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

An Integrated ADAS for Assessing Risky Situations in Urban Driving

Thomas H Weisswange, Bram Bolder, Jannik Fritsch, Stephan Hasler, Christian Goerick

2013

Preprint:

This is an accepted article published in IEEE Intelligent Vehicles Symposium 2013. The final authenticated version is available online at: https://doi.org/10.1109/IVS.2013.6629484

An Integrated ADAS for Assessing Risky Situations in Urban Driving

Thomas H. Weisswange*, Bram Bolder*, Jannik Fritsch*, Stephan Hasler* and Christian Goerick

***Abstract*— Advanced Driver Assistance Systems (ADAS) are becoming more and more popular. Many of these systems though are limited to specific scenes and often detect risky situations very late so they can only mitigate accidents. These effects are mainly caused by the use of simple physical prediction methods, e.g. to estimate the time-to-contact with another vehicle. In this paper we show an ADAS that extends the functionality of physical collision warnings by additionally estimating potential risks based on implicit predictions. As an example we demonstrate the use of vehicle orientation information for classifying situations. Through this, we can in particular assess the risk of static cars, for which physical prediction does not apply, but which can nevertheless easily cause an accident if they start moving into our driving corridor. The proposed system is evaluated online in a test car and is shown to reliably detect classical risky situations as well as those involving static cars.**

I. Introduction

In order to further reduce fatalities and make driving less stressful, modern cars are equipped with Advanced Driver Assistance Systems (ADAS). Depending on the function, those systems warn the driver of dangerous situations via some human machine interface (HMI) or autonomously control some of the car's dynamics.

However, in most cases only a very small portion of the broad range of driving situations is captured by the systems. Their functions usually target situations that lie within a very small temporal distance and pose a direct physical threat to the vehicle and its passengers. This limitation mainly results from the prediction mechanisms that are used, which rely on purely physical parameters, i.e. simply extend the current motion vector of the egocar (and potentially of other traffic participants) into the future.

This works well for environments with strongly regulated dynamics, as can be seen for example in the success of systems doing advanced cruise control (ACC) on highways (e.g. [1], [2]). Similarly, Schubert and colleagues showed a successful system using a Bayesian network to decide for the safety of an automatic lane change on a highway [3]. In other cases, static objects are considered dangerous if they are detected on the projected trajectory of the egocar, like e.g. in pedestrian warning [4], [5] and other collision mitigation systems [6]. ADAS applicable in a much broader range of situations often limit the area in which they work to the front of the car where a simple time-to-contact (TTC) feature usually suffices [7].

Physical prediction (like e.g. the TTC) can be problematic, because most dynamic entities in a traffic scene adapt their behavior to various types of context information. They will e.g. stop at red traffic lights, follow a curved road or give way to other entities, all of which leads to a deviation from a purely ballistic prediction. In the analysis of different error situations encountered by the autonomous vehicles participating in the DARPA Urban Challenge this was considered one major cause [8]. The awareness of this limitation leads to the fact that many ADAS warn or intervene only shortly before an accident and therefore often can only mitigate consequences. In addition, prediction in existing systems relies heavily on motion information (see e.g. [9]), which leaves static objects without any predictive information in a physical sense. These might nevertheless start to move into our driving corridor suddenly and should therefore, depending on the situation, also be considered as a potential threat. Figure 1 shows an example of a relevant situation, where a car is standing at an unsigned incoming street.

One possible way to increase the prediction horizon and reliability of ADAS is to explicitly incorporate context information. One can for example model the future motion vector of cars along the road course (e.g. [10], [11]). Alternatively it can be tried to model the intention of other drivers and then follow the respective model [12], or even model any possible behavior [13], [14]. But this can drastically increase the requirements for sensory processing and for integration of all this information, because this has to be done for every entity in a scene. This aplies even more, if the interaction of different traffic participants is to be included.



Fig. 1. Exampe scene for a risky inner city situation involving a static car on an incoming street.

In this paper we will instead propose an ADAS that concentrates on implicit predictions, by means of selecting simple sets of features that convey information about possible future states of the traffic scene. We also avoid the increase in complexity that results from the prediction of the full future

All authors are with the Honda Research Institute Europe GmbH, Offenbach am Main, Germany Firstname.Lastname@honda-ri.de
*These authors contributed equally.

