirstName.LastName@honda-ri.de

To be able to avoid a collision, a system should not be reactive but predictive. The goal is to make a driver ready to face a dangerous situation early enough, so that he has time to react by an appropriate driving maneuver. Driving maneuvers can often be considered as a sequence of basic maneuvers. E.g. an overtaking maneuver may consist of the following segmentation: approach leading vehicle, lane change to the left, passing, and lane change to the right. By recognizing early sub-maneuvers, one can predict later behaviors much earlier [7][8].

A method to predict complex maneuvers such as cut-in is proposed in [9]. The novelty of their method is that they use information from situational context and the interaction between vehicles to model the scene and predict if a vehicle is about to cut-in in front of another vehicle approximately one second before it actually changes lane.

All these prediction methods are accurate up to 1.5 seconds into the future, but not for a longer time horizon. A driver needs to be warned at least two seconds in advance to have a chance of reacting in time [3]. The benefit these methods is, that they are usually working in a variety of simple scenarios. One could say that these methods trade off prediction quality at longer time horizons for having a broader scope.

B. Long Term Prediction

The methods described below can be called model-based approaches. The principle is to represent the scene by creating relations between vehicles and static infrastructure and look for features indicating a lane change [10][11][12], merging on the highway [13] or a turn maneuver [14]. These indications are derived from contextual information, e.g. knowing which lane a vehicle is driving in, knowing that it will move to a faster lane if approaching another car, etc. Such information is used to define specific features, feed them into a classifier and obtain classes that represent the future behavior of other traffic participants.

In [11], Kasper et al. propose a hierarchical system to predict various maneuvers on highways. The hierarchy starts on the sensor level and ends on the creation of relations between a vehicle and its surroundings. They define situational context features specific to highways.

Garcia Ortiz et al. [10] propose to use the Locally Weighted Projection Regression (LWPR) algorithm to predict accurate lane change maneuvers limited to the highway scenario.

Reichel et al. [13] uses a random forest algorithm to predict a complex and very specific maneuver such as convoy merging on highways.

Another very specific model is proposed by Lefevre et al. [14] to predict turn maneuvers at intersections using contextual informations from a digital map.

Many methods perform complex behavior prediction with high quality. These approaches show very good accuracy in the prediction for a long time horizon but their situation specificity prevents application to a wider scope of situations.

C. Prediction With Multiple Models

It seems to be very difficult to target both, wide scope and high quality at the same time. One possibility to overcome this limitation is to have different situational models for different scenarios and to use the position of the ego-vehicle in a digital map to activate the correct one [15][16]. The different models used by some of the participants in the DARPA Urban challenge were able to perform very well, profiting from the clearly defined and well distinguishable scenarios. One model at a time got activated using a digital map. In the real world, scenarios can become more complex and overlap with each other. In such a case, it can happen that the model activated based on the current location is not suitable to the current scene. For example, a model to predict cut-ins on the highway is not accurate at entrances or exits.

Graf et al. [17] also build a system based on the idea of improving prediction by using multiple models, but with a much finer granularity. They design a tree with three scenario branches (highway, inner-city, rural road). In each branch, they learn all possible object constellation with corresponding models, which are located in the leaves. A constellation is an abstraction of the relative positions of all elements in the current scene. This high model specificity allows the use of very simple classifiers for good quality, but requires a very large training dataset. The authors mention another disadvantage of that approach: It will fail in situations which have never been seen before.

A related idea for performing ego-vehicle driver intention recognition is used by Berndt et al. [18]. The authors have designed a set of models, each of which is able to predict a single type of maneuver. To save computation time and improve the scope and quality, only a subset of the models is active at a time based on the position of the ego-vehicle in a digital map which limits the set of possible behaviors. However, the authors also mention a limitation of their approach: The same maneuver can happen in many different scenarios and for different reasons. Therefore, the model might still fail in some cases.

All these approaches combine multiple models based on their current needs but except for Graf et al. who propose the learning of the sub-models, Berndt et al. and Fergusson et al. do not have a generic way of constructing such a system for other cases. We to propose a formalism that can be used to build any kind of multi-model system for behavior prediction. To define the set of rules, we take advantage of the three ideas already mentioned above. Our approach allows the combination of models with different levels of specificity which are activated based on the current context. We want to have both, multiple models active at the same time predicting different subsets of the scene, as well as groups of models that cannot be activated at the same time because they are mutually exclusive. We therefore choose a hierarchical tree structure. We will demonstrate the usefulness of our approach by some example systems and also show that this can be easily extended.

Similar to approaches discussed in the last paragraph, we believe that by combining multiple classifiers in a meaningful way, we can get the best of both worlds. The ego-vehicle location to access more detailed context information appears to be an important piece of information to choose which classifiers should be combined and when they should be

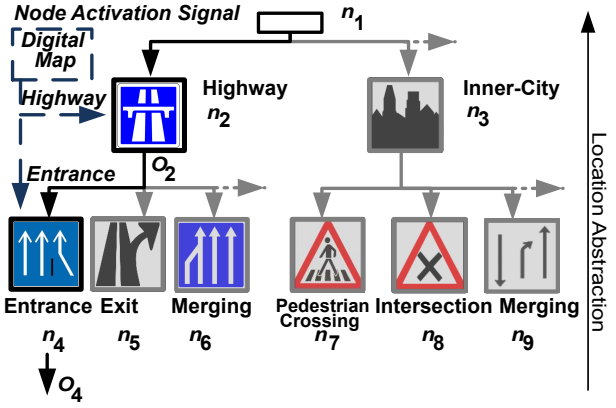


Fig. 1. Visualization of the scenario model tree. Each rectangle represents a node of the tree. The highway node and the inner-city node are on the first level of the tree. The nodes entrance, exit and merging are sub-nodes of the highway node. The nodes pedestrian crossing, intersection and merging are sub-nodes of the inner-city node. Black node frames have been activated by digital map. Black edges represent the active path. Grey node frames and grey edges are inactive. The highway is active and sends its output $\mathbf{o}_2$ to the active sub-node entrance which combines it with its own output to produce the final output $\mathbf{o}_4$.

applied. Our goal is to define a model combination formalism which allows the construction of systems that produce the best possible prediction in any scenario.

III. DEFINITION OF THE SCENARIO MODEL TREE

As explained in the related work section, a very specific model can predict behaviors with good accuracy and a long time horizon, but this comes at the cost of a very limited scope. We could increase the scope by using multiple scenario specific models. However, when combining these, three problem cases can occur:

- Case 1: Mutually Exclusive Scenarios
- Case 2: Dependent Scenarios
- Case 3: Competing Scenarios

To cover these cases, we will sort the scenarios into nodes of a tree structure. The layers of what we will call scenario model tree (SMT) represent the abstraction level, with the most generic scenarios at the root and the most specific ones at the leaves. Each node of the tree contains models that use the context information of the respective scenario as shown in Fig. 1.

