2012 15th International IEEE Conference on Intelligent Transportation Systems
Anchorage, Alaska, USA, September 16-19, 2012

A Generic Concept of a System for Predicting Driving Behaviors

Sarah Bonnin[1], Franz Kummert[2], Jens Schmüdderich[3]

Abstract— **Today, many vehicles are equipped with Advanced Driver Assistance Systems (ADAS) to warn the driver about the potential danger of a scene, but in some situations the warning is not early enough to avoid an accident. A solution for preparing the driver and giving him the time to react to such dangerous events is to predict the behavior of other traffic participants. This paper describes a method to predict the behavior of the surrounding vehicles by a classification approach. However, the behavior alternatives strongly depend on the scenario faced by the target vehicle. Where most of the state-of-the-art approaches focus on a single scenario, the concept presented in this paper aims at a generic solution, allowing for behavior prediction for a large amount of different scenes. The idea of the method is to categorize scenes into a hierarchy from the most generic ones in the top nodes to the most specific ones in the leaves. Every node of the hierarchy is a scene containing a set of classifiers to predict the possible behaviors. GPS and digital maps provide the static information about the infrastructure, which is used to determine the nodes fitting to the current situation. As a first step this paper shows accurate prediction of traffic participants behavior in highway entrance situations for a prediction horizon of up to 3 seconds.**

I. INTRODUCTION

Many road accidents occur due to a misinterpretation of the scene by the driver, or a late estimation of the risks of collision with surrounding vehicles. To face this problem and improve road safety, cars are equipped with Advanced Driver Assistance Systems (ADAS) to warn the driver about a dangerous situation, but in some cases this warning is too late to avoid the crash. One challenging issue is to predict the behavior of other traffic participants to prepare the driver and increase his time to reaction.

Several studies are performed in this field of research to anticipate behavior for highway or inner-city scenarios. On highway, recognition of driving maneuvers like cut-in is studied using Bayesian network [4]. It is possible to predict convoy merging situation on highway with a Random Forest algorithm [12]. Several approaches can provide lane change, overtake or object following maneuvers using Bayesian networks [6] [7] or employing the Dempster-Shafer theory [14]. Behavior prediction is also researched in inner-city scenarios, comparing the results of a Hidden Markov Model with the results of a Support Vector Machine [1] to detect if the vehicle will stop safely at traffic lights. To estimate turning maneuver at an intersection, the authors of [8] associate contextual data from Digital Maps with a Bayesian network or create a model based on vehicle control signals [11]. All these approaches show accurate prediction for specific scenes but they are not general.

The reason for this specificity is that a driver is influenced by his local context, adapting his behavior according to the surrounding. Considering for example a vehicle which is driving on the right most lane on the highway, it might change lane when approaching an entrance, because of the vehicles driving on this entrance lane. At the same time, the vehicle driving on the left most lane will not be influenced by the entering vehicles and will not modify its behavior. Traffic rules also depend on the context. Behavior at an unsigned intersection differs from that at a traffic light intersection. Due to the context, drivers adapt their behavior and therefore behavior prediction is very specific to the scene.

To cope with the large variety of scenes, the idea presented in this paper is to create a system that can predict generic behaviors e.g. lane changes and more specific ones e.g. stopping at zebra crossings. It categorizes scenes into a hierarchy, the top nodes modeling generic situations and the leaves the most specific ones see Fig.1. Every node of the hierarchy contains models for a certain scene with one model for each possible behavior to predict. A node activation method based on the ego-vehicle position activates the nodes appropriate for the current scene. A GPS provides the position and Digital Maps deliver static information about road infrastructure, which is used to determine the node fitting the current situation.

To illustrate our concept, this paper proposes an evaluation of the behavior prediction on entrance scenario on highway. In a future work, the specific scenario will be combined with the generic ones to predict behavior on highway.

