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

Accurate Behavior Prediction on Highways Based on a Systematic Combination of Classifiers

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

2013

Preprint:

This is an accepted article published in Intelligent Vehicles Symposium (IV). The final authenticated version is available online at: https://doi.org/[DOI not available]

2013 IEEE Intelligent Vehicles Symposium (IV)
June 23-26, 2013, Gold Coast, Australia

Accurate Behavior Prediction on Highways Based on a Systematic Combination of Classifiers

Sarah Bonnin[1], Thomas H. Weisswange[3], Franz Kummert[2], and Jens Schmüdderich[3]

Abstract— **To drive safely, a good driver observes his surroundings, anticipates the actions of other traffic participants and then decides for a maneuver. But if a driver is inattentive or overloaded he may fail to include some relevant information. This can then lead to wrong decisions and potentially result in an accident. In order to assist a driver in his decision making, Advanced Driver Assistance Systems (ADAS) are becoming more and more popular in commercial cars. The quality of these existing systems compared to an experienced driver is weak, because they rely purely on physical observation and thus react shortly before an accident. For an earlier warning of the driver behavior prediction is used. We classify existing research in this area with respect to two aspects: quality and scope. Quality means the ability to warn a driver early before a dangerous situation. Scope means the diversity of scenes in which the approach can work. In general we see two tendencies, methods targeting for broad scope but having low quality and those targeting for narrow scope but high quality. Our goal is to have a system with high quality and wide scope. To achieve this, we propose a system that combines classifiers to predict behaviors for many scenarios. To show that a combination of general and specific classifiers is a solution to improve quality and scope, this paper will introduce the generic concept of our system followed by a concrete implementation for lane change prediction for highway scenarios.**

I. INTRODUCTION

Driving is a complex task and causes many accidents and fatalities every year [16]. The majority of accidents are caused by human error. Naturally, a good driver perceives his environment and anticipates the future actions of his surroundings in order to decide for a maneuver. But depending on his concentration or traffic density, his perception can be erroneous and mislead his decision-making.

In order to improve safety on roads, car companies started to equipped vehicles with Advanced Driver Assistance Systems (ADAS). Such systems are mostly reactive, this means they will warn a driver shortly after having detected a dangerous situation, which gives the driver little time to react. A human driver in contrast will be able to anticipate a dangerous situation before it becomes apparent.

To provide a driver with enough time to react to a situation, one idea is to use behavior prediction. Thanks to today's technology, many vehicles are equipped with multiple sensors and means to perceive the environment and localize other traffic participants. Because of this, it is possible to reconstruct a scene sufficiently precise to allow for behavior prediction.

In the last ten years, behavior prediction has become a popular topic of research. Several studies about the prediction of behaviors exist focusing on scope or quality to varying extends.

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 that he has time to react. The basic way of predicting behaviors is to use physical prediction. The principle is to predict the next state based on the current one [6], [7], [14], [19]. These methods are accurate to up to one second into the future, but are not valid for a longer time horizon. A driver needs to be warned at least two seconds in advance to have a chance to react in time. The benefit of those systems is, that they are usually working in a large variety of scenarios. One could say that these methods trade of prediction quality at longer time horizons for having a broad scope. Alternatively one could say they limit the complexity of a predicted behavior to a minimum in order to reach high prediction accuracy.

There are other methods, which perform more complex behavior prediction for up to 3 seconds into the future [2], [4], [15]. The authors of [15] propose a high quality method to predict ego-vehicle lane changes on the highway based on head tracking. Such information is not usable to predict other vehicles lane change. Surrounding vehicles could be predicted with the approach proposed in [4]. However the authors show that the approach is limited to a generic highway scenario. To be able to predict behaviors early enough, it appears that the scope is reducing.

Some approaches are designed to predict a single behavior for some specific scenarios which occurs occasionally on the highway or in inner-city [9], [11], [17]. For example a merging scenario [17] or an exit scenario [9] represent only a small fraction of traveling time on the highway, but they are a source of accidents. Such scenarios cannot be covered by an approaches dedicated to predict behaviors in generic scenarios [4].

The methods described in the last two paragraphs differ from a physical prediction approach because they model the scene. Based on context and prior knowledge, they define some specific features in order to predict one behavior.

From these examples we observe that a specific classifier performs well for a specific scenario, but two specific classifiers combined together could perform better for a wider scope. This outlines the relevance of combining classifiers

S. Bonnin is with the Research Institute for Cognition and Robotics, Bielefeld University, Bielefeld, Germany sarah.bonnin@gmail.com

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

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

in order to improve prediction results and the number of predictable scenes.

The related work shows that generic methods predict behaviors with low quality but a wide scope, and more specific methods can predict behaviors for more complex scenarios with a high quality but a narrow scope. But no current existing system proposed a high quality with a wide scope. We believe that by combining multiple specific classifiers in a meaningful way, we can get the best of both worlds.

It is challenging to find a good way of combining these classifiers with each other. In this work we will present such a general framework and then demonstrate its usefulness by applying it to build a system for behavior prediction on the highway.

