

# A Learning Concept for Behavior Prediction in Traffic Situations

Regine Graf<sup>1</sup>, Hendrik Deusch<sup>1</sup>, Martin Fritzsche<sup>2</sup>, Klaus Dietmayer<sup>1</sup>

<sup>1</sup> Institute of Measurement, Control, and Microtechnology, University of Ulm, Ulm, Germany

regine.graf@uni-ulm.de, hendrik.deusch@uni-ulm.de, klaus.dietmayer@uni-ulm.de

<sup>2</sup> Department Environment Perception, Daimler AG, Ulm, Germany

martin.fritzsche@daimler.com

**Abstract**—Future driving assistance systems will need an increase ability to handle complex driving situations and to react appropriately according to situation criticality and requirements for risk minimization. Humans, driving on motorways, are able to judge, for example, cut-in situations of vehicles because of their experiences. The idea presented in this paper is to adapt these human abilities to technical systems and learn different situations over time. Case-Based Reasoning is applied to predict the behavior of road participants because it incorporates a learning aspect, based on knowledge acquired from the driving history. This concept facilitates recognition by matching actual driving situations against stored ones. In the first instance, the concept is evaluated on action prediction of vehicles on adjacent lanes on motorways and focuses on the aspect of vehicles cutting into the lane of the host vehicle.

## I. INTRODUCTION

Traffic situations can be interpreted by humans and can be assessed by them. This natural thinking process makes it possible for humans to react appropriately to critical situations. For assistance systems, the understanding of traffic situations is a crucial task in the future and therefore the behavior of other road participants have to be analysed. Studies of the behavior prediction of surrounding vehicles can be divided into logical, probabilistic or classification approaches. A logical representation of traffic situations is given by Mc Carthy [5] amongst others in 1963. In the last few years probabilistic approaches for behavior prediction and situation analysis became more common. In this connection, often Dynamic Bayesian Networks (DBN) or their special cases Hidden Markov Models (HMMs) are chosen [1] [2] [3]. In the field of probabilistic approaches for action recognition also combinations, for example a DBN with a Kalman Filter [4], are used or decision trees, Random Forests, for the classification of different situations is presented [6]. Besides probabilistic approaches, S. Vacek et al. presented a concept for the classification of different situations in urban areas based on Case-Based Reasoning. Probabilistic model approaches, for example Bayesian Networks, in comparison to the Case-Based Reasoning approach use an explicit model to solve a new problem. Bayesian Networks show many advantages, but it is difficult to express the knowledge in probability distributions [8] and the examination of the solution of the network can be quite complex. Also, it is difficult for a model to be an efficient representation and additionally to provide a wide coverage [11]. Case-Based Reasoning compares similar cases with

the new problem and solves the problem in assigning the solution of one known case to the new one. Thus, this concept requires a large storage space and cases have to be already stored to solve a problem. It has the advantages that it is easily expandable, partial matching is possible, a feedback of the system is integrated and it can cover idiosyncrasies of cases [11]. A probabilistic model approach can be integrated into the Case-Based Reasoning concept, to combine the advantages of both approaches [9].

This paper proposes an approach for situation interpretation using Case-Based Reasoning. Because of the presented advantages this approach is used and may be extended by a probabilistic model approach in the future. It differs from the presented concept in [7] by the definition of a case and additionally it is evaluated using real data. In the first instance, it is tested for cut-in situations on motorways. If on a motorway two cars are detected on the right lane by the host vehicle, and the car in front is slower, the system has to evaluate if the car behind overtakes or stays in line. In Fig. 1 the boxes around the vehicles illustrate the action prediction of the system for *no-overtaking* (green boxes) and *overtaking* (red box) situations. The aim is to detect the actual behavior early, so that early reactions to a critical situation are possible.