The paper is outlined as follows: section II motivates the reason to create a hierarchy, describes its structure and the process to activate a node. Section III explains the content of one example node representing a highway entrance scene, which behavior is predicted by the node and details of the behavior prediction rules. In section IV we experimentally demonstrate accurate prediction on highway entrance scenes, showing the benefit of specific classifiers. A conclusion and an outlook are presented in the final section.

II. CONCEPT OF THE HIERARCHY

The first reason to choose a hierarchy is that it is an appropriate structure for a top-down approach. It allows competition between classifiers, the more specific scenarios with more constraints compete with the results of the generic ones.

[1]S. Bonnin is with Cor-Lab, Bielefeld University, Germany, sarah.bonnin at gmail.com
[2]F. Kummert is the Faculty of Technology, Bielefeld University, Germany, Franz at techfak.uni-bielefeld.de
[3]J. Schmüdderich is Honda Research Institute Europe GmbH, offenbach, Germany Jens.Schmuedderich at honda-ri.de

978-1-4673-3063-3/12/$31.00 ©2012 IEEE

The second point is that this system is easily expendable to new specific scenes, by adding nodes in the 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, and a set of vertices $E = \{(x, y)|x, y \in N\}$.
- A node $n_i = M_i$ 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 e.g. in Fig.1 the node n_1 for highway contains one model m_1^1, which uses one classifier to predict the vehicle will change lane to the left, one classifier to predict the vehicle will change lane to the right, and a combination of the results of both classifiers to predict the vehicle will drive straight.
- A model $m_j^i = (\mathbf{x}_j^i, \mathbf{f}_j^i(\mathbf{x}_j^i), r_j^i)$ contains a vector 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 vector of 2-class classifiers $\mathbf{f}_j^i(\mathbf{x}_j^i) = [f_{j,1}^i(\mathbf{x}_j^i), ..., f_{j,C_j^i}^i(\mathbf{x}_j^i)]^T$ where C_j^i is the number of classifiers of the model m_j^i of the node n_i, and a set of rules to activate the model $r_j^i = \{r_{j,k}^i\}_{k=1...K_j^i}$ where K_j^i is the number of rules r_j^i of the model m_j^i of the node n_i. Once a node is activated by the Digital Map signal as defined below, the rules control which of the models within the activated nodes should be used for prediction. For example, if the ego-vehicle approaches an entrance, a node for the entrance will be activated. The rule of this activated node will control which model should be applied to which vehicle as it is explained in section III-B.

For convenience we define functions to obtain the set of children (1) and the parent node (2) of a node n_i:

$$s : N \to N, s(n_i) = \{n_j|\exists(n_i, n_j) \in E\} \tag{1}$$

$$p : N \to N, p(n_i) = \{n_j|\exists(n_j, n_i) \in E\} \tag{2}$$

B. Node Activation Principle

The activation mechanism is chosen such that it only allows a top-down node activation. It is not possible to activate two nodes which are on the same level.

- A time t, the Digital Map sends a set of signals $Z^t \subset E$ to activate the vertices of the hierarchy.
- A node is activated if its parent is activated and the Digital Map has activated the vertice between the parent and the node. At time t, the set of activated nodes is $N^t = \{n_i \in N|\exists x \in N : (x, n_i) \in Z^t\}$.

C. Concept of the Competition between Classifiers

- Every node provides a confidence vector to predict the lane change to the left, to the right and the function g_i combines both confidences to compute the confidence for going straight. The confidence vector for the node n_i is $\mathbf{c}_i = g_i(\mathbf{f}_j^i) = [c_{left_i}, c_{right_i}, c_{straight_i}]$.
- For each sub-node n_i, we define a function b_i to compute the final confidence output $\mathbf{o}_i$, by implementing a competition between the confidence $\mathbf{c}_j$ of the parent node $n_j = p(n_i)$ using (2) and the confidence $\mathbf{c}_i$ of the sub-node n_i, $\mathbf{o}_i = b_i(\mathbf{c}_i, \mathbf{c}_j)$.