In this paper we present the formal concept of the hierarchy system and its application to highway scenarios. To illustrate it, we implement and evaluate an example system consisting of the highway node and an entrance node in order to show that a combination between classifiers gives better results than two classifiers separately. We also propose a conceptual extension of the hierarchy to additional scenarios specific to highways.

Some approaches like [5][10] proposed a model based on a hierarchical structure to recognize a driving maneuver. They start from the sensor level and each level of the hierarchy is used to reconstruct the scene by relating the vehicles to the lanes and to other traffic participants. The highest model of the hierarchy is used for the behavior prediction. In our hierarchy, each model will have the same function, predicting behaviors. The level of abstraction will be based on the location specificity.

The paper is organized as follows: Section II gives an overview of our approach, section III introduces the concept of a hierarchy of classifiers, then section IV describes the highway instance of such a hierarchy and details the specific scene models that will be used. Section V illustrates our concept with experiments on real data based on the combination between specific classifiers. Finally section VI gives a conclusion and presents the future work.

II. OVERVIEW OF THE APPROACH

The idea for predicting behaviors on a highway is to have a generic method constantly active to predict behaviors of surrounding vehicles as long as the ego-vehicle is on a highway.

From time to time, this generic highway model makes prediction errors. These errors happen when the scene is changing, the scenario is becoming too specific for the generic model. Such scenarios like entrances or exits, oblige the vehicles to adopt different behaviors, following rules that the highway model does not know about. For these scenes it is necessary to define a specific model to be able to predict behaviors. The idea is to model each scene that has a different context separately. Then depending on ego-vehicle location, the corresponding model will be activated and will be combined with the results of the generic method in order to improve the accuracy. Fig.1 shows a combination between the highway model and the entrance model when the ego-vehicle is approaching an entrance, or a combination between the highway model and the exit model when the ego-vehicle is approaching an exit. To ensure an improvement of the prediction accuracy based on the combination between classifiers, our idea is to create a hierarchy of scene models.

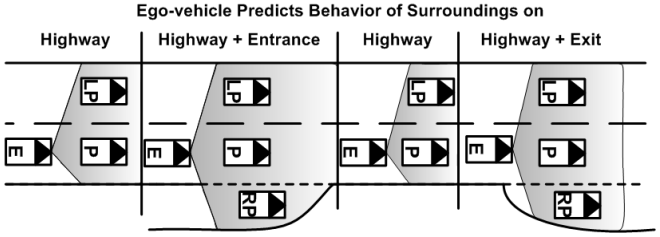


Fig. 1. Ego-vehicle 'E' predicts the behavior of the right predecessor (RP), left predecessor (LP) and predecessor (P) on highway. A generic 'Highway Model' is constantly active to predict behaviors on highway. When approaching a specific situation, a specific model will be active: 'Entrance Model' or 'Exit Model' to predict behaviors while the ego-vehicle is in the related area. The results of both models are then combined to improve quality and scope.

A hierarchy based structure is a suitable solution to order scenes depending on location specificity. We decided that only one node of a same level can be active at a time. The generic scene is in the top node and the most specific ones in the leaves.

A node is activated depending on ego-vehicle location e.g. while passing a highway entrance. Specific models are important to tackle specific scenarios the generic node does not know about and will overwrite the results of the top node in order to improve the prediction.

The next section will explain the formal concept of the hierarchy of scene models.

III. CONCEPT OF THE HIERARCHY

In this section we will present a hierarchical organization of situation models. We will show that this organization is beneficial for a top-down approach by not allowing two models on the same level to be active at the same time. The goal of such a structure is to simplify a complex problem such as the combination between classifiers. The principle is to develop a competition between the results of the more specific nodes with the results of the generic ones. We will propose a node activation principle showing the robustness against errors: if a wrong localization prevents the activation of a specific node, the generic node will still be active as a fail safe mechanism. Finally we will outline the fact that this system is easily extendable to new specific scenes, by adding nodes as new leaves.

A. Structure of the Hierarchy

- a hierarchy $H = (N, E)$ is composed of :
- a set of nodes $N = \{n_i | i = 1...I\}$ where I is the total number of nodes N.
- a set of vertices $E = \{(x, y) | x, y = 1...I, x < y\}$.

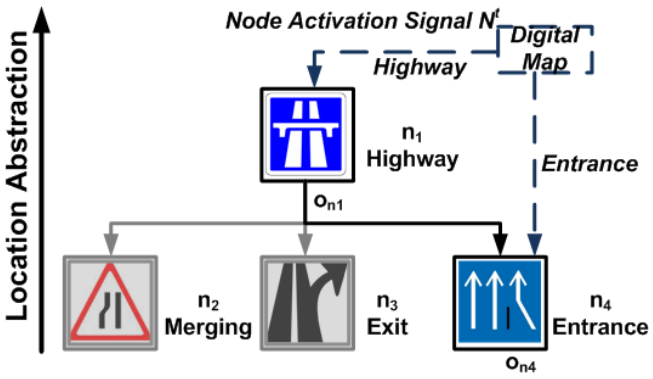


Fig. 2. Visualization of the highway hierarchy. Each rectangle represents a node of the hierarchy. The node $n1$ represents the highway node and is activated by digital map. The three nodes n_2, n_3 and n_4 represent the sub-nodes merging scenario, exit scenario and entrance scenario specific to the highway. They are connected with the highway with grey or black lines. The grey lines mean the sub-node is not active. The entrance node n_4 is also activated by digital map. It receives the output $\mathbf{o}_i$ from the highway node and returns its own output.