The paper is outlined as follows: Section II-A gives a system overview and the concept of Case-Based Reasoning is introduced in section II-B. The interpretation of situations with Case-Based Reasoning is presented in section III. In Section IV the concept is evaluated with real world situations and the paper concludes in Section V.



Fig. 1. The action prediction of the developed concept is shown. Cars which overtake are illustrated with a red, cars which do not overtake are illustrated with a green box around them.

## II. CONCEPT FOR MODELING HUMAN REASONING AND THINKING

### A. Overview

For behavior prediction of vehicles it is necessary to know the position of neighbouring vehicles and the lanes they are driving in. Therefore, a radar and a mono camera are used and the sensor information is fused to track the objects. From the radar signals we know not only the position of the vehicles, but also the velocity. A digital map or the detection and extraction of lane markings can be used to know which vehicle is driving in which lane. For this application, the lane markings are detected by the mono camera and they are extracted with an edge detection algorithm [10]. With this sensor set up, neighbouring vehicles can be detected and they can be assigned to lanes by the extracted lane markings.

### B. Case-Based Reasoning

To develop the concept, we first have to think of human decision processing while driving. Humans extract characteristics of the learned experiences in the past and compare them to present characteristics in current situations. With the help of this matching process, drivers are able to predict solutions of the behavior of other traffic participants. The real solution, the drivers know after the finished action, adds to the driver's experience. This human decision process is illustrated in Fig. 2 and can be compared to the Case-Based Reasoning process [11].

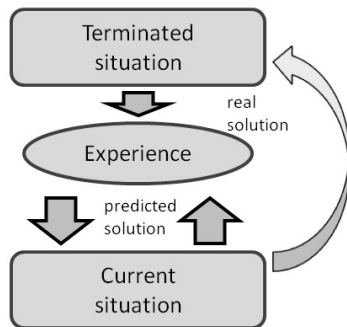


Fig. 2. The abstract process of human decision processing. The experience is used to predict the solution for new/current situations. At the end, the new situation with the real solution extends the experience for better decision making in the future.

Case-Based Reasoning (CBR) is a methodology which solves related problems with the knowledge of similar known ones. Because this concept is a natural learning mechanism, it offers the advantage to learn from experience and shows more robustness in comparison to rule-based systems upon encountering new situations.

The CBR concept, described in detail in [11], is a cyclic process. If the solution of a problem is known, the problem and its real solution are stored as a *case* into the *case base*. The case base is comparable to human's experience or their memory and a case is corresponding to a situation. The general process for solving a new problem with the help of CBR consists of the following steps:

- A) Accept and analyze
- B) Retrieve cases from memory
- C) Select most relevant cases
- D) Construct solution
- E) Evaluate solution
- F) Execute solution and analyze results
- G) Update memory

In the first step, we have to define a case and indexes, special features which can be used for the retrieval process. In the second step, we use these indexes to retrieve similar cases from the memory to solve a new problem. For this retrieval process various techniques are given in the literature [11], [15]. In the next step, the most relevant cases are determined to be the basis for the solution finding process. The solution for a current case is constructed by adapting the solution of the best fitting retrieved case. After this step, the proposed solution is tested, evaluated and potential strength and weaknesses are assessed. To analyze the proposed solution, a feedback from the system testing the solution is important and the predicted solution is compared with the real one. In the updating step, if the real and predicted solution are not the same, the CBR system includes the possibility in saving these inabilities in solution finding.

## III. CONCEPT FOR BEHAVIOR PREDICTION

The CBR process is presented in this section to predict driving behavior and is evaluated on a motorway scenario for cut-in situations.