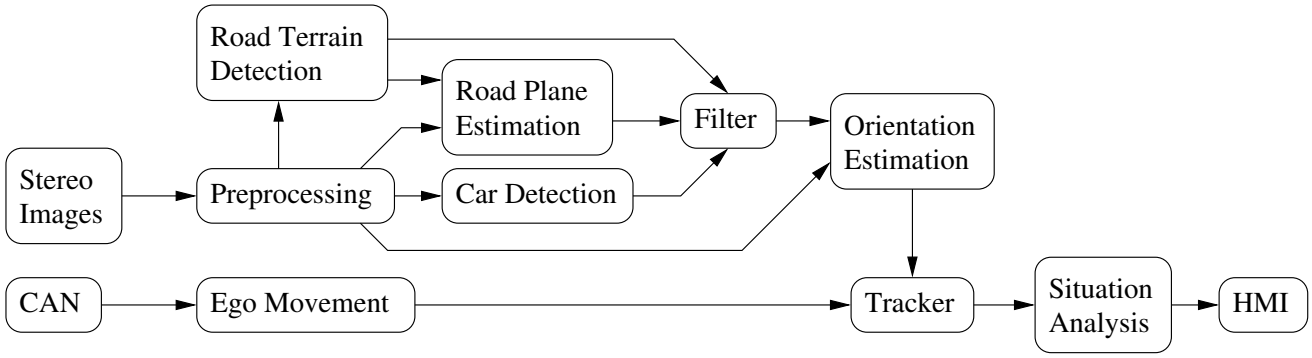

Fig. 2. Overview over the system components and the processing flow.

state for every entity by limiting the set of future states to those relevant for our system. The first version of our system that we will present in the following extends existing ADAS functonalities by also incorporating the risk introduced by static entities using orientation information.

The remainder of the paper is structured the following way: First we will introduce the system architecture which is designed to be easily extended with additional feature detection modules to support the recognition of more situation types in the future. We will continue with a short description of the main modules used for the current functionality. Afterwards we will evaluate the system on a number of real world test drives and conclude the paper with a discussion about benefits and limits of our approach.

II. System Overview

Our system architecture is designed to serve as a scientific platform able to support a variety of different tasks in a modular fashion. The presented instance is tuned to have a sufficient online processing speed. For an overview of the system architecture, please refer to Fig. 2.

Our car is equipped with a stereo camera system and with access to the CAN bus to extract the vehicle speed and the yaw rate. The camera images and the CAN bus information are timestamped to allow asynchronous processing of these two data streams. The CAN data is used to estimate the movement of the ego vehicle. The images are rectified and the left image is provided to the car and road terrain detection modules. Additionally, a stereo disparity map is calculated on a reduced image resolution. The road plane is estimated using the stereo information located at pixels which the road terrain detection classifies as belonging to the road.

The car detection searches the whole image at different scales with a simple heuristic search restriction regarding the assumed vertical locations for the different scales. Since this does not prevent enough false detections, an additional heuristics filter uses the overlap with the road terrain, the height above ground, and the metric size. For all remaining detections, the orientation is estimated. The tracker then stabilizes these detections both in space and time, before situation analysis is performed. If a risky situation is detected, this is communicated to the driver via an HMI display.

The whole system is implemented on 4 onboard computers, two car PCs with a Core2 Quad CPU and two laptops with a Core i7 CPU and an Nvidia GeForce 580M GPU bearing the high load of the car detection with orientation estimation and the road terrain detection. All GPU programming is done using OpenCL. The system is implemented and distributed over the different computers using the RTBOS middleware [15]. The stereo cameras have a baseline of 24 cm and a resolution of 1280×1024. Since we target inner city scenarios, we need a large opening angle ($63°$), which in turn requires the high resolution to enhance our perception at larger distances.

III. System Components

A. Road Terrain Detection

One very important source of information for a driver assistance system is the road course. On highways the detection of lane markings is sufficient, but in general the extraction of the road course is more challenging. Lane boundaries can consist of bad lane markings, curb stones, parked cars, and implicit boundaries by other ground material like, e.g., grass or cobblestone.