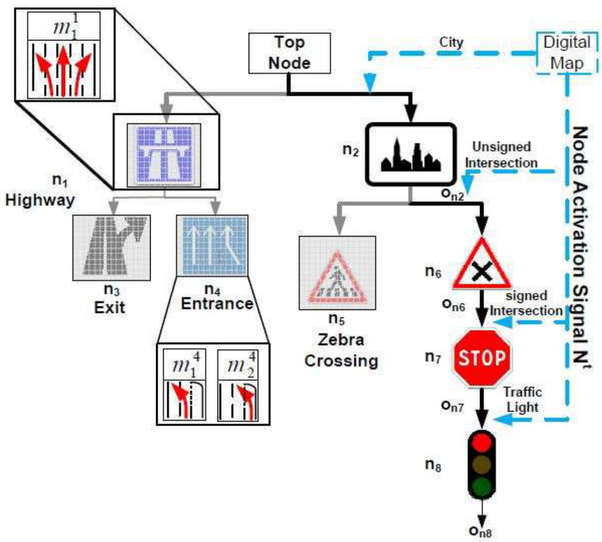

Fig. 1. A visualization example of a hierarchy. For the highway node n_1 the models m_1^1 is visualized predicting the vehicles will change lane to the left, to the right or drive straight. The model m_1^4 and m_2^4 of the entrance node n_4 will be detailed later in this paper. On the right hand side the node activation is for example pictured for a situation approaches a signaled intersection. $\mathbf{o}_i$ is the output of node i as defined in section II-C.

To illustrate the concept of competition, Fig.1 shows a hierarchy. The ego-vehicle is driving in inner-city and it is approaching a traffic light. Fig.1 shows that before activating the traffic light, the Digital Map has to activate all the generic types of intersections. Each child node has to compute its confidence vector and apply the competition function to compete its results with the confidence vector of its parent node. Then it sends its output to its sub-node. The reason for such design is to facilitate the competition method applied on a pair of nodes. Imagine the traffic light is deactivated, the vehicle will have to respect the associated traffic sign or the zebra crossing. With this structure, if the traffic light is deactivated, the node traffic light will not provide any prediction vector and keep the confidence vector of its parent node as a final result. If the nodes were all on the same level, it would be very difficult to imagine a general competition function to know which node is correct. Furthermore, this structure promotes extensibility. Assuming that the nodes have the same parent node, adding a node would result in a change in the competition function. It is not possible to change the competition function each time a leaf has to be added as too many constraints have to be taken into account.

The Digital Map provides static information about the infrastructure and perception is required to get the dynamic context of the scene, detecting pedestrians, the state of traffic lights. The next section will detail the entrance node, a sub-node of the highway node as an example.

III. ENTRANCE SCENE

To test the concept presented in this paper, it is necessary to start with developing a specific situation that a generic classifier cannot predict. The entrance scenario shows the benefit of differentiating the generic behavior from the most specific. We plan to use an approach similar to Dagli [4] is used in the highway node n_1 to evaluate generic behaviors on highway, eg. cut-ins but it has not been implemented yet. At an entrance, a generic classifier cannot estimate the behaviors of the entering vehicle as the classifier has no concept of the entrance lane end where vehicles have to change lane. The model 'Entrance Enter' predicts the behaviors of the entering vehicles. Vehicles driving on the left neighboring lane of the entrance lane modify their behavior to adapt to the new context. They cannot change lane to the right and they have to watch out for the entering vehicles. The model 'Entrance Giveway' determines the behavior of the vehicles on the left neighboring lane. The entrance scene is a leaf of the highway node. This scenario is a relevant example to test our concept. This section presents the two models used to predict the behavior at an entrance.

A. General Method to Create a Model

A model represents the part of the scene used in the system. One model characterizes one behavior. In one situation, a driver can have different behaviors and different reasons to act. The reasons why a driver would change its behavior are based on different aspects of the scene and are symbolized by features. One model combines a set of features, a classifier and the activation rules to check if the model is suitable to anticipate the behavior of the driver. For example, if the entrance node is activated, the model 'Entrance Enter' will only compute the behavior of the vehicles on the entrance lane. The model 'Entrance Giveway' will take into account the vehicles on the left neighboring lane. Behaviors of vehicles on the left most lane will be predicted by the upper node, as they have no reason to act differently. The features utilized in a model are specific to the context, this makes the model unique and enables to distinguish the models from each other.