### A. Accept and analyze

One possible definition of cases is to use a snapshot [7] as one case and combine this with a suitable hierarchy structure in the case base. In our approach, a case describes a whole situation, from the detection to the finished action. For each case a real solution can be determined, because the situation ends with the enforced action. Thus, a case is equivalent to a situation in the following. In the case base, similar cases for the new case have to be found and these similar features for the retrieving process have to be defined. Therefore, we sort all situations into three possible categories: *city*, *rural road* and *motorway*. This categorization can be used, because different traffic participants can appear in these different scenes and these can show different complexities. For situation interpretation, the constellation of the objects in the particular scene has to be known as this is significant for the behavior recognition of traffic participants. Hence, for the retrieval process we use two factors:

- 1) scene
- 2) object constellation

The complexity of the object constellation is dependent on the scene. For example, in inner-city scenarios specific places, like for example roundabouts and junctions have to be considered and the object constellation has to be divided into the position and classification of all relevant traffic participants. In general, on motorways the object constellation starts with detected objects on neighbouring lanes relative to the position of the host vehicle. Two cases

are possible:

- Right [r] : The detected object is on the right side in respect to the host vehicle  
 Left [l] : The detected object is on the left side in respect to the host vehicle

In the following, the object constellation is extended with respect to this so-called Observed Vehicle (OV). An OV is a vehicle on the lane to the left or right, with regard to the lane of the host vehicle and it has at least one vehicle in front. Normally, if a vehicle has no preceding one there is no reason to change its behavior. Exceptions can be seen, if vehicles on entrance lanes intend to change the lane. In these situations, the OV can cut in without having a car in front. These influences, taking place on lanes not being adjacent to the host vehicle, are neglected in the first instance. From the point of view of the OV, the following cases are relevant:

- Side [s] : A detected object is on the side of the Observed Vehicle and in front of the host vehicle  
 Front [f] : A detected object is in front of the Observed Vehicle

In Fig. 3(a) the OV is illustrated in red, the host vehicle in green and the object constellation starts with [r] because the OV is on the right lane with regard to the position of the host vehicle. In this illustrated example the white car is driving adjacent to the OV and in front of the host vehicle. Thus, the next factor for our object constellation is [s] for side. To distinguish this test situation from other situations the vehicle driving in front of the OV is considered in the object constellation with a [f]. Hence, the object constellation for this situation is [rsf].

Figure 3 also shows a possible change in object constellation during one situation (from 3(a) to 3(b)). The object constellation

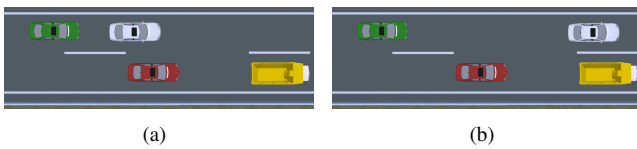


Fig. 3. In Fig. 3a to 3b a change in the object constellation can be seen. The white vehicle passes the red one and is no more in its relevant field. Thus, the white vehicle in 3b is no longer considered in the object constellation.

changes from [rsf] to [rf], because the white vehicle departs from the OV and is no more in the relevant area to restrict actions of the OV. In these cases we could extend the object constellation, but we decided to store these cases with object constellation [rsf], because if there is a vehicle adjacent to the OV, the actions of the OV are dependent on this car. This is also determined in the retrieving process.

With this indexing scheme, it is also possible to learn cut-in situations from vehicles on the left side. In this case, the OV would be on the left side and the object constellation would start with a [l] than a [r]. Until now, the presented concept concentrated on vehicles driving next to the host vehicle. Vehicles which change two lanes at once on motorways with

three lanes are not covered by the presented concept, but it is possible to extend it in an appropriate way, to introduce [rr] for the right lane on a three lane motorway, if the host vehicle is driving on the left lane.

#### B. Retrieve cases from memory

In this section we have to design our memory or case base to store features of terminated cases. Therefore, we use a relational data base and store indexing features, as scene and object constellation, for every case. Additionally, typical features, which are discussed in detail in III-C, are stored as binary objects into the data base. Thus, the relational database has 4 columns: an id column, a column for the scene, one for the object constellation and an additional one is integrated to store special features, an Object Identifier (OID) column. For the retrieval process, stored cases with the same indexing features as the current one are retrieved.