- a node $n_i = (M_i, b_i)$ is made of a set of models $M_i = \{m_j^i\}_{j=1...J^i}$ where J^i is the number of models m_j^i of the node n_i and a competition function b_i.

- a model $m_j^i = (\mathbf{x}_j^i, f_j^i(\mathbf{x}_j^i), r_j^i)$ contains:
  - a set of features $\mathbf{x}_j^i = [x_{j,1}^i, ..., x_{j,L_j^i}^i]^T$ where L_j^i is the number of features of the vector $\mathbf{x}_j^i$ of the model m_j^i.
  - a set of 2-class classifiers $f_j^i(\mathbf{x}_j^i) = \{f_{j,1}^i(\mathbf{x}_j^i), ..., f_{j,C_j^i}^i(\mathbf{x}_j^i)\}^T$ where C_j^i is the number of classifiers of the model m_j^i of the node n_i.
  - a set of conditions to activate the model $r_j^i = \{r_{j,k}^i\}_{k=1...K_j^i}$ where K_j^i is the number of conditions r_j^i of the model m_j^i of the node n_i. Once a node is activated by the context signal as defined below, the conditions control which of the models within the activated nodes should be used for prediction. For example, if the ego-vehicle approaches an entrance, the node for the entrance will be activated as shown in Fig.2. The active model rule of this activated node will control which model should be applied to which vehicle.

B. Activation Principle

- At time t, a signal activates the most specific node that fits the current context in the hierarchy. This also activates all its parent nodes.
  Context information can be obtained from different sources like for example from digital maps. The authors of [9] use a digital map to activate their system when the ego-vehicle is approaching an exit area. To avoid errors from digital map or gps localization, it is also possible to confirm the context data by sensors like lidar or vision. For example it is possible to use [9] to detect an exit, or [20] to detect road environment and [12] to detect roadworks.

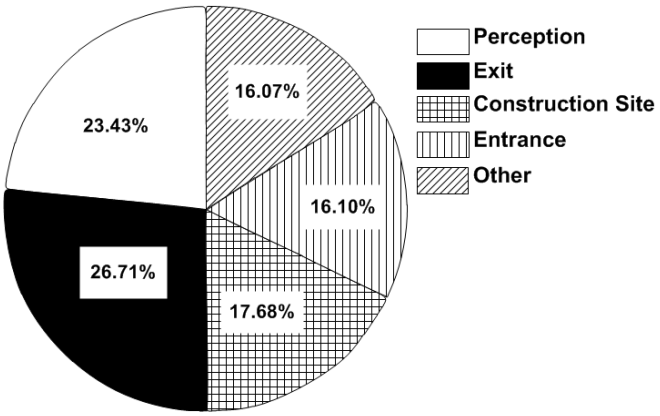


Fig. 3. Error analysis of the highway model on 30 minutes of recordings. 60.49% of the errors can be attributed to the complexity of the scenario and only 39.5% to perception or others.

C. Vehicle Prediction Computation

For each vehicle, the top node applies the conditions r_j^i to activate a model. The active model computes its prediction. The prediction is then sent to the active sub-node which does the same process. To make the best prediction out of two nodes, finally the active sub-node n_i computes its output $\mathbf{o}_i = b_i(\mathbf{c}_i, \mathbf{o}_j)$ by applying its competition function b_i between its prediction $\mathbf{c}_i$ and the output of its parent node $\mathbf{o}_j$.

Each node will have its own competition function based on context and prior knowledge.

To illustrate the concept of competition, the next part will describe the hierarchy of a highway system and introduce competition functions in a lane change prediction setup.

IV. APPLICATION TO HIGHWAY SITUATIONS

Fig.2 presents an example of the hierarchy for predicting behaviors on the highway. Each sub-node represents a scenario specific to the highway setting and introduces some additional context information. Each node will be detailed below including its activation rule and its competition function. Afterwards, we will evaluate the architecture based on a sub-set consisting of the highway and the entrance node.

At the top node, we use a generic highway model developed by [18] to predict if any of the surrounding vehicles will change lane to the left or right with a time horizon of 2 seconds. This model usually performs well on highway even in the presence of occlusion or errors in low level perception as shown in Fig.3. The error analysis categorizes failures by their cause and reveals that most errors are caused by specific, exceptional scenarios. The reasons for this scene specificity is that the highway node predicts vehicles irrespective of the surrounding environment. It does not consider changes of the infrastructure, such as starting or ending lanes. We need to combine different classifiers to compensate the errors of the highway node.