B. Model 'Entrance Enter'

To illustrate this model, Fig.2 presents the scene perceived by the ego-vehicle driving on highway and approaching an entrance. The vehicle a_1 is on the entrance lane and will have to change lane to the left.

- Scenario

A vehicle driving on an entrance lane that ends must always enter the left lane, paying attention to other vehicles. For an ADAS which reacts to cut-in behaviors, it is interesting to know if the vehicle will enter on the highway before or after its left successor. The model 'Entrance Enter' predicts this behavior.

- Features

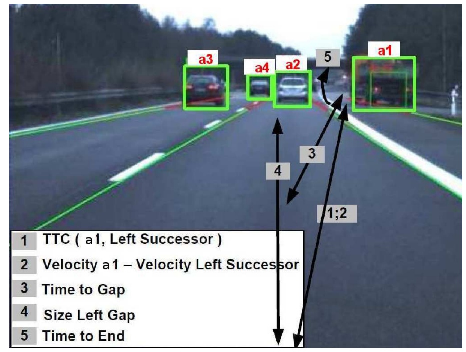

Fig. 2. Ego-vehicle approaching an entrance predicts if a_1 will enter before or after him by computing the different features

The features are designed to answer two questions: first, does the vehicle a_w have to enter the highway, secondly can it enter according to the free space and the other vehicles on its left. The features used for that are:

1- 'Time-to-Contact' (TTC) between a_w and its left successor (LS_w) allows to estimate the time before the two vehicles collide. [1]

$$x_1(a_w, LS_w) = TTC(a_w, LS_w) =$$
$$\triangle d(a_w, LS_w) / \triangle v(a_w, LS_w)$$

with $\triangle d$ the distance and $\triangle v$ the difference of velocity.

2- If the vehicles are close to each other the value of the TTC is small, but if the LS_w is faster than a_w the vehicles will not collide. $x_2(a_w, LS_w) = \triangle v(a_w, LS_w)$

3- 'Time-to-Gap' (TTG) is the time of a_w to reach the left gap ($LP_w; LS_w$). If the vehicle is equidistant to the LS_w and left predecessor (LP_w): $x_3(a_w, LS_w, LP_w) = TTG(a_w, LS_w, LP_w) = 0$. If the vehicle is closer to the LS_w than to the LP_w: $x_3(a_w, LS_w, LP_w) = TTG(a_w, LS_w, LP_w) = TTC(a_w, LS_w)$. If the vehicle is closer to the LP_w than to the LS_w: $x_3(a_w, LS_w, LP_w) = TTG(a_w, LS_w, LP_w) = TTC(a_w, LP_w)$

4- 'Size Gap' is useful to know if the vehicle fits into the left gap according to the length of the gap and the distance to the gap.

$$x_4(a_w, LS_w, LP_w) = SizeGap(a_w, LS_w, LP_w) =$$
$$\triangle d(LP_w, LS_w) * TTC(a_w, LS_w) * TTC(a_w, LP_w)$$
$$* \triangle d(a_w, LS_w) * \triangle d(a_w, LP_w)$$

5- The 'Time-to-End' (TTE) is the time of a_w before reaching the end of the entrance.

$$x_5(a_w, Tend) = TTE(a_w, Tend) = \triangle t(a_w, Tend)$$

6- To evaluate the moment a_w should change lane according to the LS_w and the TTE of the entrance.

$$x_6(a_w, Tend, LS_w) = ShouldEnter(a_w, Tend, LS_w)$$

[1]The indices i, j to identify the model are dropped here for convenience.