#### C. Select most relevant cases from memory

In this step only the most relevant cases are selected. Cases with the same scene and object constellation are possible similar cases for motorway scenarios and a further selection is not done. In the future, for inner-city scenarios, more complex object constellations in different levels can be used for selection. In the first level, in inner-city scenarios we group the cases in different places, like roundabouts or junctions. In the second level the object constellation of the objects in the different places is compared and in the third level the classification of the different objects is considered.

#### D. Construct solution

In this step the retrieved cases have to be compared to the current case and features for comparison have to be defined. Therefore, the relations between relevant vehicles for each case have to be studied. For our test situation three constellations (FRONT, EGO, SIDE) have to be considered and they are defined with reference to the OV.

- EGO : The relationship of the OV to the host vehicle  
 FRONT : The relationship of the OV to the vehicle driving in front of it  
 SIDE : The relationship of the OV to the vehicle driving in front of the host vehicle and adjacent to the OV

In the literature, characterizing factors for this scenario are the  $T_{TC}$  (time to collision), the  $T_N$  (net time gap) [4] and the lateral distance  $d_l$  [2] from the OV to its left lane marking. In this approach the basic factors  $d_{OVj}$  (distance) and  $v_{relOVj}$  (relative velocity) for  $j = 1, 2, \dots, N$ , with  $N$  as the number of relevant vehicles with respect to the OV, are analyzed for overtaking and no-overtaking situations. These factors are recorded over time for different overtaking and no-overtaking situations. By analyzing the several time courses, we find out that overtaking and no-overtaking situations can not be divided explicitly for one constellation, but the combination of all constellations shows better characteristics for the two actions. In most of the cases, the time courses of  $d_{OVj}$  and  $v_{relOVj}$  of the two possible actions (no-overtaking,

overtaking) show differences in the increase and in the levels of the parameters. However, these time courses for overtaking and no-overtaking situations can be quite similar, but with the help of the CBR it may be possible to learn these small differences or to determine constellations which can not be solved definitely. The lateral distance is not used in this application, because if a decreasing  $d_l$  is detected, it is often too late to react and the distance decrease is dependent on the driving behavior.

To store the analyzed characteristics of the time courses for all constellations, we first have to extract them. In the literature [16], for time series analysis a wide range of different approaches can be found. Successive values in time series can be correlated or time series can show a seasonal or trend component, as well as a cyclic course. For our application, we are focusing on trend extraction, because the level and the trend are significant features in analyzing overtaking and no-overtaking situations. In the field of time series transformation, in extracting relevant features from the time series, they can be aggregated or decomposed, filtered or approximated, smoothed or denoised to extract relevant information out of the time series [12].

To give examples for model approaches, for stationary time series common approaches are the Box-Jenkins models and the ARMA processes. For non stationary time series, the ARIMA model can be used. For smoothing time courses Kalman Filters [13] or triple exponential smoothing methods amongst others are possible approaches. In the field of intelligent data analysis Temporal Abstraction methods [18] (TAs) can be used for analyzing and interpreting time series. They have been already applied in combination with CBR [14] and can give an abstract description of time stamped data. Basic TAs detect clusters and aggregate events into episodes. In this context, state TAs detect episodes associated with qualitative levels and trend TAs detect patterns like decrease, increase and stationarity in time courses. In contrast, complex TAs aggregate episodes into more abstract ones by using the Allen temporal logic operators. In our application we used a combination of state and trend Temporal Abstraction, because the analysis of the time series for the several distances and relative velocities ensued that trends and levels of the time series are characteristics for behavior recognition of cut-in situations.