In the highway hierarchy presented in this section, a node will be activated by a digital map as shown in Fig.2. Once a node is activated, it will decide which model of the node

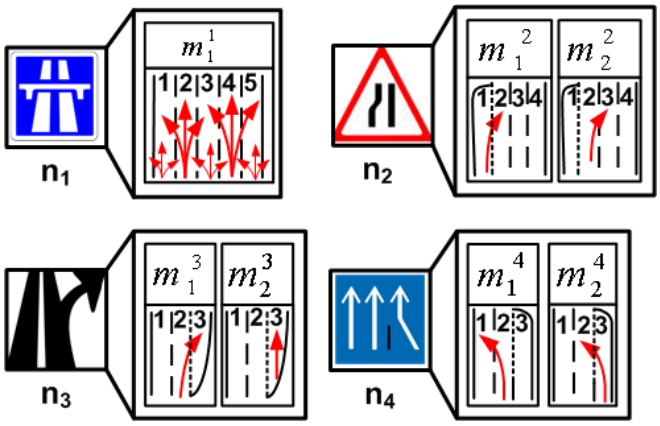


Fig. 4. The highway, merging, exit and entrance nodes are presented here to predict lane changes on highways with their corresponding models. The arrows in the models show for which lane the model is active and in which direction a vehicle will change lane. For example, the model m_2^4 of the entrance node n_4 will predict if a vehicle driving on an entrance lane (lane 3) will change lane to the left.

should be activated based on the lane of a vehicle. Fig.4 shows the nodes of the hierarchy with their corresponding models and which lanes they are applied to.

A. Highway Node

This node predicts if a surrounding vehicle will change lane or keep going straight on. It makes errors in the prediction when the scene becomes too specific as shown in Fig.3. 60.49% of prediction errors are caused by specific scenarios and can be avoided with a good combination between generic and specific classifiers.

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 the model m_1^1 of the node n_1 in Fig.4). The model represents the scene and identifies the reasons why a vehicle will change lane by computing some context based features. For example, the main reasons why a driver is changing lane to the left is when he is approaching a slower vehicle and wants to overtake it (Fig.5.a). After an overtaking maneuver, a driver might change lane to the right if the right lane is free (Fig.5.b).

Reasonable context based features thus evaluate speed relations between a vehicle and its surrounding vehicles. With these features, a classifier can be trained in order to predict a future lane change behavior. In this work we do not detail the content of the node as we focus on the combination between nodes. We use the implementation described in [18]. The model m_1^1 of the node n_1^1 is based on specific features like the time-to-contact with the predecessor or the size of the left gap and the time to reach the left gap. These features are used to train two single layer perceptrons to predict lane change left and lane change right.

The highway node provides for each surrounding vehicle confidences for the predictions of going straight, for changing lane to the left and to the right which are grouped into a confidences vector $\mathbf{c}_1 = [c_{straight_1}, c_{left_1}, c_{right_1}]$. All these confidences vectors are grouped into one prediction vector $\mathbf{o}_1$ which is then sent to the active child-node. In the example in Fig.2, the highway node n_1 sends the prediction vector $\mathbf{o}_1$ to the active entrance node n_4.

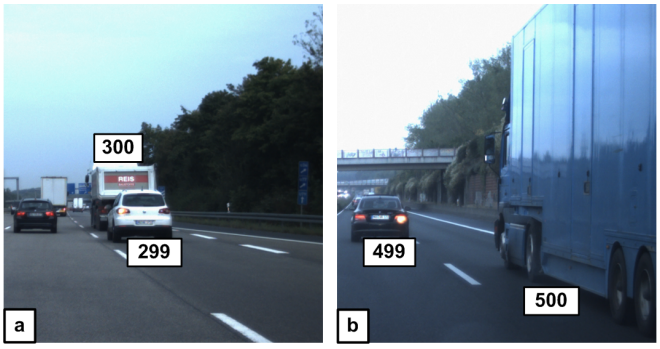


Fig. 5. Here is an example of a lane change to the left and a lane change to the right on the highway. In figure a, vehicle 299 is driving on the right of the ego-vehicle and because 300 is slower, 299 will change lane to the left. In figure b, Vehicle 499 will change lane to the right after it has passed vehicle 500 and because the right most lane is free.

B. Entrance Node

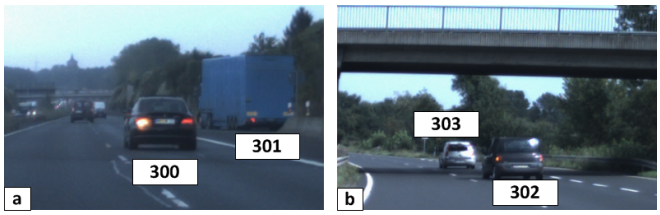
The entrance node represents an entrance scenario where for a short period of time there is an additional lane (lane 3) and an entering vehicle has to change lane before the lane ends (node n_4 in Fig.4).