In order to capture the road terrain independent of the boundary type, we developed an approach that improves an appearance-based classification by incorporating the spatial layout of the scene [16]. In a first stage, the system represents visual properties of the road surface, the boundary, and lane marking elements in confidence maps based on analyzing local visual features. From these confidence maps SPatial RAY (SPRAY) features that incorporate properties of the global environment are calculated. After this two-staged extraction process, a decision is taken by applying a boosting classifier for road terrain detection that is trained on annotated ground truth data. Fig. 3 shows an overview of the processing steps.

Based on training data, the proposed approach can learn the spatial layout of driving scenes as the features implicitly represent both local visual properties and their spatial layout. We can train the approach for different types of road terrain, e.g. road area or ego-lane.

Depending on the type of situation analysis one of these categories can be chosen. For the situation analysis described in this paper, the *road-area* segmentation results are used.

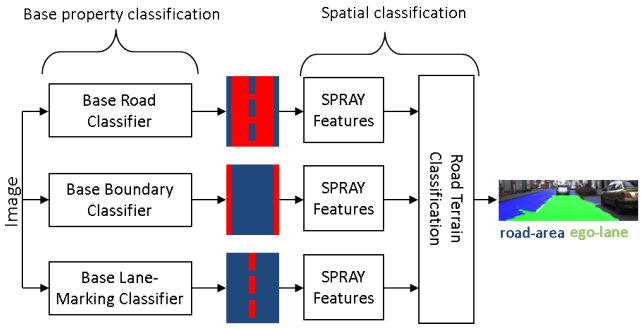

Fig. 3. Two-staged processing for road terrain detection.

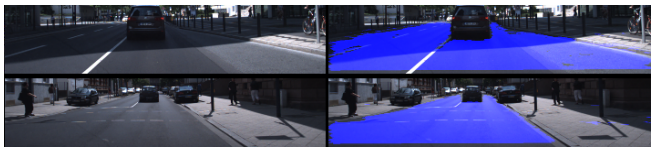

Fig. 4. Two examples for *road-area* segmentation results.

Fig. 4 shows some example results for the road area rendered into the coresponding real world image.

B. Car Detection

To determine the position and orientation of cars in the scene we apply a three stage process. First an appearance based detector delivers a confidence map over the scene, where the local maxima denote possible car locations. Then we use some heuristics to reject implausible hypotheses based on depth and context information. Finally, an orientation classifier is applied on the remaining high confident image regions.

For the detection the gray-scale image is projected to the analytic feature representation proposed in [17]. This feature hierarchy first calculates SIFT descriptors on a regular grid. Each descriptor is matched to a set of nearly 100 analytic features which are the result of a supervised selection process. After this, a local maximum operation is performed to increase robustness against small translations. This is in contrast to [17], where for an identification task the global maximum per feature was determined. Finally, on top of the hierarchy a single layer perceptron (SLP) is used in a convolutive manner to generate the car response map.

Initially, the SLP is trained to separate few thousands car segments from a larger set of randomly chosen non-car segments. During several bootstrapping steps more negative examples are generated for the training environment. Please note that we make use of only one SLP node to detect all possible car poses. This is very efficient but on the other hand a very demanding restriction. To deal with cars at different distances, we train analytic features and SLPs on three different segment sizes and use them on the largest image resolution, while the largest template is also used on successively reduced resolutions.

In a heuristics step, for each candidate region we check a number of size and position features (see section II) and a rejection is performed based on hand-tuned thresholds. Finally, a local competition removes further weak hypotheses if they have a too strong overlap with more confident ones.

We tested the standalone detector with heuristics on 30 minutes of video data with one labeled frame per second (1800 images). The streams covered different weather conditions (sunny, rainy, overcast) and scene types (city, highway, rural, industry). For the evaluation of the receiver operating characteristic (ROC) shown in Fig. 5, the common 50% mutual overlap criterion between labels and detections was used. On a different benchmark we confirmed that the performance of our detector is comparable to that of the standard histogram of gradients framework [18]. However, the main conceptual difference is the fact that [18] applied an SVM with many support vectors directly on top of the local gradient histograms, while we use an additional projection to the analytic features and a simple SLP as classifier. Figure 5 also clearly shows the dependency of the performance on the visibility of the cars. Our method is able to deal to some degree with medium occlusions, but if only a small part of the object is in the image it fails most of the time. We therefore decided to do the evaluation of the full system in an environment with limited occlusions. For those a working point with 0.064 false positives per image was chosen. The corresponding recall is below 50% but the tracking framework presented in the next subsection is able to improve this for the final application.