SMT, solution to case 1: Generic scenarios are usually mutually exclusive because of a clear spatial separation. A model designed to predict behaviors in the highway scenario will perform poorly at predicting vehicles in inner-city and vice-versa. For example, lane changes in the inner-city can happen due to many different reasons (e.g. turning lanes, parking vehicles). To ensure that only one generic scenario can be active at a time, we define a node activation mechanism. A node gets activated based on the ego-vehicle's GPS position.

SMT, solution to case 2: A more specific scenario is usually connected to a more generic one in the sense that it uses some additional context information. Depending on each traffic participant's local context, the model for one or the other scenario will produce the best predictions. For example, at a highway entrance, vehicles in the left most lane do not behave differently from any other part of the highway and could therefore be predicted by the model of the generic highway scenario. On the other hand, the prediction for vehicles in the entrance lane will profit from taking into account the information of the entrance scenario. Due to the hierarchical ordering of the scenarios, a specific node will always be activated together with its parent.

SMT, solution to case 3: In some cases, two scenarios might both influence a traffic participant behavior at the same time. As in case 2, these scenarios overlap in time, but in addition they have a combined effect. For example, a vehicle driving in the lane to the left of the entrance lane might give way to an entering vehicle but might also overtake a slower predecessor. The models for both scenarios can contribute to improve the prediction. To ensure accurate results, the final behavior prediction will be a combination between the outputs of all active nodes based on a competition function.

This tree structure automatically provides a fall-back in case of localization errors and the non-activation of a specific node, because prediction based on the generic parent scenario will always cover the most usual behaviors.

In the following we will define the set of rules needed to create our tree structure.

A. Structure Of The SMT

- An SMT is composed of a set of nodes $N = \{n_i | i = 1...I\}$ where I is the total number of nodes N as shown in Fig.1.
- Each node n_i has exactly one parent n_g.
- A node $n_i = (M_i, b_i)$ is made of a set of models $M_i = \{m_i^j\}_{j=1...J_i}$, where J_i is the number of models m_i^j of the node n_i, and a competition function b_i.
- For each traffic participant detected in every timestep, a node n_i has an output o_i, which is the confidence of the prediction for this traffic participant.

To create models specific to a scenario, the principle is to define a set of features based on the situational context of the scenario and the relevant relations between traffic participants. The features are used as inputs to a classifier trained on examples from the respective scenario. The models also contain a set of activation rules to check if the model is suitable to anticipate the behavior of a vehicle.

- A node n_i has a set of binary, mutually exclusive conditions R_i that are evaluated for each entity when node n_i is active. Each model m_i^j has a subset r_i^j of these conditions that defines when the model is active. For example, if the ego-vehicle approaches an entrance, the node for the entrance will be activated as shown in Fig.1. For each entity we determine the activated model m_i^j based on R_i. The only activated model returns the output c_i for the respective entity.
- A model $m_i^j = (\mathbf{x}_i^j, \mathbf{f}_i^j(\mathbf{x}_i^j), \mathbf{r}_i^j)$ contains:
  - A vector of features $\mathbf{x}_i^j = [x_{i,1}^j, ..., x_{i,L_i^j}^j]^T$ where L_i^j is the number of features of the vector $\mathbf{x}_i^j$ of the model m_i^j.

- A set of 2-class classifiers $\mathbf{f}_i^j(\mathbf{x}_i^j) = \{f_{i,1}^j(\mathbf{x}_i^j), ..., f_{i,C_i^j}^j(\mathbf{x}_i^j)\}^T$ where C_i^j is the number of classifiers of the model m_i^j of the node n_i.
- A set of conditions $\mathbf{r}_i^j = \{r_{i,1}^j, ..., r_{i,A_i^j}^j\}$ where A_i^j is the number of conditions which have to be true to activate the model.

B. Activation Principle

The SMT uses an activation mechanism to decide which nodes are to be activated at time t. This decision requires information about the current context, which can be obtained from different sources. One example is using GPS localization on annotated digital maps to activate their system when the ego-vehicle is approaching an exit lane, as e.g. shown in [19][20][21]. Alternatively, it is possible to acquire this information with sensory detection algorithms (e.g. highway exit detection [19] or detection of specific road environments [22][23][24]). It might be actually beneficial to use multiple sources, so as to be robust against annotation/detection errors. One could for example confirm information from the digital map by sensory detection, as it is done for current product level street sign recognition. Certain information sources might be better suited for certain cases, temporary scenarios like roadworks are easier to be covered by visual detection [25]. In cases where information is missing or ambiguous with respect to which node to activate, one could decide to only activate the parent node if its more generic context is better supported ("fall-back mechanism"). Alternatively, one could introduce an additional node covering this specific scenario or design a more detailed activation principle.

In general, the activation principle of an SMT should always be designed to ensure that at any time, there is only one node active at the same level. In addition, the activation of a given node has to always activate its parent node.

C. Competition Between Nodes

For each vehicle, the parent node n_j produces an output o_j. This is sent to the active sub-node n_i which also has its own prediction c_i produced by its models. To make the best prediction out of the two nodes, the active sub-node n_i computes its output $o_i = b_i(c_i, o_j)$ by applying its competition function b_i. Each node has its own competition function based on context and prior knowledge.

The demonstration setup presented in the following section will focus on predicting lane changes. Our current system will not predict the behavior of the predecessor of the ego-vehicle, but a future extension could also include this case.However, predicting our predecessor requires knowledge about its own predecessor, which is currently not possible with our sensor setup.

IV. SCENARIO MODEL TREE FOR HIGHWAY SCENARIO

The active path of Fig.1 presents an example of a SMT for predicting a lane change on the highway. Each child-node of node n_2 represents a scenario specific to the general highway scenario, which introduces some additional context information. The concept of the highway and entrance nodes will be detailed below, including the activation rules and the competition functions. Afterwards, we will run the subset of the architecture consisting of these two nodes on real world data to demonstrate the benefits of the SMT.

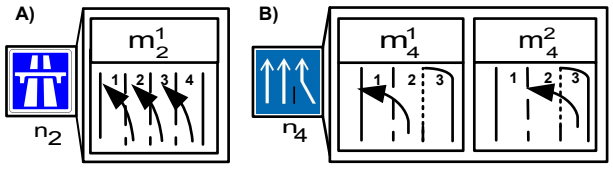


Fig. 2. (A) Highway and (B) entrance nodes of the SMT with their corresponding models. The solid arrows represent the path of other traffic participants.

In the highway node n_2, we use a generic highway model developed by [26] to predict if any of the surrounding vehicles will change lane to the left within a time horizon of two seconds. This model usually performs well on the highway even in the presence of occlusions or errors in low level perception. In an error analysis we categorized failures of this system and revealed that most errors (around 50%) can be found during specific, exceptional scenarios (see also [27]). The highway node predicts vehicles irrespective of the static environment. The behavior of a driver, in contrast, will change drastically for a short period of time while such a specific scenario occurs. Such a specific behaviors, e.g., lane changes at entrances, are stereotypic and predictable with a specific classifier.