$$= TTC(a_w, LS_w) - TTE(a_w, Tend_w)$$

7- The prediction of its LS_w from the previous timestamps to know earlier when the gap is free or if a_w can change lane:

$$x_7(a_w, LS_w) = previousPrediction(LS_w)$$

- Activation rules

The ego-vehicle is driving on a highway. The highway node n_1 (Fig.1) is enabled to predict the generic behavior of the vehicles a_1, a_2, a_3, a_4 in the scene shown in Fig.2. When the ego-vehicle is approaching the entrance, the Digital Map activates the vertice $(highway, entrance)$. The entrance node n_4 becomes active. The highway node sends for each vehicle in the scene their prediction vector to the entrance node. The activation rules r_2^4 of the model 'Entrance Enter' then checks if the vehicle detected in the scene is on the entrance lane by lateral position. If not, the model does not provide any prediction for this vehicle and keeps the prediction vector of the top-node. Following this rule, the model evaluates only the behavior of the vehicle a_1 in Fig.2.

- Classifier

By creating a very specific model, we want to show if it is possible to choose features which are linearly separable and to use a simple classifier such as Single Layer Perceptron (SLP) [3].

The highway node n_1 sends for each vehicle a confidence vector with the prediction vector for changing lane to the right, to the left and going straight: $\mathbf{o}_1 = [c_{left_1}, c_{right_1}, c_{straight_1}]$. The model m_2^4 provides a confidence vector $\mathbf{c}_4 = [f_1^4, 0, 1-f_1^4] = [c_{left_4}, c_{right_4}, c_{straight_4}]$ to predict if a_1 will change lane to the left before its LS. The confidence for going right is set to 0 because the vehicle a_1 cannot change lane to the right. Then the model applies the competition function detailed in the algorithm (1) to decide if confidence c_{left_4} of node n_4 or confidence $\mathbf{c}_{left_1}$ of the node n_1 is better for the prediction of the vehicle a_1.

Algorithm 1 Competition function for the models of the node n_4

```
if (c_left4 < o_left1) then
    o4 = o1
    o_right4 = 0
else
    o4 = c4
end if
```

C. Model 'Entrance Giveway'

Fig.3 represents another scene perceived by the ego-vehicle: vehicle a_1 is on the left lane of the entrance lane and will have to change lane to the left to clear the lane for the vehicle a_2 on the entrance lane.

- Scenario

The highway node predicts the behavior of the vehicles on the highway. But this node does not reflect changes that are caused by an entrance lane. The model 'Entrance Giveway' predicts if the vehicle a_1 will change lane to the left to give way to the entering vehicle a_2.

- Features

The features are designed to answer two questions, does the vehicle a_w have to change lane due to the right predecessor (RP_w), and can it change lane to the left according to the free space and the other vehicles on its left.

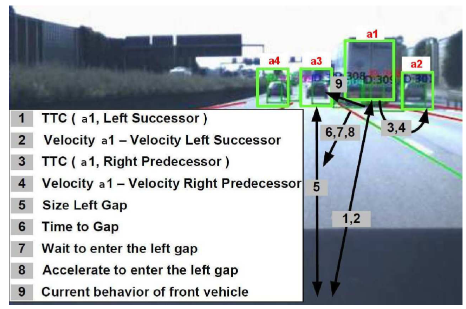

Fig. 3. Ego-vehicle approaching an entrance predicts if a_1 will change lane to the left to give way for a_2

1- $x_1(a_w, LS_w) = TTC(a_w, LS_w) = \triangle d(a_w, LS_w)/\triangle v(a_w, LS_W)$
2- $x_2(a_w, LS_w) = \triangle v(a_w, LS_w)$
3- $x_3(a_w, RP_w) = TTC(a_w, RP_w) = \triangle d(a_w, RP_w)/\triangle v(a_w, RP_w)$
4- $x_4(a_w, RP_w) = \triangle v(a_w, RP_w)$
5- $x_5(a_w, LS_w, LP_w) = SizeGap(a_w, LS_w, LP_w) = \triangle d(LP_w, LS_w)$
6- $x_6(a_w, LS_w, LP_w) = TTG(a_w, LS_w, LP_w)$
7- To estimate if the vehicle is going to fast and has to wait taking before entering the gap: $x_7(a_w, LS_w, LP_w) = Wait(a_w, LS_w, LP_w) = TTG(a_w, LS_w, LP_w) * SizeGap(LS_w, LP_w) * \triangle d(a_w, LP_w)$
8- To compute the acceleration required by the vehicle to enter the gap. $x_8(a_w, LS_w, LP_w) = Accelerate(a_w, LS_w, LP_w) = TTG(a_w, LS_w, LP_w)*$