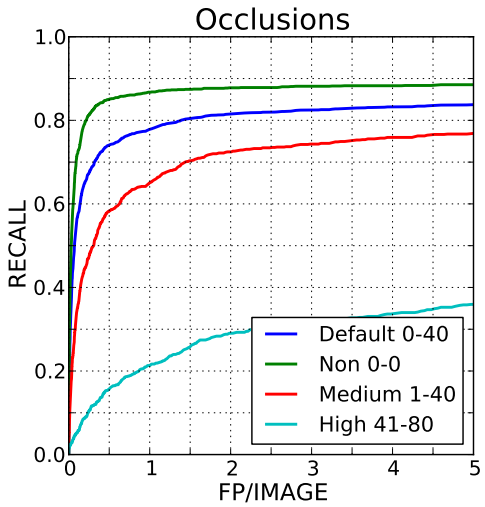

Fig. 5. ROC plots for the car detection module with heuristics. The performance strongly degrades with the amount of occlusion of the cars.

For the orientation estimation, we use the same approach as for the car recognition but on a limited region, an alternative approach using a tree-structured classifier was proposed in [19]. Each detected car region is cropped from the scene and normalized to a given segment size. After transformation to the analytic feature representation an SLP with four output nodes delivers a confidence for each of the car poses: left, front, back, and right. For the first training of the orientation SLP we used sequences of different cars ro-

tating on a turntable, which made labeling of the orientations particularly easy. Afterwards an additional bootstrapping step was performed with segments that were produced by the car recognition module described above on a variety of real world recordings and then labeled by hand. For a combined test set of real world and turntable segments the orientation classification rate was 96% without any temporal integration.

C. Tracking

For each processed image, car detections are generated and their orientation is estimated. To allow the tracking of these entities, we thus need a method that can handle multiple simultaneous detections. In the family of probabilistic data association (PDA) trackers (see [20] for an introduction), the joint integrated PDA (JIPDA) is recently becoming popular. For a simpler implementation and in order to support real-time processing, we instead opted for the linear multi-target IPDA (LMIPDA) as described by Mušicki [21].

To allow for an easy handling at the image borders and for controlling depth uncertainty for large depths, the car detections provided to the tracker consist of the minimum and maximum image coordinates and the disparity:

$$m_i = (u_{\min,i}, u_{\max,i}, v_{\min,i}, v_{\max,i}, d_i)\,. \tag{1}$$

In case the measurement overlaps with the image border, the corresponding entries are associated with an infinite variance.

The coordinate system selected for tracking has its origin on the road plane directly below the center of the rear axle of the ego car. Since this coordinate system moves with the ego vehicle, all tracked entities are subject to ego motion compensation in each timestep. Ego motion is estimated from speed and yaw rate information provided by the vehicle via the CAN bus. A simple single track model is assumed — we increased the tracker's covariances to compensate for deviations from this model. A tracking state consists of 3D position, calculated from disparity, 2D velocity on the plane and the width and height of an entity's bounding box.

$$s_j = (\vec{p}_j, \vec{v}_j, w_j, h_j)\,. \tag{2}$$

Propagation is done assuming a linear movement model with constant velocity, size, and elevation. Since the mapping of these measurements into our reference coordinate system is nonlinear, we apply an unscented Kalman filter (UKF) as the tracking model inside the LMIPDA framework.

We initialize tracks using two detections at positions not too far apart. This reduces the number of false positives of the car detection to enter the tracker and the evaluation. Additionally, we use the difference in positions of these two measurements to initialize the velocity. This also reduces the time till the velocity converges.

The orientation of the car detections has the form of a vector of confidences for the 4 canonical directions:

$$O_i = (c_{0^\circ,i}, c_{90^\circ,i}, c_{180^\circ,i}, c_{270^\circ,i})\,,\ 0 \le c_{k,i} \le 1. \tag{3}$$

Due to the special nature of this data (periodic boundary conditions, high correlation between confidences 180° apart), orientation is not tracked inside the UKF and LMIPDA framework. Instead, the orientation of a tracked entity is extracted from the sum of the last 7 orientation vectors (i.e. the last 0.8 s). Depending on the needs, either the maximum entry or an interpolation can be used to extract the final entity orientation. Since this orientation is estimated in image space we finally transform it into a global orientation within the egocar-centered coordinate system.