In the highway SMT presented in this section, a node will be activated by the digital map as shown in Fig.1. Once a node is activated, it will decide which model of the node should be activated based in the lane of a vehicle. Fig.2 shows for some nodes of the SMT their corresponding models and when they are applied.

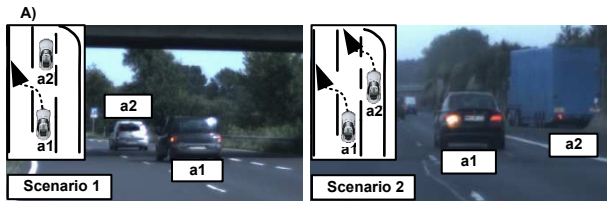
A. Highway Node

The highway node predicts if a surrounding vehicle will change lane or stay within its lane.

When driving on a highway, a vehicle can drive straight but can also change lane to the left or to the right (as shown in model m_2^1 in Fig.2.A) (Note that figures in this paper depict a right-sided traffic.). One major reason why a driver is changing lane to the left is when he is approaching a slower vehicle and wants to overtake it.

The highway node computes context based features that evaluate speed relations between a vehicle and its surrounding vehicles. These features are fed into a classifier to predict the future lane change behavior. The principal features used in this model will be detailed in section VI. For more details, the interested reader is kindly referred to [26].

The highway node provides a prediction o_2 for each surrounding vehicle which is then sent to the active child-node as shown in Fig.1.



B)

	Scenario 1		Scenario 2	
	a1	a2	a1	a2
Highway Node	Change lane	Straight	~~Straight~~	~~Straight~~
Entrance Node	~~Straight~~	Straight	Change Lane	Change Lane
Competition Results	Change Lane	Straight	Change Lane	Change Lane

Fig. 3. (A) Two entrance scenarios on the highway. Dashed arrows represent the future behavior of the vehicles. In scenario 1, $a1$ changes lane to the left because of $a2$ being too slow, and $a2$ drives straight on. In scenario 2, $a2$ enters the highway and $a1$ changes lane to the left because of $a2$. (B) Table visualizes the prediction result of the highway model, the entrance model and the competition results of both models exemplary for both scenarios. Grey fields indicate a correct prediction, crossed out field indicate wrong predictions.

B. Entrance Node

The entrance node represents an entrance scenario where, for a short period of time, there is an additional lane and an entering vehicle has to leave this lane before it ends. Fig.2.B shows an example where lane 1 and 2 represent the highway and lane 3 refers to the entrance lane.

Scenario: A vehicle driving on the right most lane (lane 3) will change lane to the left before the lane ends, even if there is dense traffic and not much free space. To find out if the vehicle will enter the highway before or after its left successor we use model m_4^2 (Fig.2.B).

It might happen that a vehicle driving in lane 2 will move to the left to free the lane ('give way') for the entering vehicle. This behavior is predicted by the model m_4^1.

The implementation of these models which will be used in our example application in section VI was discussed in detail in a previous publication see [28].

Model Conditions: Model m_4^1 is active for all vehicles driving in the left neighboring lane of the entrance lane. Model m_4^2 is active for all vehicles driving in the entrance lane. The confidence for vehicles which are not predicted by either of both models is set to 0.

Competition Function: The competition function is applied to all predicted vehicles. The prediction of the node which has the highest confidence for changing lane to the left will be used for the final vector o_4 (maximum selection).

The two models of node n_4 use the same competition function. Fig.3 displays two exemplary situations at an entrance which motivate this choice for the competition function. In scenario 1 the lane change intention of $a1$ arises from the slower vehicle $a2$. In this case, the highway node predicts a lane change with a high confidence which will win over the low confidence of the entrance node prediction. In scenario 2 the lane change intention of vehicle $a1$ is caused by $a2$ entering the highway. Here, the maximum selection of the competition function makes the entrance node override the wrong prediction of the highway node.

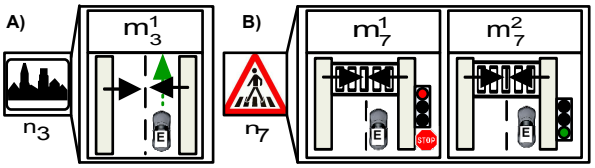


Fig. 4. (A) Inner-city and (B) Pedestrian crossing nodes of the SMT with their corresponding models. The solid arrows represent the path of other traffic participants. The dashed arrows represent the path of the ego-vehicle 'E'.

V. SCENARIO MODEL TREE FOR INNER-CITY SCENARIO

The right side of Fig.1 presents an example of a SMT for predicting the risk of collision in inner-city.

We will focus on the generic inner-city node and one of its child nodes, namely for pedestrian crossing areas (Fig.4 nodes n_3 and n_7). We will also restrict prediction to pedestrians to show that our approach is applicable to other types of traffic participants. One or more car specific models could be added inside each node of the same SMT for including car predictions.

In the inner-city SMT presented in this section, a node will be activated by the digital map as explained in the previous section. Once a node is activated, it will decide which model of the node should be activated based on the activation mechanism in order to predict if there is a risk of collision between the ego-vehicle and a pedestrian.

A. Inner-city Node

The generic inner-city scenario covers usual behaviors of pedestrians in inner-city based on physical motion with additional knowledge of the context.

Scenario: For inner-city scenarios we consider the risk of collision with a pedestrian. A pedestrian walking towards the road has an intention to eventually cross the road, whereas a pedestrian walking along the side does not. Therefore orientation and movement directions as well as distance to the road are helpful features. The pedestrians' behavior will be predicted by model m_3^1. This model could use the method proposed by [29] to predict on a short time horizon if a pedestrian will cross or not. The authors use the position of the pedestrian, the velocity and compute features such as the Time-to-Curb which is the estimated time to reach the curbstone and the Time-to-Stop which is the time needed to stop walking.

Model Conditions: Model m_3^1 is active for all pedestrians in the ego-vehicle's surrounding.

B. Pedestrian Crossing Node

This node represents a pedestrian crossing scenario where, for a short period of time, there is an area where pedestrians often cross the road and vehicles will eventually have to stop to let pedestrians cross (node n_7 in Fig.4.B). It will cover all areas where pedestrian crossing is supported, most of the time this will be a zebra crossing or a traffic light but it might also be, e.g., a traffic island. We will activate the node using a digital

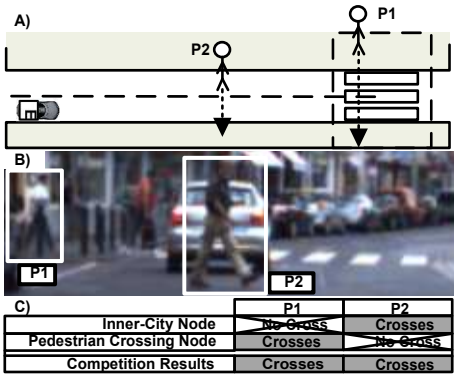


Fig. 5. (A) the ego-vehicle 'E' is approaching a zebra crossing represented by the dashed rectangle. Dashed arrows represent the future behavior of the pedestrians. $P1$ and $P2$ are pedestrians about to cross the road. (B) $P1$ and $P2$ in a real world image. (C) The prediction result of the inner-city model, the pedestrian crossing model and the competition results of both models exemplary for both pedestrians. Grey fields indicate a correct prediction, crossed out field indicate a wrong prediction.