Scenario: A vehicle driving on the entrance lane (lane 3) will change lane to the left before the lane ends even if there is a dense traffic and not much free space. The question is to find out if the vehicle will enter the highway before or after its left successor by using the model m_2^4 (node n_4 Fig.4). We defined some specific features like the time-to-contact with the left successor and the time to the end of the entrance to train a single layer perception to predict if the vehicle driving on the entrance lane will enter the highway before or after its left successor.

It might happen that a vehicle driving on lane 2 will give way to the left to make the lane free for the entering vehicle. This behavior is predicted by the model m_1^4. We defined some specific features like the time-to-contact with the right predecessor or the size of the left gap and the time required to enter it to train a single layer perceptron to predict if a vehicle driving on the left neighboring lane of the entrance lane will change lane to the left to free the lane.

The implementation of the models that we will use inside this node was discussed in more detail in a previous publication (see [1]).

Competition Function: For this scenario, the competition function is applied for the prediction of vehicles driving on lanes 2 and 3. All predictions $\mathbf{o}_{n1}$ of the highway node for vehicles not driving on these two lanes will be preserved in the final confidence vector $\mathbf{o}_{n4}$. The competition rule is applied only for the prediction of the vehicles driving on these two lanes. The final prediction will be the prediction of the node which has the highest confidence for changing



	Highway Node	Entrance Node
300	~~Straight~~	Change Lane
302	Change Lane	~~Straight~~

Fig. 6. Vehicle 300 and 302 are driving on the left neighboring lane of the entrance lane on the highway. Vehicle 300 will change lane to free the lane to the entering vehicle 301. Vehicle 302 will change lane because of the slow predecessor 303. Highway node wrongly predicts that 300 will drive straight. Entrance node wrongly predicts that 302 will drive straight.

lane to the left as shown in the example below. Then the confidence for changing lane to the right of the final confidence $\mathbf{o}_{n4}$ is set to 0 because changing lane to the right is not allowed for these lanes.

The two models of node n_4 use the same competition function. In this situation, the results of the highway node and the entrance node have to be combined in order to compensate for prediction errors of the highway node. When driving on the highway, the highway node predicts that a vehicle changes lane when approaching a slow predecessor as shown in Fig.6.b. The entrance node predicts that a vehicle driving on the left neighboring lane of the entrance lane changes lane to the left due to an entering vehicle as shown in Fig.6.a. Because both nodes predict lane change behaviors for two different reasons and could have opposite point of views, it is beneficient to combine both nodes via our hierarchy in order to cover all situation cases.

C. Merging Node

This node represents a merging scenario where the left most lane (lane 1) on the highway and its right neighboring lane (lane 2) will merge into a single lane (node n_2 in Fig.4).

Scenario: A vehicle driving on lane 1 will change lane to the right before the lane ends even if there is a dense traffic and not much free space. The model m_1^2 (node n_2 Fig.4) is designed to predict surrounding vehicles behavior driving on lane 1.

It might happen that vehicles driving on lane 2 change lane to the right. These vehicles are then predicted by the model m_2^2.

This model is really similar to the entrance node. They differ in the behaviors they want to predict: the merging node predicts lane changes to the right instead of the left. They could use the same models and same features but with a different scaling, a different competition function, and are not activated with the same signal.

Competition Function: The merging node predicts only behaviors of vehicles driving on lanes 1 and 2. The final confidence vector will get the prediction of the node which has the highest confidence for changing lane to the right. Then, the confidence for changing lane to the left of the final confidence vector obtained after the competition is set to 0, because the vehicles driving on these lanes should never change lane to the left.

Both models of the node n_2 use the same competition function.

We will provide an example for a scenario where the highway and the merging node will be conflicting in order to illustrate the benefit of using this competition function. A vehicle is driving on lane 2 and approaches a slow vehicle. The highway node will predict this vehicle has to change lane to the left. The merging node will forbid this maneuver because the left most lane is ending and it is not allowed to change lane anymore. Then the merging node will win in this case because he has a confidence of 0 for the left and therefore usually has some small value for going right and predicts a vehicle will drive straight.

D. Exit Node

This node represents an exit scenario where there is an additional lane (lane 3) for a short period of time which vehicles will take to leave the highway (node n_3 in Fig.4).

Scenario: A vehicle driving on the exit lane will keep going straight and is predicted with the model m_2^3 (node n_3 in Fig.4).

A vehicle driving on lane 2 will change lane to the right to take the exit or drive straight and its behavior is predicted by the model m_1^3 (node n_3 in Fig.4). The model m_1^3 does not need to predict lane changes to the left because the highway node will do this.

Competition Function: The competition rule is applied only for the prediction of the vehicles driving on lanes 2 and 3.

The exit node is capable of predicting if a vehicle will change lane to the right or keep going straight. The highway node provides a prediction o_{n1} for going straight, left or right.