D. Situation Analysis

All the information from the lower level processing has to be combined and interpreted with respect to the system's desired functionality. Besides general information from physical prediction through tracking and recognition of cars, we focus on features that allow risk estimation for static cars. In this work we use the distance to the road area and the orientation of cars relative to our egocar. In contrast to other visual ADASs which concentrate on cars directly in front of the egocar [7], we additionally include the areas to the front left and front right side, since those commonly contain potentially dangerous static entities (besides the obvious "static object on road" case) as can be seen in Fig. 1.

We use the decision tree shown in Fig. 6 to classify each tracked entity into potentially dangerous or not, where each layer tests a certain feature against a threshold. First we check coarse relative lateral position (as *left/right of*, or *in* our driving corridor). We then compute a TTC based on physical predictions using the motion vectors derived by the tracking module (see section III-C). This helps us to cover both, cases with small distance to a static or moving obstacle in our way and cases with vehicles moving from the side into our lane. We continue with a check for the object type, which in our case will be always a car, but we planned the system with general objects in mind. For all cars we will compute the orientation using the method described in section III-B and check if they are oriented towards our driving corridor. Finally we compute the distance of a car to the visible road area. Based on the type of leaf node reached, cars will be given a risk value, and we will then highlight the most risky ones on a warning display. The decision tree can be easily adapted to also include e.g. pedestrians or additional features like road layout or traffic signs. As thresholds for the continous features we use a typical lane width (3m) around our car to define the center position (left and right respectively), a TTC of 1.5s and a distance of 1m to the road area.

Figure 7 shows the different types of situations that we communicate to the driver, together with an abstract risk quantification. First of all we highlight any car in front of our egocar within a certain distance (7A-C). If the TTC to such a car becomes smaller than a threshold, its color will turn red to demonstrate an increased risk. Those front cars that directly face us are always considered more risky (and therefore will always be red).

In the display we do not explicitly differentiate between moving and static cars that are detected to the sides of our

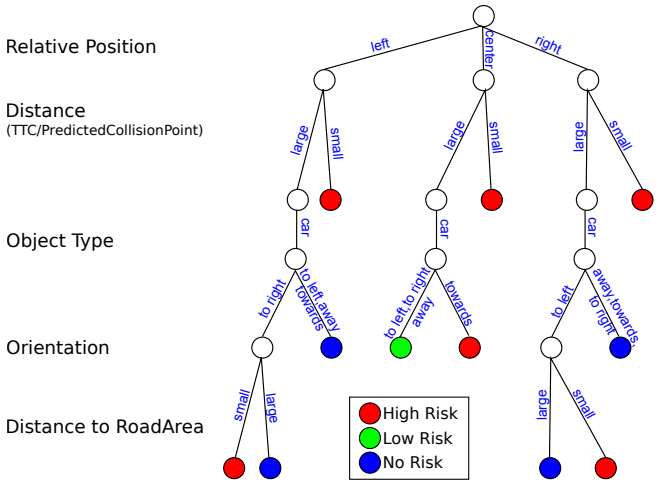

Fig. 6. Decision tree structure for evaluation of risk of all entities in a scene. At each node the respective feature value will be compared to a threshold and passed on to the next layer. If a leaf node is reached, the color determines if it is risky for the egocar or not. Note that all cars that are in front of us and do not point towards us, are given a small risk value, to include the potential thread of sudden braking.

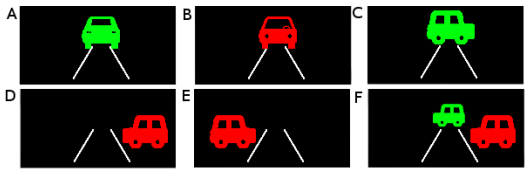

Fig. 7. Symbols showing different risk and situation estimations. For a detailed explanation please see text.

driving corridor[1]. Cars that are oriented towards our driving lane are potentially dangerous if they lie close to the road area, because they could start to move and would then collide with the egocar. We display those in red (7D-E), whereas cars to the side with other orientations will not be displayed. Moving objects will always be highlighted if a continuation of their current dynamics will intersect with our driving corridor within a certain distance. If multiple entities get assigned a risk we also communicate an abstraction of the relative distance between them by a scaling of the warning symbols (7F), but only, if they are detected in different areas of the scene (e.g. center and on right).