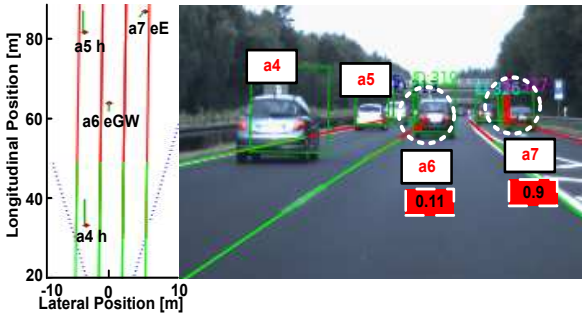


Fig. 6. The left part shows the ego-vehicle at the origin, the lane markings and the radar targets with their relative velocity. The letter 'h' denotes that the vehicle has been predicted by the model m_2^1 of the highway node, letters 'eE' and 'eGW' mean that vehicles have been predicted by model 'Entrance Enter' m_4^2 and model 'Entrance Giveway' m_4^1 which won the competition. The right part shows the detected vehicles. The white dashed rectangles outline the probability of a lane change for the vehicles driving in the left neighboring lane and the entrance lane. The probability is also represented by a red bar in the white circles. Vehicle $a7$ will change lane to the left with a high probability and $a6$ will change lane with a low probability.

map but there are also existing computer vision approaches for the detection of those structures [30], [31] that could be used alternatively. The features used by the pedestrian crossing node are similar to the ones in the inner-city node, however, the conditional probabilities learned by the classifier are different. A pedestrian standing at the side of the road is more likely to cross in the vicinity of a zebra crossing than in other areas without pedestrian crossing.

Scenario: Pedestrian crossings can be separated into two different scenarios: those scenarios where pedestrians have priority, e.g., at zebra crossings or green traffic lights, and those where pedestrians do not have priority, e.g., at traffic islands or red traffic lights. The first group of scenarios is predicted using model m_7^1, the second by using m_7^2. To our knowledge, no existing model targets at predicting pedestrians at crossing areas. Such a scenario can be covered by using simple classifiers, similar to what we did for the entrance node, by defining features such as the Time-to-CrossArea, which is the time the pedestrian needs to enter the crossing area (see dashed rectangle in Fig.5), the motion direction, to know if the pedestrian is walking in direction of the crossing area or the distance to the nearest vehicle driving on the road, to predict whether or not the pedestrian will cross.

Model Conditions: The activation rule will decide whether or not a pedestrian has the priority to cross the road and choose the correct model accordingly. This node is active when the ego-vehicle is driving on a normal road (not at intersection) to predict pedestrians while approaching a pedestrian crossing area.

Competition Function: If the pedestrian crossing node is active, a competition function is applied between it and the inner-city node n_3 for the prediction of all detected pedestrians. The final prediction will be that of the node with the highest confidence for crossing the street. Fig. 5 displays an exemplary zebra crossing scenario which motivates this choice for the competition function.

In this scenario, the intention of the two pedestrians to cross the road can be inferred. $P2$ is already on the road walking toward the ego-lane. The inner-city model predicts that $P2$ is going to intersect the path of the ego-vehicle. However, $P2$ is very far from the zebra crossing area and not walking towards it, so the pedestrian crossing model predicts that $P2$ is not going to cross. $P1$ who is very far from the road is not going to cross according to the inner-city model, but the pedestrian crossing model disagrees because it knows that the pedestrian is approaching the zebra area to cross the road. The final prediction will be the output of the maximum function applied to the result of the inner-city node and the output of the pedestrian crossing model.

VI. EXPERIMENTS

In this section we will demonstrate the usefulness of our approach using an exemplary subset of the described highway SMT consisting of the highway node n_2 (with the model m_2^1) and the entrance node n_4 (with the models m_4^1 and m_4^2) tested to predict lane changes to the left on the highway.

All tests have been done using real data recorded on several German highways (curved and straight) under varying weather conditions (overcast, rain, fog, low sun) and traffic conditions (light, medium, and dense traffic). The vehicle provides us with radar data and camera images. We use a fusion algorithm [32] to extract vehicle position and velocity as well as lane markings (as shown in Fig.6) with a frame rate of 10Hz. As outlined earlier, the mechanism of node activation is based on the ego-vehicle location and the current contextual environment. We use a digital map to localize the ego-vehicle in its environment and get the relevant information about the scenario. The static infrastructure of the road, e.g. number and spatial alignment of lanes and the start and end of an entrance, has been annotated manually. We used this to focus the evaluation on the quality of the prediction rather than that of the sensory systems.

We use an event based evaluation in order to show how the system would interact with a driver. We will show that our SMT produces fewer errors than the highway model alone. In

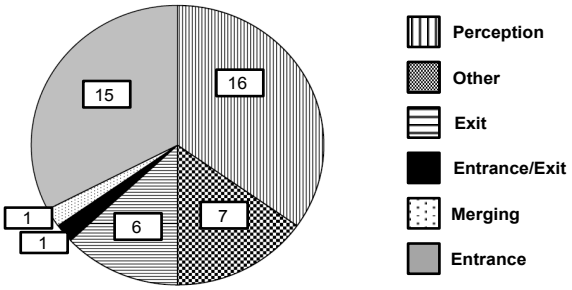


Fig. 7. Prediction errors produced by the highway model alone classified into 6 categories. The focus of these experiments is to reduce errors at entrances.

the following we compare the performances of the highway model alone to the SMT.

A. Dataset Creation

We use a supervised learning method to predict behaviors, therefore we need to build one dataset which is used for training the classifiers and one dataset which is used only for testing. To train and test a classifier we need to compute these features and provide the ground truth at each timestep. That means each timestep has a corresponding annotation as a negative or positive example. On the highway we are interested in predicting if a vehicle will change lane to the left. A vehicle is considered as changing lane when its right wheel touches the lane marking. When a lane change is detected, each timestep from that moment to three seconds before is annotated as a positive example. Two seconds before and two seconds after the lane change are ignored in the evaluation because there is no obvious start or end point for a prediction, everything else is annotated as negative examples.