In our approach, in each time step the current time series grow and the extraction to determine the trend of the time series is done by using piecewise linear approximation. For each trend and level section the trend or level parameter, the start value and the time are stored. In the case of  $v_{relOVj}$ , additionally the distance value is stored. As a result, the time courses of the velocity and the distance are extracted into trend and level intervals. An example of the extraction is shown in Fig. 4. To extract the trend of the time series, three categories rising ( $r$ ), falling ( $f$ ) and constant ( $c$ ) are used. To determine the levels, we choose high ( $h$ ), middle ( $m$ ) and low ( $l$ ) for abstraction. The bounds of the different trend and level parameters are determined in the analyzing process at the beginning. An example for the used extraction

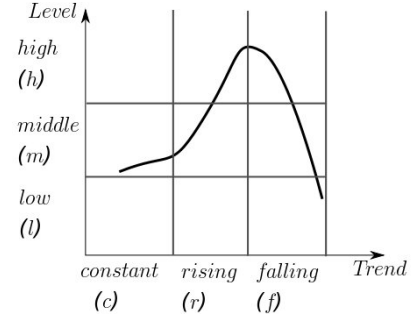


Fig. 4. This figure shows the chosen extraction of the time courses. The course is divided into different level and trend parameters. For this example, the trend is first constant, then it rises and falls. The sequence of the level is middle to high and from middle to low.

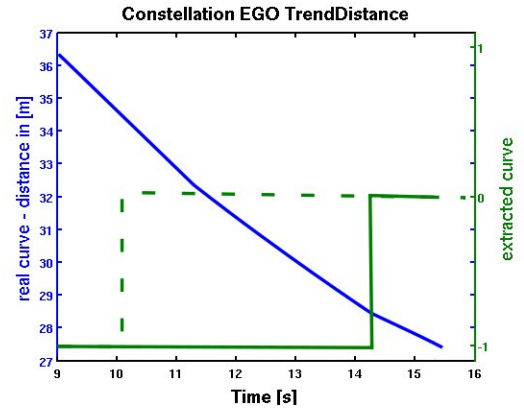


Fig. 5. An example of a distance time series for constellation EGO is plotted in blue. The trend intervals which will be stored into the data base are illustrated by the solid green line. For the rising trend value 1, for the constant trend value 0 and for a falling trend value -1, is used. The dotted green line shows another example of an extracted time course of a retrieved case.

of time series is presented in Fig. 5. After the extraction the comparison of the retrieved cases to the current one has to be made to calculate a similarity value to decide for the most similar case in the data base. If this case can be found the solution of this case can be assigned to the current case.

In the comparison step, for each constellation  $m$ , the comparison of  $i$  retrieved cases with the current one is done by comparing the trends and levels of  $d_{OVj}$  and  $v_{relOVj}$  time series. The number of all constellations is  $M$ . Therefore, in the first step we are searching a distance interval  $[d_{stored_o,m,i}, d_{stored_k,m,i}]$ , with  $o = 1, \dots, J-1$ ,  $J$ : Number of intervals,  $k = j+1, \dots, J$ , of each retrieved case which includes the current distance ( $d_{cur,m}$ ). If we find an interval ( $d_{cur,m} \in [d_{stored_o,m,i}, d_{stored_k,m,i}]$ ), we calculate the increase of the parameters in this interval of the current and retrieved cases, subtract them, and take the reciprocal value (equation 1, 2). The reciprocal value for a zero value after the subtraction is set to 1 and all values are normalized to 1. For the level features, we use the difference of the absolute values (equation 3). The same level of the retrieved and the current distance is a precondition in the interval searching process. Therefore, to determine the similarity of a retrieved

and a current case we calculate following trend and level values  $tv_{dist,m}$ ,  $tv_{vel,m}$ ,  $lv_{vel,m}$  for each constellation. The same values of all constellations are added and the average is calculated.