$$SizeGap(LS_w, LP_w) * Acceleration(a_w)$$

9- The Coefficient of Visibility is a feature used to approximate the percentage of the entrance lane hidden by the vehicles. To compute the area occluded by the vehicle, we compute the dashed area as shown in Fig.4. $x_9(a_w, LS_w, LP_w) = VisCoefficient(a_w, LS_w, LP_w)$
10- The current behavior of the predecessor to evaluate the future gap on the left as if the predecessor changes lane it will become the new LP_w: $x_{10}(a_w, P_w) = currentBehavior(P_w)$
11- The prediction of itsLS_w from the previous timestamps to know earlier when the gap is free or if a_w can change lane: $x_{11}(a_w, LS) = previousPrediction(LS_w)$
12- The prediction of its RP_w from the previous timestamps to know earlier if the RP_w wants to change lane:

$$x_{12}(a_w, RP_w) = previousPrediction(RP_w)$$

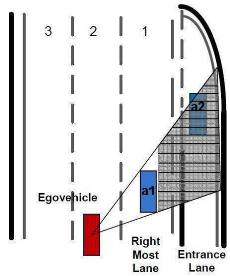

Fig. 4. The dashed area represents the coefficient of visibility which is the percentage of the road not visible by the radar.

- Activation rules

The activation rules r_1^4 of the model 'Entrance Giveway' check if the vehicle detected in the scene is on the left neighboring lane of the entrance by lateral position. If not, the model does not provide any prediction for this vehicle and keeps the prediction vector of the top-node. Following this rule, the model evaluates only the behavior of the vehicle a_1 in Fig.3.

- Classifier

The model 'Entrance Giveway' is also a very specific model, and the features are linearly separable. The competition function used in this model is the same as the 'Entrance Enter' detailed in the algorithm (1).

This section shows the advantages of using specific models to conceptualize a particular scene on the example of an entrance scenario. The next section presents some experimental evaluations.

IV. EXPERIMENTS

All tests have been done using real data recorded on the highway. The vehicle used for recordings is equipped with a stereo camera, a radar scanner and the inertial and odometry sensors for egomotion. The dataset consists of about 100km of video. It contains 20 vehicles detected on the entrance lane and 31 vehicles on the left neighboring lane.

In our approach, the static infrastructure of the road as the number of lanes on the highway or the start and the end of the entrance are manually annotated, but these information could be obtained by Digital Map [15] [13] [8] [5]. The ego-lane is also labeled, but it is possible to use the methods from [10] , [9] to localize and track the ego-lane.

To train and test the classifiers, we used a K-cross validation method [3]. we divide the dataset into k groups, training the classifier on $k-1$ groups and testing on 1 group, K times. We set K to 3 and each group contains the same number of vehicles labeled with 1 when they change lane and labeled to 0 when they do not. To evaluate the results we plot the ROC curve for the prediction of the vehicles used for testing. The ROC curve represents the True Positive rate (true positives/(true positives + false negatives)) with respect to the False Positive rate (false positives/(true negatives + false positives)) .