The dataset used for this evaluation consists of 4 hours of driving with 783 vehicles, 40 of them perform a lane change on the highway and 33 do a lane change at entrances. To evaluate the effect of the prediction system on some ADAS function, we restrict prediction to vehicles within a certain distance, respectively within a certain time-to-collision window. Vehicles that have a time-to-collision (distance divided by velocity difference) larger than 10s or a time-gap (distance divided by ego velocity) larger than 2s will not be predicted, because they do not represent a danger for the ego-vehicle. This leads to a dataset of 695 vehicles with 29 lane change maneuvers on the highway and 17 at entrances.

B. Highway Model Implementation

In this evaluation, the highway node is always active. This guarantees meaningful predictions in case of digital map failures. This node contains a model capable of predicting if the right predecessor (RP) a_w of the ego-vehicle is going to perform a lane change to the left or drive straight depending on its predecessor (P). This model uses features like the time-to-collision (TTC) to the predecessor, the size of the left gap and the time needed to enter the left gap as show in Table I (features 1-5). The model uses a single layer perceptron to predict if a right predecessor of the ego-vehicle is going to change lane.

TABLE I
FEATURE DEFINITION

1	TTC	Longitudinal Time-to-Collision between two vehicles a_w and $a_{surrounding}$
2	TTG	Time-to-Gap is the time of a_w to reach the left gap between its left successor $(LS)_w$ and left predecessor $(LP)_w$
3	DiffVelocity	Difference of velocity between a_w and $a_{surrounding}$
4	Size-Left-Gap	Indicator to know if the vehicle fits into the left gap according to the length of the gap and the distance to the gap
5	Accelerate-to-Gap	Is Acceleration needed to enter the left gap
6	PrevPredGW	Previous prediction of the SMT for the vehicle on the left next lane
7	TTE	Time-to-End is the time of a_w before reaching the end of the entrance
8	VisCoeff	Coefficient of Visibility is used to approximate the percentage of the entrance lane hidden by other vehicles
9	PrevPredE	Previous prediction of the SMT for the vehicle on the entrance lane

We evaluate the highway model for the best threshold that guarantees a maximum of 4 false positives per hour. Fig.7 shows an error analysis of this model. For each vehicle predicted by the highway model we counted the false positives and false negatives for a threshold of 0.75 and categorized them with respect to their causes. This figure shows that many errors arise from perception (e.g. wrong lane assignments or limited sensor range), but also that a large proportion are caused by specific situations. The category 'Other' contains errors that cannot be attributed to systematic failures. In the following, the paper focuses on reducing errors in the entrance scenario to demonstrate that such structural errors can be addressed by the proposed framework.

C. Entrance Model Implementations

When a model is activated, it will compute the features that are the inputs to its classifier. In this section we will detail the features used for the models 'Entrance Enter' and 'Entrance Giveway' of the entrance node. For more details on the formulas see [28].

1) Features 'Entrance Enter' Model:

This model becomes active if the entrance node is activated and a vehicle a_w is driving in the entrance lane. To illustrate this model, Fig.6 presents an example of an entrance scene perceived by the ego-vehicle, driving on a highway and approaching an entrance. Vehicle $a7$ is in the entrance lane and will have to change lane to the left. The features 1-7 of Table I are designed to answer the question: will vehicle a_w driving in the entrance lane enter the highway before or after its LS?

2) Features 'Entrance Giveway' Model: This model becomes active if the entrance node is activated and a vehicle a_w is driving in the left neighboring lane. The features 1-6, 8 and 9 are designed to answer the question: will vehicle a_w driving on left of the entrance lane change lane to the left because of its RP and can it change lane to the left according to the free space and the other vehicles on its left?

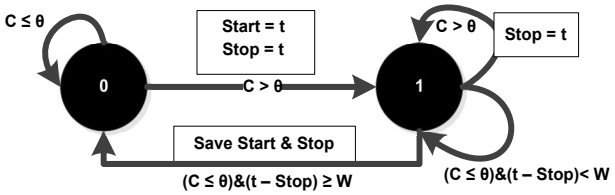


Fig. 8. This state machine is used to count the number of events according to different thresholds θ. States are black circles and transitions are black lines connecting the states with each other. C is the confidence and θ the threshold. The white rectangles represent the different actions done during the different transitions. The current timestep is symbolized by letter t and W is a time window.

3) Feature Normalization: To fit the characteristics of a single layer perceptron, all features are scaled to $[0.1; 0.9]$. A feature like the TTC will be within $[-inf, +inf]$, but after a manual inspection of the values, we observed that the TTC of a vehicle which does change lane is always positive and below 15s. For that reason, it makes sense to only preserve the variability of those values inside this range. To define this range, a minimum and a maximum threshold has to be defined for each feature. After defining thresholds, we choose a ii function to scale the features because it limits the impact of the choice of the thresholds compared to ,e.g., a piecewise linear function:

$$\text{fermi}(x, min, max) = 1/(\exp((x - \mu)/k) + 1) \quad (1)$$

The parameter $\mu = min + (max - min)/2.0$ is used to center the results in the interval and the parameter $k = (max - \mu)/log(1/0.9 - 1)$ changes the slope of the curve.

4) Competition Function: According to the lane the RP of the ego-vehicle is driving in, one or the other model of the entrance node becomes active. The active model returns a confidence value c_4 between $[0; 1]$. The entrance node n_4 applies the maximum function between the confidence value o_2 of the highway node n_2 and the confidence of the active model of the entrance node to get the final prediction $o_4 = max(o_2, c_4)$.

D. Event Based Evaluation

When building a system, it is important to consider user's acceptance. As our system should warn the driver when there is a risk of collision with a right predecessor (RP) that wants to enter its lane, we want to know how often the system will send an incorrect warning to the driver. In contrast to the commonly applied frame-based evaluation, an event-based evaluation considers the temporal grouping of system signals and the relevance of a predicted vehicle for the driver of the ego vehicle. This evaluation consists of two parts. First we count events, which are the number of warnings, and then we count the number of wrong warnings, when a driver is warned for no reason. For each vehicle at each time t the system provides a confidence value for a lane change.

1) Grouping System Signals To Events: An event starts for one vehicle whenever its 'lane change left' confidence C exceeds a threshold θ and an event finishes when it falls below θ.

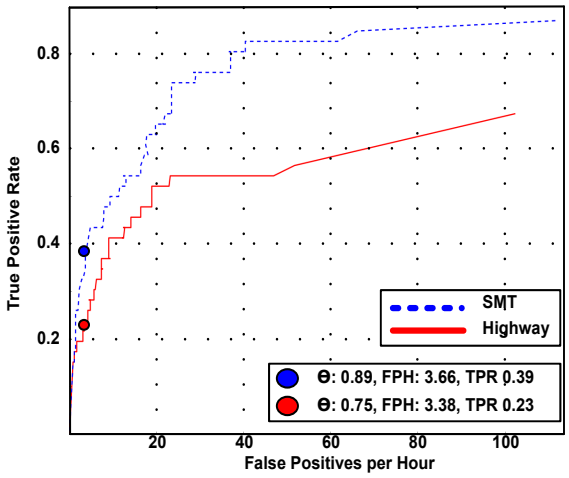


Fig. 9. True Positive Rate against False Positives per Hour for predicting that the right predecessor of the ego-vehicle will change lane to the left. The solid line represents the highway model alone. The dashed line represents the SMT. The two points show the best FP/H for both curves according to our quality criterion and their corresponding threshold $theta$.