$$tv_{dist,m} = \frac{1}{\left| \left( \frac{\Delta d_{stored,m,i}}{\Delta t_{stored,m,i}} - \frac{\Delta d_{cur,m}}{\Delta t_{cur,m}} \right) \right|}} \quad (1)$$

$$tv_{vel,m} = \frac{1}{\left| \left( \frac{\Delta v_{stored,m,i}}{\Delta t_{stored,m,i}} - \frac{\Delta v_{cur,m}}{\Delta t_{cur,m}} \right) \right|}} \quad (2)$$

$$lv_{vel,m} = \frac{1}{|v_{stored,m,i} - v_{cur,m}|} \quad (3)$$

Thus, the three different values are multiplied and a similarity value

$$sv_i = \left( \frac{1}{M} \sum_{m=0}^M tv_{dist,m} \right) \cdot \left( \frac{1}{M} \sum_{m=0}^M tv_{vel,m} \right) \cdot \left( \frac{1}{M} \sum_{m=0}^M lv_{vel,m} \right) \quad (4)$$

for each retrieved case can be calculated. In each time step, the average over time of this similarity value for each retrieved case can be determined. If no interval can be found, the retrieved cases from the case base can not be compared with the present one and the similarity value is set nearly to zero.

An example for the comparing step is shown in Fig. 5. The extracted time course of the distance of the current case for one constellation (EGO) is illustrated by the solid green line. The time course of the distance of one retrieved case for the same constellation is shown by the dotted line. For the current distance (the last value of the solid line) a distance interval of the retrieved case can be found. The increase of the current distance interval of the current case (from 14.2s to 15.5s) and the increase of the found distance interval of the retrieved case (from 10.1s to 16s) can be calculated. Thus, the trend value for the distance value can be determined. With the presented concept we can determine a solution out of the similarity value of all retrieved cases. Out of  $i$  retrieved cases, the one which is most similar ( $\max(sv_i)$ ) to the present one, is used to predict the solution for the present case.

#### E. Evaluate Solution

In this step the solution, determined in the last step, is critically examined. We analyze, if the predicted solution is a possible solution for this case. In our approach, for the first instance only cases with the solution *overtaking* versus *no-overtaking* are stored. In extending the data base for cases with different solutions we have to verify the solution and use rules or environment perception for verification. For example, if the solution would be that a vehicle changes to the left lane and no left lane is present, the stored case with this solution is not a similar case to the current one. The lane detection in this case is used for examination.

#### F. Execute solution and analyze results

If the situation has terminated that means the OV overtook or the host vehicle passed the OV, we are able to determine

the real solution and compare the predicted solution with the real one. The overtaking process can be determined out of the position of the OV and the detected lane markings. The measurement uncertainties were considered to make a robust decision for the real solution if the vehicle really overtook. In the case that the OV did not overtake, the real solution is determined, if the host vehicle passes the OV. There are two cases possible:

- 1) The predicted solution is the same as the real one,
- 2) The predicted solution is wrong.

In the first case, the system was correct, the second case is presented in detail in the next subsection.

#### G. Update memory

In this last step of the CBR process we update the data base with the new case and the real solution for this case. If in the last step by analyzing the results a wrong predicted solution is noticed, a link between the new case and the case, the new case was wrongly compared with, is set. In the future, if a new case has similar features to both cases, with the help of this link we do not decide for one solution but decide at first for no solution. With an increasing data base it could be that certain features from new cases can be extracted, which make it possible to separate the cases that a solution finding is possible again.