The competition function will compare the highest confidence value of the highway node with the highest confidence value of the exit node and keep the highest maximum confidence value.

The two models of node n_3 use the same competition function. The following example illustrates a situation where the highway and the exit node are in conflict. If a vehicle is driving on lane 2, the highway node will predict that this vehicle should change lane to the right. The exit node has detected that the vehicle is not slowing down, it then predicts that the vehicle will drive straight. On the other end, a vehicle driving on the exit lane should always be predicted to drive straight, but depending on the traffic density, the vehicle might slow down due to a slower predecessor. The exit node will predict that the vehicles should drive straight whereas the highway node will predict that the vehicle will change lane to the left. Also in this example, the the proposed

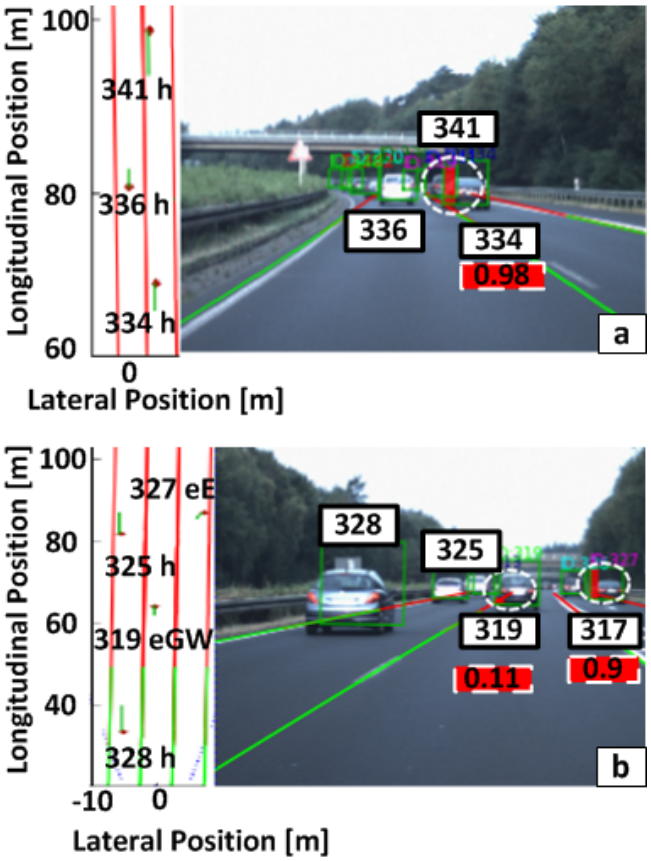


Fig. 7. The left part of both figures shows the ego-vehicle at the position 0 on the x axis, the lane marking and the radar targets with their relative velocity. In figure a, the letter 'h' denotes that the vehicle has been predicted by the model m_1^1 of the highway node. In figure b, letters 'eE' and 'eGW' mean that vehicles have been predicted by model 'Entrance Enter' m_2^4 and model 'Entrance Giveway' m_1^4. The right part of both figures represents a scene which can be classified by the highway node on top and by the entrance node below. The white dashed circle outlines the probability of a lane change for the important vehicles in the scene. Vehicles 334 and 317 will change lane to the left with a high probability and vehicle 319 will change lane with a low probability.

competition function will lead to the correct result in both cases

V. EXPERIMENTS

In this section we will demonstrate the usefulness of our approach using an example subset from the described highway hierarchy. This subset consists of the highway node and the entrance node tested to predict lane changes to the left on the highway.

All tests have been done using real data recorded on German highways. The vehicle used for the recordings is equipped with a stereo camera, a radar scanner and inertial and odometry sensors for egomotion. The dataset used for training the classifiers covers 100km under diverse weather conditions. We process the data with a frequency of 10Hz. The static infrastructure of the road, like number and spatial alignment of lanes, the start and end of an entrance has been manually annotated. Alternatively, this information could be obtained from a Digital Map [13], [22] or computed based on sensor data [20], [21].

In order to test our system we will evaluate the performances of the individual nodes and compare them with the performance of our hierarchy. We will show that our hierarchy performs better than the two individual nodes. In the following we will present the evaluation for the highway node, the entrance node and the hierarchy on the complete dataset.

Fig. 7.a shows a scene on default highway. The highway node predicts with a high probability of 0.98 that the vehicle 334 will change lane to the left due to the slow vehicle 341 in front.

Fig. 7.b shows a highway entrance scene. The entrance node predicts with a high probability of 0.9 that the vehicle 317 will enter the highway and predicts with a low probability of 0.11 that the vehicle 319 will giveway.

The classifiers of the highway model have been previously trained as described in [18]. The classifiers of the entrance models have been previously trained as described in [1]. These two nodes are used to compute the predictions for changing lane to the left of the surrounding vehicles. To test our system we use a test dataset which has not been used in the training phase which contains 100km of recordings on the highway and five entrances.