To count the events, we use the state machine presented in Fig.8. A new state machine is initialized with state 0 for each predicted vehicle. For each time t, the confidence value C is compared to the threshold θ. If the confidence is above the threshold, the current time t is saved in the variables 'start' and 'stop' and state 1 is active. While the confidences are above the threshold they are considered to belong to the same event. As soon as a confidence value is below the threshold, state 1 counts the number of confidences which are below the threshold. If the number of confidences below the threshold reaches the limit value W, the variables 'start' and 'stop' are saved and state 0 is active. Otherwise it stays in state 1 until the event ends.

A wrong choice of W could lead to multiple events for a single lane change which would trigger multiple successive warnings. Therefore it is important to find the time window W within which we will fuse multiple events. In this paper we use $W = 1s$, which showed the best trade off between reducing the number of warnings and still allowing for the correct behavior in cases where a vehicle changes lane multiple times.

2) Classifying An Event: An event is a false positive (FP) when the system sends a warning but the right predecessor does not change lane to the left within the following three seconds. An event is a true positive (TP) when the system sends a warning and the right predecessor changes lane within the next three seconds. We plot the true positive rate (TPR) against the number of FP/H for a number of different values of the threshold θ.

3) Results: Fig.9[1] presents the TPR against the FP/H for the highway model alone and for the SMT. We observe that the SMT generally produces fewer errors than the highway model. The ROC curve of the SMT shows its best quality for a threshold of 0.89 with a FP/H of 3.66. On the highway, the

[1]The roc curve shows small artifacts because the grouping of events (the number) change with threshold.

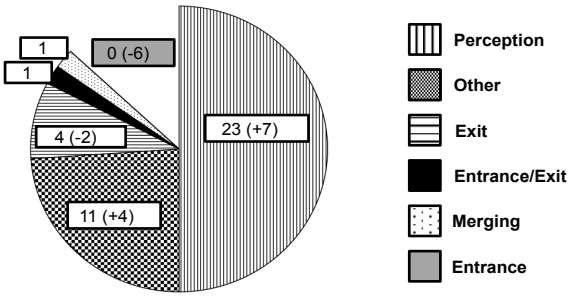


Fig. 10. Prediction errors produced by the SMT classified into 6 categories. Then numbers in brackets are the difference to Fig.7.

TABLE II
CONFUSION MATRIX

	Highway	Entrance Enter	Entrance Giveway	Total
TP (Highway)	151	0	0	151
TP (SMT)	113 (-38)	105 (+105)	20 (+20)	238 (+87)
FP (Highway)	149	0	0	149
FP (SMT)	54 (-95)	29 (+29)	8 (+8)	91 (-58)
FN (Highway)	683	173	100	956
FN (SMT)	722 (+39)	67 (-106)	80 (-20)	869 (-87)
TN (Highway)	33213	294	3366	36874
TN (SMT)	33304 (+91)	265 (-29)	3363 (-3)	36932 (+58)

best FP/H of 3.38 is obtained for a θ of 0.75.

Fig.10 shows an error analysis of the SMT at threshold 0.89. The 15 errors produced by the highway model alone due to the specificity of the entrance scenario have disappeared (see Fig.7). A number of events (6) are now predicted correctly by the SMT. However, the perception errors have slightly increased, which is due to the fact that the SMT uses additional sensor information. We also found a few cases that cannot be predicted well by the SMT ('Other'), mostly caused by non typical behaviors of the traffic participants. Some false positives caused by the exit scenario have disappeared, because the SMT allows us to work with a higher threshold since the combination of specific models results in higher prediction confidence.

Our event-based analysis confirms that a combination of specific classifiers produces more accurate predictions and fewer errors compared to using only a single classifier.

E. Time Horizon Evaluation

As mentioned above, we labeled a window of three seconds before our right predecessor has moved into our lane for the prediction target.

To evaluate the prediction horizon, we averaged the earliest points of lane change prediction over all predicted vehicles at various thresholds. The resulting prediction horizon is plotted in Fig.11. This plot shows a general decrease of prediction horizon with increasing threshold.

The observable saw-tooth pattern is due to the fact that the number of vehicles predicted decreases while the threshold increases.

In the plot, we highlighted the thresholds of equal False Positive Rate where the highway model alone performs slightly better than the SMT. However, on average, the SMT achieves a higher prediction horizon than the highway model alone.

F. Model Activation Robustness

In our current system, the entrance node of the SMT becomes activated via a digital map when the ego-vehicle is approaching an entrance scenario. The start and end of the entrance have been manually annotated. In future work, the manual annotation should be replaced by a commercial digital map. The accuracy of the GPS to get the position of the ego-vehicle and the accuracy of the digital map influence the performance of our system. We evaluated the effect of an inaccurate digital map or GPS signal by adding noise, varying the start and the end of an entrance by +/-10 meters from the original annotations [33]. We observed a reduction of the TPR (from 0.46 to 0.42) when the entrance node is activated. We also observed that the FP/H of 9.87 slightly increases for all variations to 10.44. These numbers show that the system seems to be quite robust with respect to realistic activation noise.

G. Frame-Based Analysis

We also evaluated our system using a frame-based evaluation. Table II shows the number of TP, FP, true negative (TN) and false negative (FN) produced by the highway model alone and the SMT for their given thresholds (0.75 and 0.89).

The highway model alone predicts the right predecessor independent of its lane (this includes entrance and left neighboring lane). While most of the FP are caused by perception, most of the FN happen because the highway model cannot predict vehicles at entrance scenarios (as shown in the error analysis Fig. 7).

Table II shows that the SMT produces more TP and fewer FN compared to the highway model. The majority of improvements can be seen for predictions in the entrance scenario. We observe two additional effects: the number of FP on the highway decreases because we can now set a higher threshold without changing our quality criterion (see Fig. 9). The second effect is that the number of FP at entrances increases, which is due to the fact that the entrance models suffer from an additional type of perception error. However, the positive effects are more prominent.

H. Computation Time Analysis

The generic computational complexity of the SMT depends on the number of nodes n, the number of vehicles v detected and the number of neighboring vehicles s of a vehicle but not on the number of models since our activation rules guarantee that only one model is active per n and v. This means $O(n * v * s)$.