### IV. EVALUATING THE CONCEPT OF BEHAVIOR PREDICTION

To evaluate the concept, we use sequences recorded on german motorways. Therefore, vehicles are detected by a radar and mono camera mounted in our vehicle and lane markings are detected with the help of the mono camera. For this approach, cases have first to be stored into the data base as basis of the solution finding process. This basis are 21 situations with 9 overtaking and 12 no-overtaking situations. The two parameters for evaluation, we are interested in, are the percentage of the correct decision ( $P_{CD}$ ), thus, in how many cases a correct decision is made by the developed concept, and at what time the correct decision was made before the situation terminated, thus the OV overtook or the host vehicle passed the OV. The time was measured to the moment the vehicle drives in the host vehicles lane. These two parameters were evaluated on real data and the results are shown in Table I. In the first sequence 26 situations, 12 overtaking and 14 no-overtaking situations, were detected. The  $P_{CD}$  for those 26 situations was 73% and the correct decision was made in average 2,1 seconds before the OV overtook or the host vehicle passed the OV. An additional sequence on a motorway was evaluated on the basis of now 47 cases in the data base. In this new sequence 33 situations, thereof 11 overtaking and 22 no-overtaking situations, were detected. For these 33 situations the  $P_{CD}$  was 79 % and the decision was made in average 2,3 seconds before the situation terminated. This examination shows that with a growing data base the positive decision rate increases. Although this result has to be proven also on a larger data base these examinations show promising results.



In Fig. 6 an example of the predicted solution over time can be seen. The star illustrates the real solution, the time course the predicted one (1 stands for *overtaking*, 0 for *no-overtaking*, and 2 for *no solution*). At the first time step no decision can be made, so the course starts with value 2. The figure shows that in the first instance the system decides for *no-overtaking*, because a car adjacent to the OV is driving. After a change in the object constellation (cf. Fig. 3) the concept predicts the correct solution more than 2s before the OV overtook. The figure shows that the decision process for solution finding is robust.

TABLE I  
EVALUATION OF 59 SITUATIONS IN 2 SEQUENCES.

Evaluated Situations	Cases in data base	$P_{CD}$ [%]	Time [s]
26	21	73	2,1
33	47	79	2,3

The concept has the disadvantage that at the beginning, when only a few cases are stored in the data base, the comparison of the current case with the stored ones can fail, because no similar case is stored. As a consequence, the system needs to have a sufficiently big data base for comparing current cases with stored ones. The first results are promising and for further work the data basis for evaluation will be extended.

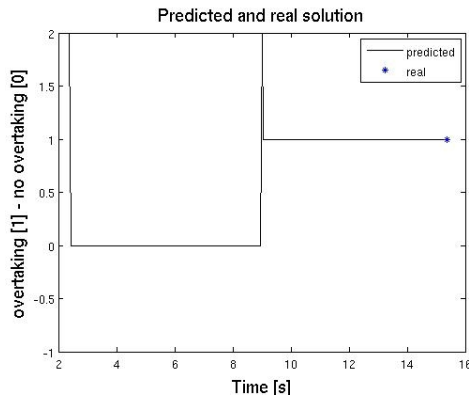


Fig. 6. An example for the predicted solution of an overtaking action over time. The predicted solution is abstracted: 0 stands for *no-overtaking*, 1 for *overtaking* and 2 for *no solution*. In the first time step, no prediction can be made. At the beginning, a car is driving adjacent to the Observed Vehicle. The system decides for *no-overtaking*. The car on the side passes by and the system decides for *overtaking* more than 2s before the Observed Vehicle overtook and the solution finding is robust.

## V. CONCLUSIONS AND FUTURE WORK

To conclude, we present a concept to adapt human reasoning and thinking to a system based on Case-Based Reasoning. CBR was explained and a concept for using this methodology for behavior prediction was shown. The approach was explained in considering situations in inner-cities, on rural roads and on motorways. To evaluate the presented concept, a testing scenario for behavior prediction on motorways was used. After analyzing typical features for this behavior prediction, of overtaking and no-overtaking vehicles, the CBR process was presented for a motorway

scenario but with a possible extension of the concept to inner-city and rural road scenarios. The evaluation showed that the concept can be used for behavior recognition and further evaluations will be done.

For further work, different types of situations, such as inner-city scenarios, will be integrated into the concept. Additionally, we will focus on situations which occur in certain fixed locations. Thus, georeferenced data will be integrated into the process, to warn the driver in certain locations that there is a critical situation. For further situations on motorways, georeferenced data can be used to identify entrance lanes and use this additional information to identify potentially dangerous situations.