IV. System Performance and Evaluation

A. Function Evaluation

In general, to make a comparison with similar systems easier, it would be interesting to evaluate our system on a public real world dataset. This has already been established for elementary functions like optical flow, stereo and object detection (e.g. in the KITTI dataset [22]). However the frequency of risky situations relevant for evaluating a complete ADAS is too low. We therefore chose to evaluate our system on example situations setup on a local road round course. This setup includes many features from inner city environments, e.g. many parked cars, complex backgrounds, varying lighting/shadow conditions, but we were also able to get a high frequency of relevant situations (please see Fig. 8 for an impression of these situations and the setting).

We run the system in real time on the car while driving similar scenes at a variety of weather conditions, that include sunny, overcast, cloudy and snowy. Overall we drove for 22 minutes distributed over seven experiments and encountered 767 cars, 67 were identified as risky with respect to our targets. Speed in the tests was between 6 and 11 m/s. For some example pictures of the running system taken from the in-car screen see Fig. 8.

We evaluated the results in an event-based way, where every car in the scene is considered an event and any warning signal for a given car will be considered in the analysis. Out of all cars 4.5% where falsely classified as risky, that is they were shown in the HMI for at least one frame. An analysis on the error types showed that more than half of these false positives occured because of a wrong orientation estimation in some of the frames for a given event in particular for partially occluded cars. The remaining errors originated either in a mislocalization of an entity or in a misdetection of the car recognizer. For true positives we require an initial detection with a projected TTC of at least 2 sec (minus the processing latency – see section IV-B) and a continuous warning for the remaining time. 95.5% of the 67 risky situations were correctly detected. The median of the detection distance for true positives was 29.8 m with a minimum of 12.7 m at 6 m/s speed. The three false negatives were due to the negative influence of strong sunshine reflections on the car recognition module in two of the sessions.

B. Computational Performance

Our system runs with 9 frames per second on the hardware installed in the car. The total delay between the start of the image exposure until this image is processed by the situation analysis is roughly 320 ms. A calculated TTC of 2 seconds is thus actually only 1.7 s away at the time it is evaluated. Since we only evaluate entities that were seen at least twice, the latency between the first time a car appears in the image and the time a warning is be generated increases to 430 ms. Assuming we just missed an entity that appeared directly after an image exposure completes, the maximum warning latency will then be 540 ms.

The main computational ressources are used for car detection, road terrain detection, and stereo calculation. Since these three have no interdependencies, they can run in parallel. Therefore the processing speed is limited by the slowest of these three, which currently is the car detection (roughly 100 ms).

V. Conclusion and Future Works

In this paper we have provided an overview of our system approach for future ADAS and its current performance.

[1]Note however that this is only because we want to keep the warning scheme simple, all information for a more detailed risk assignment is present in the system.

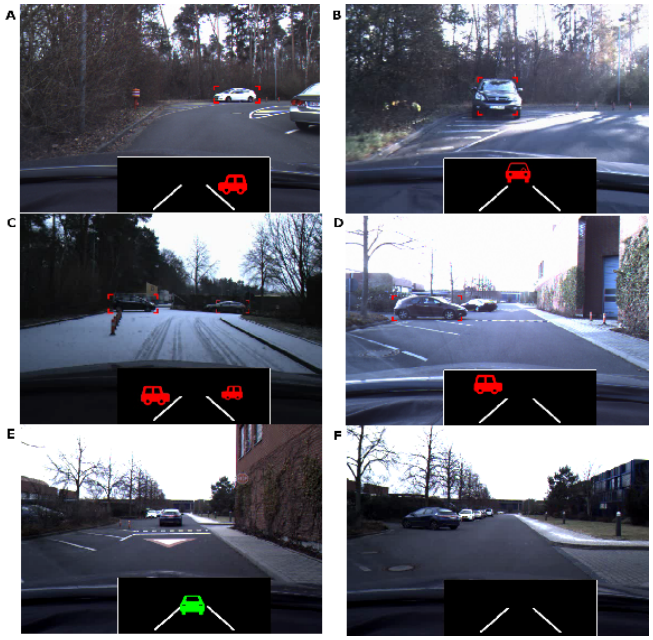

Fig. 8. Example screenshots of the warning display for some prototypical situations from different test streams. **A**: A vehicle pointing towards the driving corridor from the right. **B**: A vehicle pointing towards the driving corridor from straight ahead. **C**:A simulated crossing situation with cars pointing towards the corridor from both sides. Relative distance information is also shown on the display. **D**:A car waiting at an incoming street. **E**: A car driving in front of the egocar with similar speed. **F**: A car turning away from the driving corridor imposes no risk.