A. Performance Evaluation

Fig.8 shows a Receiver Operating Characteristic (ROC) curve [3] of the highway model with 0.33 of true positive rate (TPR) at a false positive rate (FPR) of 0.018.

These tests aim at evaluating the quality of the entrance models on the highway and entrances for each lane on the highway. Fig.8 shows a ROC curve for the model 'Entrance Enter' with 0.005 TPR at at a FPR of 0.018. The ROC curve of the model shows that the FPR increases without having much effect on the TPR.

The hierarchy combines the highway node and the entrance node. The highway node is constantly active while the entrance node is active only when the ego-vehicle is approaching an entrance scenario. Fig.8 shows the ROC curve for the hierarchy which has 0.64 of TPR at at a FPR of 0.018.

The majority of the errors that the highway model produces are found for predictions in specific scenarios.

The low performance of the entrance node on the complete dataset is not surprising, since we apply the model to dedicated and mostly non dedicated scenarios. The entrance model is designed to predict behaviors of the vehicles driving on an entrance lane and on the left neighboring lane. In this test, we have activated the entrance node on the complete dataset and applied the models to every lane of the highway. This way we ensure to compare both models on the same dataset. The model has a strong bias to predict lane changes to the left and makes a lot of prediction errors when applied to the overall highway due to the small number of vehicles which actually change lane.

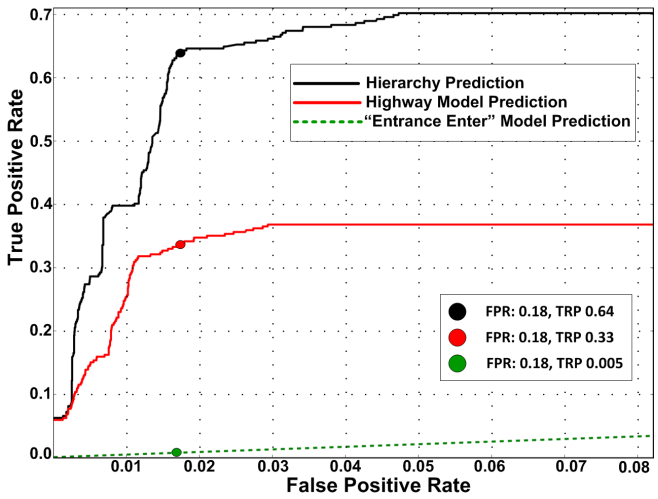


Fig. 8. ROC Curves of the hierarchy, the highway node and the entrance node to predict a lane change left on the highway. The hierarchy presents the best performance with a true positive rate of 0.64 at a false positive rate of 0.018. For a same false positive rate, the ROC curve of the highway node has a true positive rate of 0.33. The lowest ROC curve shows the performance of the model 'Entrance Enter' which has been applied on the complete dataset and shows a true positive rate of 0.005.

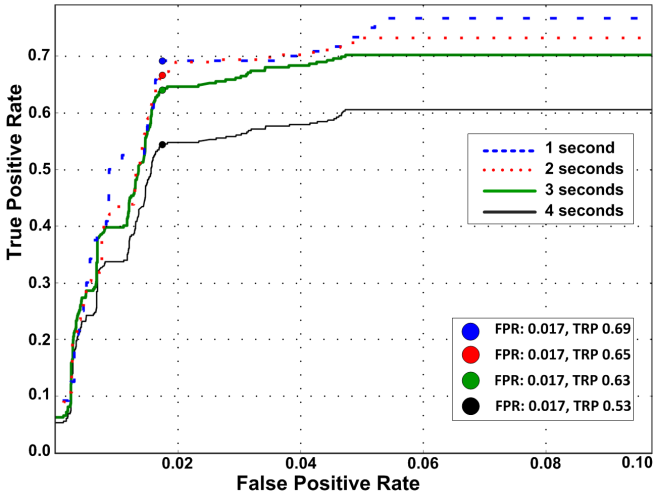


Fig. 9. ROC curves of the hierarchy for different time windows to measure how much in advance a prediction of a lane change left on the highway starts.

The ROC curve of the hierarchy shows that our system is able to compensate the errors made by the single classifiers when these errors result from the scenario. This confirms that a well orchestrated combination between nodes and classifiers can improve the prediction accuracy. The remaining errors in the prediction mostly come from errors in low level perception, radar range limitations or scenarios we did not yet included in our hierarchy (e.g. exits).

B. Time Horizon Evaluation

In this evaluation, we are investigating how much in advance the system starts the prediction. For this evaluation we use the dataset built to test our hierarchy and make the groundtruth for different time windows. We have trained our system with a groundtruth of three seconds. This means that a vehicle which changes lane to the left is annotated has a positive example for a lane change to the left from the moment he crosses the left line to three seconds before. For testing the prediction horizon, we created different test sets by annotating a vehicle which changes lane to the left from the moment he crosses the line to one second before, two seconds before, three seconds before and four seconds before.