- Model 'Entrance Enter'

For this model, we used a SLP with 7 input neurons and a bias set to 1, a learning rate of 0.002 and a sigmoid function to activate the output. Fig.5 presents the ROC curve for the model 'Entrance Enter'. The result for this model is 80%TPR at 0% FPR. One comon error is visualized in Fig.6. In this scene, the entrance is made of two lanes, the right most lane is the entrance lane and the left neighboring lane is merging with the highway. The vehicles a_{54} and a_{55} are driving on the merging lane but the picture shows that they are detected on the entrance lane by our perception system. The ego-lane and the merging lane are separated by a dashed area which has the same width of a lane, and because the system knows there are only two lanes on the right of the ego-vehicle, the dashed area is computed as the giveway lane and the real giveway lane as the entrance lane. The detection of false positives is due to the prediction of vehicles by the model 'Entrance Enter' which are not well detected. They should be predicted by the model 'Entrance Giveway'.

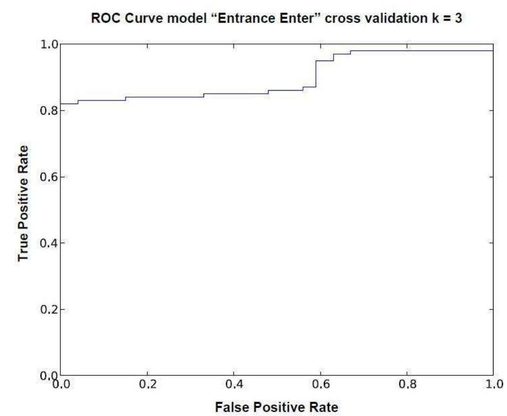

Fig. 5. The ROC curve for the model 'Entrance Enter' with 80%TPR at 0% FPR

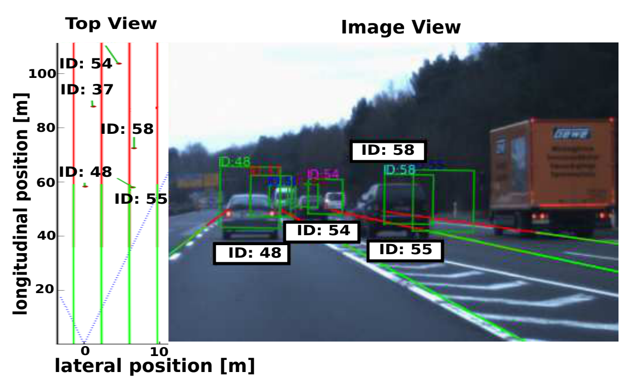

Fig. 6. The left part of this image shows the ego-vehicle at the position 0 on the x axis, the lane marking and the radar targets with their relative velocity. The scene on the right is an example which cannot be classified by the model 'Entrance Enter'. The vehicles are driving on the give way lane but they are detected on the entrance lane, the right most lane drawn on the left.

- Model 'Entrance Giveway'

The classifier used is a SLP with 12 input neurons and a bias set to 1, a learning rate of 0.001 and a sigmoid function

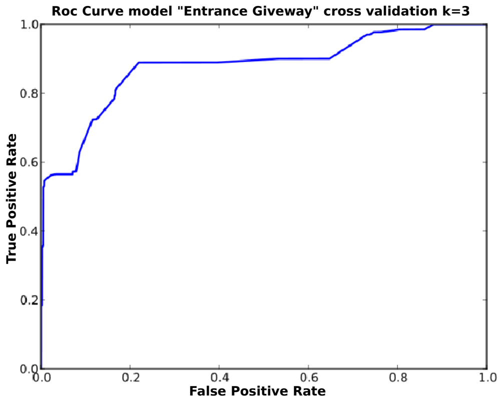

Fig. 7. The ROC curve for the model 'Entrance Giveway' with 57%TPR at 0% FPR

to activate the output. Fig.7 introduces the ROC curve for the model 'Entrance Giveway'. The result for this model is 57%TPR at 0% FPR. One comon error is visualized in Fig.8. In this situation the vehicle a_{268} driving on the giveway lane is faster than the truck a_{262} driving in front and it will change lane to overtake the truck. There is no vehicle driving on the entrance lane. This example gives false positives because the model 'Entrance Giveway' is not in charge of this scenario. The top-node highway which is not yet implemented will be able to predict this situation. This example demonstrates that the competition between classifiers is useful to predict generic and specific scenes.