For our highway system we have a constant number of nodes and the maximum number of surrounding vehicles that are considered is three. Therefore the system's complexity is only $O(v)$. On average, the highway node has a computation time of 20 ms and the entrance node has a computation time

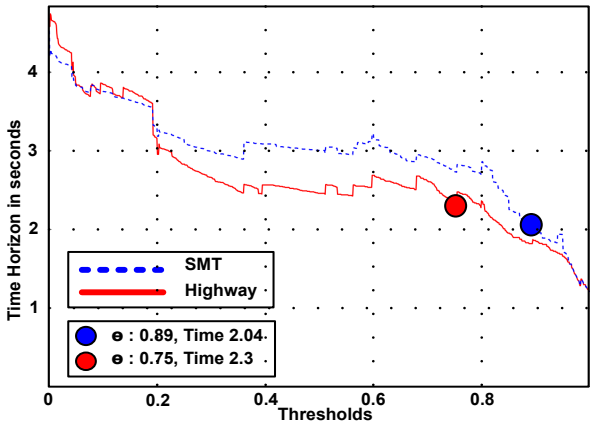


Fig. 11. Average prediction horizon for lane changes according to threshold θ. The solid line represents the highway model. The dashed line represents the SMT. The points represent the time horizon at the best threshold according to our quality criterion.

of 4 ms on a single core of a Xeon X5550 (2.6GHz), 8 GB RAM. The overall computation time will be between that of the highway node alone and the combination of both nodes (24ms).

VII. CONCLUSION AND OUTLOOK

In this paper, we outlined that none of the existing work is suited for behavior prediction with both, a wide scope and a high quality. We proposed a generic concept based on SMT, a tree like structure that enables it to predict the behavior of surrounding vehicles for a large variety of scenarios. This concept allows creating context-based models of those scenarios that cannot be covered by generic ones. These models are ordered into the SMT according to their context specificity. Finally, competition rules between nodes are applied in order to obtain the most accurate prediction. This structure allows combining specific models with high quality but narrow scope to one overall system featuring high quality and a broad scope.

The SMT has been evaluated exemplary for lane change prediction in different highway scenarios. These experiments showed that the system could predict behaviors two seconds in advance.

Comparing the SMT to existing work on lane change prediction for general highway situations allows for two conclusions.

First, the SMT is a suited approach to improve the prediction quality, because it increases the ratio of correct predictions and decreases the ratio of wrong predictions. This is especially valid for structural errors caused by scenario specific behaviors. We thus expect a further performance increase by adding additional scenario specific models.

Secondly, the increased specificity of models used in the SMT can pose additional requirements on the perception system. For example, we need to know when an entrance zone starts. In our experiments we used a combination of digital maps and GPS. As is true for most sensors, GPS is not always perfect. But, the structure of the SMT ensures the existence of a prediction, even in cases where sensory information is missing or erroneous. Our experiments also showed the robustness of the system against localization noise.

A potential ADAS application of our system would be an improved advanced cruise control (ACC) system. Using the prediction of a lane change maneuver of our right predecessor would allow us to maintain a safe distance before it enters our lane, and therefore smoothen the deceleration of the ego-vehicle. We constructed our experimental setup with such an application in mind, for example by only evaluating those cars that would enter our lane below the safety distance. Tests were done on German highways because they can be seen as a worst-case for ACC due to the high range of relative speeds. However, further steps towards a product would probably require additional adaptations. It might be necessary to include more scenarios, to adapt models to e.g. driving style in different countries, or tune the activation function to a given set of sensors. The framework proposed in this paper was designed to allow such flexibility when building systems with a high quality and a wide scope.

ACKNOWLEDGMENT

The authors gratefully acknowledge the support of G. Endicott, M. Kleinehagenbrock and the reviewers of this paper for their important feedback.

REFERENCES

[1] M. M. Peden, R. Scurfield, D. Sleet, D. Mohan, A. A. Hyder, E. Jarawan, and C. D. Mathers, "World report on road traffic injury prevention," 2004.

[2] J. R. Treat, N. J. Castellan, R. L. Stansifer, R. E. Mayer, R. D. Hume, D. Shinar, S. T. McDonald, and N. S. Tumbas, *Tri-level Study of the Causes of Traffic Accidents*, 1977.

[3] J. I. Herrero Zarzosa, M. E. G. Biraud, S. Damiani, J. P. Magny, J. Merino, and E. Koren, "REPOSIT: Relative positioning for collision avoidance systems," Tech. Rep., 2007.

[4] C. Grover, I. Knight, F. Okoro, I. Simmons, G. Couper, P. Massie, B. Smith *et al.*, "Automated emergency brake systems: Technical requirements, costs and benefits," *Automated emergency brake systems: technical requirements, costs and benefits*, pp. 1–109, 2013.

[5] D. Meyer-Delius, J. Sturm, and W. Burgard, "Regression-based online situation recognition for vehicular traffic scenarios," in *Proc. IEEE Intelligent Robots and Systems*, 2009, pp. 1711–1716.

[6] M. Platho and J. Eggert, "Deciding what to inspect first: Incremental situation assessment based on information gain," in *Proc. IEEE Intelligent Transportation Systems*, 2012, pp. 888–893.

[7] T. Gindele, S. Brechtel, and R. Dillmann, "A probabilistic model for estimating driver behaviors and vehicle trajectories in traffic environments," in *Proc. IEEE Intelligent Transportation Systems*, 2010, pp. 1625–1631.

[8] M. Tsogas, X. Dai, G. Thomaidis, P. Lytrivis, and A. Amditis, "Detection of maneuvers using evidence theory," in *Proc. IEEE Intelligent Vehicles Symposium*, 2008, pp. 126–131.

[9] I. Dagli, G. Breuel, H. Schittenhelm, and A. Schanz, "Cutting-in vehicle recognition for ACC systems-towards feasible situation analysis methodologies," in *Proc. IEEE Intelligent Vehicles Symposium*, 2004, pp. 925–930.

[10] M. Garcia Ortiz, F. Kummert, and J. Schmüdderich, "Prediction of driver behavior on a limited ssensory setting," in *Proc. IEEE Intelligent Transportation Systems*, 2012, pp. 638–643.

[11] D. Kasper, G. Weidl, T. Dang, G. Breuel, A. Tamke, A. Wedel, and W. Rosenstiel, "Object-oriented bayesian networks for detection of lane change maneuvers," *Intelligent Transportation Systems Magazine, IEEE*, pp. 19–31, 2012.

[12] R. S. Tomar and S. Verma, "Safety of lane change maneuver through a priori prediction of trajectory using neural networks," *Network Protocols and Algorithms*, pp. 4–21, 2012.

[13] M. Reichel, M. Botsch, R. Rauschecker, K. H. Siedersberger, and M. Maurer, "Situation aspect modelling and classification using the scenario based random forest algorithm for convoy merging situations," in *Proc. IEEE Intelligent Transportation Systems*, 2010, pp. 360–366.