Fig.9 shows the ROC curve of the hierarchy for four different time windows. We observe that for the different time window, for a same false positive rate of 0.017, the true positive rate decreases from 0.69 for 1 second to 0.65 for 2 seconds, 0.63 for 3 seconds and quite drastically to a TPR of 0.53 for 4 seconds. Because our system has been trained to predict a lane change three seconds in advance, it is natural that the accuracy decreases when predicting behaviors beyond three seconds. This way we also make sure that the system is capable of predicting a lane change one to three seconds in advance without loosing much in performance.

With this experiment we show that our hierarchy can give an accurate prediction two to three seconds in advance which is the necessary time to provide an early warning to the driver [8].

C. Computation Time Analysis

The computation time of our hierarchy will be influenced by the number of nodes n, the number of models m, the number of vehicles v in the scene and the number of surroundings s of a vehicle. Theoretically, the hierarchy complexity is $O(n * m * v * s)$.

Because the number of nodes n is constant, the hierarchy complexity is $O(m * v * s)$.

The number of models m is constant which reduces the hierarchy complexity to $O(v * s)$.

The only part where it depends on the number of other cars in the scene is in the calculation of some of the features (e.g. distance to left Successor). But only a small subset of all the cars has to be considered, namely those in the direct surrounding. In our case we use maximally three other cars in the 'Giveway' model and two in the 'Entrance Enter' model. At the end the hierarchy complexity is $O(v)$.

We observe that our hierarchy which has a final complexity of $O(v)$ will scale linearly for complex scene. As an example, we computed the computation time for an average of three vehicles on the highway. On average, the highway node has a computational time of 20.02 milliseconds and the entrance node has a computational time of 4.09 milliseconds on a standard desktop computer.

VI. CONCLUSION AND OUTLOOK

In this paper, we outline that none of the existing work is able to predict driving behavior with a high quality in a wide scope of sitations.

We propose a generic concept based on a hierarchy of scene models in order to be able to predict the behavior of surrounding vehicles for a large variety of scenes. The idea is

to identify the scenes that a generic model cannot cover and create context based models of these scenes. We then order them into a hierarchy according to their location specificity. Finally we apply competition rules between nodes in order to always get the most accurate prediction.

We tested our hierarchy on highways on the task of predicting lane change behaviors of surrounding vehicles. The experiments show the benefit of using a hierarchy in order to improve scope and quality.

We decided to give a detailed description of the concept of the hierarchy. We developed the entrance node as a sub-node of the highway node and showed the improvement of the behavior prediction by our hierarchy after applying the competition function. We did not extend the hierarchy to other highway scenarios as the nodes will be very similar to the entrance node.

The proposed structure allows us to combine specific methods with narrow scope and high quality in order to get a system able to predict behaviors for a lot of scenes.

The good performances of the hierarchy are influenced by the accurate localization of the ego-vehicle into the digital map. Additional sensor support can be added in order to verify or correct the localization of the ego-vehicle. For example, an entrance could also be detected using lane markings [9]. Each node of the hierarchy strongly depends on the low level perception. The hierarchy can not improve the prediction of the individual nodes. It is designed to improve the prediction by combining the results of the different nodes based on an accurate sensing.

In the next step, we want to do an event based evaluation to analyse how the system might influence a real driver.

In the future work we will focus on more complex scenarios as we will extend our concept to inner-city scenarios.

REFERENCES

[1] 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.

[2] 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.

[3] T. Fawcett, "An introduction to ROC analysis," *Pattern recognition letters*, pp. 861–874, 2006.

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

[5] A. Gerdes, "Driving maneuvre recognition," in *Proc. IEEE Intelligent Transportation Systems*, 2006, pp. 1–8.

[6] 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.

[7] L. He, C. Zong, and C. Wang, "Driving intention recognition and behaviour prediction based on a double-layer hidden markov model," *Zhejiang University - Science C*, pp. 208–217, 2012.

[8] 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.

[9] 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.

[10] 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.

[11] 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.

[12] 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.

[13] J. McCall, D. Wipf, 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.

[14] 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.

[15] B. Morris, A. Doshi, and M. Trivedi, "Lane change intent prediction for driver assistance: On-road design and evaluation," in *Proc. IEEE Intelligent Vehicles Symposium*, 2011, pp. 895–901.

[16] OECD, *Health at a Glance 2011*. OECD Publishing.

[17] M. Reichel, M. Botsch, R. Rauschecker, K. 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.

[18] J. Schmüdderich and S. Rebhan, "A method and system for predicting movement behavior of a target traffic object," Patent EP 11 178 252.0.

[19] N. Sekiyama and T. Watanabe, "A method for calculating future behaviors of vehicles toward effective collision prediction," in *Proc. IEEE Engineering and Computer Science*, 2009.

[20] 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.

[21] M. Thuy and F. P. León, "Lane detection and tracking based on lidar data," *Metrology and Measurement Systems*, pp. 311–322, 2010.

[22] C. Wang, Z. Hu, and R. Chapuis, "Predictive lane detection by interaction with digital road map," *Journal of Information Processing*, pp. 287–296, 2012.