This paper does not compare our system with the existing ones because it does not exist and it would not be fair to compare our approach with trajectory based approach.

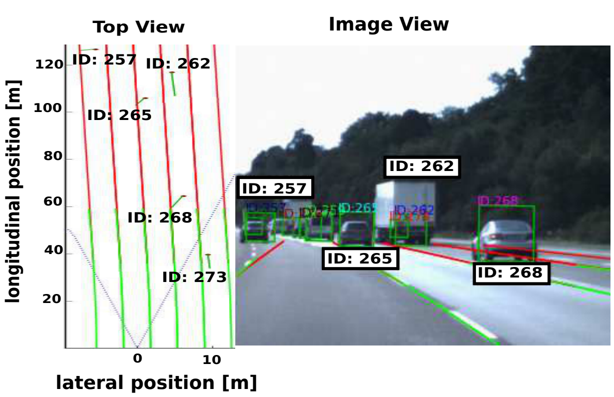

Fig. 8. The scene on the right is an example which cannot be classified by the model 'Entrance Giveway'. The vehicle 268 will change lane to overtake the truck in front of it, not because of a vehicle on the entrance lane.

V. Conclusion and Outlook

This paper describes a concept to predict the surrounding behavior vehicles on a large variety of scenes. It categorizes scenes into a hierarchy, the top nodes modeling generic situations and the leaves the most specific. Then GPS and Digital Maps provide the static information about the infrastructure, which is used to determine the nodes fitting to the current situation. Existing system can deal with errors in Digital Map [2]. The experiments were run on an entrance scenario showing that it is beneficial to distinguish very generic scenes to very specific ones depending on the context. This is exactly what the hierarchy does. It is pertinent to define which behaviors correspond to which scenario and predict the behaviors with specific classifiers. The evaluation proves that a specific classifier performs well. The advantage of such a structure is the extensibility of the system to new scenes.

The next step will be to combine the entrance scene with the generic classifiers using a similar approach to [4] to predict behaviors on highway. We will implement the competition between the classifiers of the top-node and the classifiers of the sub-nodes. Future work will include further evaluations for the entrance scene and extend the hierarchy to inner-city scenarios.

References

[1] G. Aoude, V. Desaraju, A. Stephens, and J. How, "Behavior classification algorithms at intersections and validation using naturalistic data," in *Proc. IEEE Intelligent Vehicles Symposium*, 2011, pp. 601–606.

[2] T. Bernd, L. Jan, D. Sinisa, and O. Hans-Ulrich, "The actmap-feedmap framework a basis for in-vehicle adas application improvement," in *Proc. IEEE Intelligent Vehicles Symposium*, 2008, pp. 263–268.

[3] C. Bishop, *Pattern Recognition and Machine Learning*. Springer, 2007.

[4] 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.

[5] D. Ferguson, T. Howard, and M. Likhachev, "Motion planning in urban environments," in *Proc. IEEE Intelligent Robots and Systems*, 2008, pp. 61–89.

[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] D. Kasper, G. Weidl, T. Dang, G. Breuel, A. Tamke, and W. Rosenstiel, "Object-oriented bayesian networks for detection of lane change maneuvers," in *Proc. IEEE Intelligent Vehicles Symposium*, 2011, pp. 673–678.

[8] S. Lefevre, C. Laugier, and I. Javier, "Exploiting map information for driver intention estimation at road intersections," in *Proc. IEEE Intelligent Vehicle Symposium*, 2011, pp. 583–588.

[9] W. Liu, H. Zhang, B. Duan, H. Yuan, and H. Zhao, "Vision-based real-time lane marking detection and tracking," in *Proc. IEEE Intelligent Transportation Systems*, 2008, pp. 49–54.

[10] 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.

[11] T. Naito, H. Morimoto, T. Ito, T. Yamamoto, and H. Fukumoto, "Driver turning behavior prediction at intersection improving prediction performance with driver behavior adaptation," in *Proc. IEEE JSAE*, 2007, pp. 115–122.

[12] 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.

[13] 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.

[14] 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.

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