[14] S. Lefèvre, C. Laugier, and J. Ibañez-Guzmán, "Exploiting map information for driver intention estimation at road intersections," in *Proc. IEEE Intelligent Vehicles Symposium*, 2011, pp. 583–588.
[15] M. Campbell, M. Egerstedt, J. P. How, and R. M. Murray, "Autonomous driving in urban environments: approaches, lessons and challenges," *Philosophical Transactions of the Royal Society A: Mathematical,Physical and Engineering Sciences*, pp. 4649–4672, 2010.
[16] D. Ferguson, T. M. Howard, and M. Likhachev, "Motion planning in urban environments: Part ii," in *Proc. IEEE Intelligent Robots and Systems*, 2008, pp. 61–89.
[17] R. Graf, H. Deusch, M. Fritzsche, and K. Dietmayer, "A learning concept for behavior prediction in traffic situations," in *Proc. IEEE Intelligent Vehicles Symposium*, 2013, pp. 672–677.
[18] H. Berndt, J. Emmert, and K. Dietmayer, "Continuous driver intention recognition with hidden markov models," in *Proc. IEEE Intelligent Transportation Systems*, 2008, pp. 1189–1194.
[19] S. Hold, S. Görmer, A. Kummert, M. Meuter, and S. Müller-Schneiders, "ELA - an exit lane assistant for adaptive cruise control and navigation systems," in *Proc. IEEE Intelligent Transportation Systems*, 2010, pp. 629–634.
[20] C. Wang, Z. Hu, and R. Chapuis, "Predictive lane detection by interaction with digital road map," *Journal of Information Processing*, pp. 287–296, 2012.
[21] J. C. Mccall, D. Wipf, M. M. Trivedi, and B. Rao, "Lane change intent analysis using robust operators and sparse bayesian learning," in *Proc. IEEE Computer Vision and Pattern Recognition*, 2005, pp. 431–440.
[22] K. Takagi, K. Morikawa, T. Ogawa, and M. Saburi, "Road environment recognition using on-vehicle lidar," in *Proc. IEEE Intelligent Vehicles Symposium*, 2006, pp. 120–125.
[23] M. Thuy and F. P. León, "Lane detection and tracking based on lidar data," *Metrology and Measurement Systems*, pp. 311–322, 2010.
[24] T. Kuehnl, F. Kummert, and J. Fritsch, "Visual ego-vehicle lane assignment using spatial ray features," in *Proc. IEEE Intelligent Vehicles Symposium*, 2013, pp. 1101–1106.
[25] B. Mathibela, M. A. Osborne, I. Posner, and P. Newman, "Can priors be trusted? Learning to anticipate roadworks," in *Proc. IEEE Intelligent Transportation Systems*, 2012, pp. 927–932.
[26] J. Schmüdderich and S. Rebhan, "A method and system for predicting movement behavior of a target traffic object," Patent EP 11 178 252.0.
[27] S. Bonnin, T. H. Weisswange, F. Kummert, and J. Schmüdderich, "Accurate behavior prediction on highways based on a systematic combination of classifiers," in *Proc. IEEE Intelligent Vehicles Symposium*, 2013, pp. 242–249.
[28] S. Bonnin, F. Kummert, and J. Schmüdderich, "A generic concept of a system for predicting driving behaviors," in *Proc. IEEE Intelligent Transportation Systems*, 2012, pp. 1803–1808.
[29] C. Keller, C. Hermes, and D. Gavrila, "Will the pedestrian cross?: Probabilistic path prediction based on learned motion features," in *Proc. Pattern recognition*, 2011, pp. 386–395.
[30] R. de Charette and F. Nashashibi, "Real time visual traffic lights recognition based on spot light detection and adaptive traffic lights templates," in *Proc. IEEE Intelligent Vehicles Symposium*, 2009, pp. 358–363.
[31] S. Se, "Zebra-crossing detection for the partially sighted," in *Proc. IEEE Computer Vision and Pattern Recognition*, 2000, pp. 211–217.
[32] M. Nishigaki, S. Rebhan, and N. Einecke, "Vision-based lateral position improvement of radar detections," in *Proc. IEEE Intelligent Transportation Systems*, 2012, pp. 90–97.
[33] M. Donath, P. Cheng, S. Shekhar, and X. Ma, "A new approach to assessing road user charges: Evaluation of core technologies," Tech. Rep., 2003.



Sarah Bonnin received the master degree in Intelligent Systems from Paul Sabathier University, Toulouse, France, in 2010 after doing her master thesis at Honda Research Institute Europe GmbH (HRI EU) at Offenbach Am Main, Germany.
Since 2010 she is a Ph.D. student at the Research Institute for Cognition and Robotics at Bielefeld University, Germany working in collaboration with HRI.
Her research interests are situation understanding and behavior prediction in the automotive domain.



Thomas H. Weisswange received the Dipl.-Bioinf. degree in bioinformatics from Goethe University Frankfurt, Germany, in 2008. In 2008 he joined the Computational Neuroscience group of Prof. Jochen Triesch at the Frankfurt Institute for Advanced Studies in Frankfurt, Germany. He received a Ph.D. in computer science from Goethe University Frankfurt, Germany in 2012.
Since 2011 he has worked as a Senior Scientist at the Honda Research Institute Europe GmbH in Offenbach, Germany.
His current research interests include methods for spatial representations, situation recognition, and cognitive system concepts for intelligent automotive systems.



Franz Kummert received the diploma and the Ph.D. (Dr.-Ing.) degree in computer science from the University of Erlangen-Nürnberg, Erlangen, Germany, in 1987 and 1991, respectively. In 1996 he received the venia legendi (Habilitation) in computer science from the Faculty of Technology of Bielefeld University, Germany. Since 2002 he is an apl. professor for pattern recognition. From 1987 to 1990 he worked at the research group for Pattern Recognition (Institut für Informatik, Mustererkennung) at the University of Erlangen-Nürnberg, Erlangen, Germany.
Since 1991, he is with the research group Applied Informatics (Angewandte Informatik) at Bielefeld University, Germany. Since 2003 he is the dean of studies of the Faculty of Technology, and from 2004 to 2010 he was a member of the Senat of Bielefeld University.
His fields of research are speech and image understanding and human-robot interaction. He has published various papers in these fields, and is author of a book on the control of a speech understanding system and on the automatic interpretation of speech and image signals. Dr. Kummert is Member of the Institute of Electrical and Electronics Engineers (IEEE).



Jens Schmuedderich received the Diploma and Ph.D. (Dr.Ing.) degree in computer science from Bielefeld University, Germany, in 2006 and 2010, respectively.
From 2006 to 2008 he worked at the Research Institute for Cognition and Robotics at Bielefeld University.
Since 2009, he has been with the Honda Research Institute Europe GmbH Offenbach, Germany, where he is currently working as a Project Manager in the area of Advanced Driver Assistance Systems.
His research interests are situation understanding and behavior prediction in the automotive domain.