## REFERENCES

- [1] D. Meyer-Delius, J. Sturm, W. Burgard, "Regression-Based Online Situation Recognition for Vehicular Traffic Scenarios" in *IEEE International Conference on Intelligent Robots*, St. Louis, USA, 2009.
- [2] T. Grindele, S. Brechtel, R. Dillmann, "A Probabilistic Model for Estimating Driver Behaviours and Vehicle Trajectories in Traffic Environments", in *IEEE Conference on Intelligent Transportation Systems*, Madeira Island, Portugal, 2010.
- [3] J. Firl, Q. Tran, "Probabilistic Maneuver Prediction in Traffic Scenarios" in *European Conference on Mobile Robots*, Örebro, Sweden, 2011.
- [4] I. Dagli, G. Breuel, H. Schittenhelm, A. Schanz, "Cutting-in Vehicle Recognition for ACC Systems - Towards Feasible Situation Analysis Methodologies" in *IEEE Intelligent Vehicles Symposium*, Parma, Italy, 2004.
- [5] J. McCarthy, "Situations, actions, and causal laws". Technical report, Stanford Department of Computer Science, Stanford, 1963.
- [6] M. Reichel, M. Botsch, R. Rauschecker, K.-H. Siedersberger, M. Maurer, "Situation Aspect Modelling and Classification Using the Scenario Based Random Forest Algorithm for Convoy Merging Situations" in *IEEE Conference on Intelligent Transportation Systems*, Madeira Island, Portugal, 2010.
- [7] S. Vacek, T. Grindele, J. Marius Zöllner, R. Dillmann, "Situation classification for cognitive automobiles using case-based reasoning" in *IEEE Intelligent Vehicles Symposium*, Istanbul, Turkey, 2007.
- [8] L. Uusitalo, "Advantages and challenges of Bayesian networks in environmental modelling", *Ecological Modelling*, 2007.
- [9] H. Tirri, P. Kontkanen and P. Myllymäki, "A Bayesian Framework for Case-Based Reasoning", *Advances in Case-Based Reasoning*, 1996.
- [10] A. Wimmer, T. Jungel, M. Glück and K. Dietmayer, "Automatic Generation of a Highly Accurate Map for Driver Assistance Systems in Road Construction Sites" in *IEEE Proceedings of Intelligent Vehicles Symposium*, San Diego, CA, 2010.
- [11] J. Kolodner, "Case-Based Reasoning", *Morgan Kaufmann Publishers*, San Mateo, CA, 1993.
- [12] T. Meinl, "A Novel Wavelet based approach for time series data analysis", Dissertation, Karlsruhe, 2011.
- [13] T. Alexandrov, S. Bianconcini, E. B. Dagum, P. Maass, T. McElroy, "A review of some modern approaches to the problem of trend extraction", *Research report series (Statistics 2008-3)*, U.S. Census Bureau, Washington D.C., 2008.
- [14] S. Montani and L. Portinale, "Case Based Representation and Retrieval with Time Dependent Features", *6th International Conference on Case-Based Reasoning*, Chicago, IL, ICCBR 2005.
- [15] I. Watson, "Applying Case-Based Reasoning", *Morgan Kaufmann Publishers*, San Francisco, CA, 1997.
- [16] J.-P. Kreiß, G. Neuhaus, "Einführung in die Zeitreihenanalyse", *Springer Verlag*, Berlin Heidelberg, 2006.
- [17] I. Batal, L. Sacchi, R. Bellazzi and M. Hausknecht, "Multivariate Time Series Classification with Temporal Abstraction", *Association for the Advancement of Artificial Intelligence*, 2009.
- [18] R. Bellazzi, C. Larizza, P. Magni, S. Montani, M. Stefanelli, "Intelligent analysis of clinical time series: an application in the diabetes mellitus domain", *Artificial Intelligence in Medicine*, 2000.