Through combining several state-of-the-art perception modules with a generic situation analysis, our system can not only handle classical risky situations but can also infer the potential danger of static cars facing the road. For this, we enhanced vehicle recognition with an orientation detection stage and created a decision tree for evaluating relevant (i.e., risky) situations. While the evaluation of the system on a controlled round course shows very good results, a remaining challenge are the many (partial) vehicle occlusions in real world traffic scenes. Currently, these result in missed laterally oriented vehicles and, therefore, in a late or absent driver warning. Ongoing work aims at improving the system in this respect and at extending the situation analysis to incorporate more spatial aspects.

VI. ACKNOWLEDGMENTS

The authors would like to thank Mark Dunn, Heiko Wersing and Tobias Kühnl for both technical and conceptual support.

REFERENCES

[1] P. Fancher, Z. Bareket, S. Bogard, C. MacAdam, and R. Ervin, "Tests Characterizing Performance of an Adaptive Cruise Control System," *SAE Technical Paper*, no. 970458, 1997.

[2] A. Vahidi and A. Eskandarian, "Research advances in intelligent collision avoidance and adaptive cruise control," *IEEE Transactions on Intelligent Transportation Systems*, vol. 4, no. 3, pp. 143–153, Sept. 2003.

[3] R. Schubert, K. Schulze, and G. Wanielik, "Situation Assessment for Automatic Lane-Change Maneuvers," *IEEE Transactions on Intelligent Transportation Systems*, vol. 11, no. 3, pp. 607–616, Sept. 2010.

[4] E. Coelingh, A. Eidehall, and M. Bengtsson, "Collision Warning with Full Auto Brake and Pedestrian Detection - a practical example of Automatic Emergency Braking," in *13th IEEE Intelligent Transportation Systems Conference (ITSC 2010)*. IEEE, Sept. 2010, pp. 155–160.

[5] C. G. Keller, T. Dang, H. Fritz, A. Joos, C. Rabe, and D. M. Gavrila, "Active Pedestrian Safety by Automatic Braking and Evasive Steering," *IEEE Transactions on Intelligent Transportation Systems*, vol. 12, no. 4, pp. 1292–1304, Dec. 2011.

[6] J. Hillenbrand, A. M. Spieker, and K. Kroschel, "A Multilevel Collision Mitigation Approach - Its Situation Assessment, Decision Making, and Performance Tradeoffs," *IEEE Transactions on Intelligent Transportation Systems*, vol. 7, no. 4, pp. 528–540, Dec. 2006.

[7] I. Gat, M. Benady, and A. Shashua, "A monocular vision advance warning system for the automotive aftermarket," in *SAE World Congress & Exhibition*, 2005.

[8] M. Campbell, M. Egerstedt, J. P. How, and R. M. Murray, "Autonomous driving in urban environments: approaches, lessons and challenges." *Philosophical Transactions of the Royal Society A: Mathematical, Physical, and Engineering Sciences*, vol. 368, no. 1928, pp. 4649–72, Oct. 2010.

[9] U. Franke and S. Heinrich, "Fast obstacle detection for urban traffic situations," *IEEE Transactions on Intelligent Transportation Systems*, vol. 3, no. 3, pp. 173–181, Sept. 2002.

[10] A. Alin, M. V. Butz, and J. Fritsch, "Incorporating environmental knowledge into Bayesian filtering using attractor functions," in *IEEE Intelligent Vehicles Symposium (IV 2012)*. IEEE, June 2012, pp. 476–481.

[11] J. Wiest, M. Höffken, U. Kreßel, and K. Dietmayer, "Probabilistic trajectory prediction with Gaussian mixture models," in *IEEE Intelligent Vehicles Symposium (IV 2012)*. IEEE, 2012, pp. 141–146.

[12] M. Liebner, M. Baumann, F. Klanner, and C. Stiller, "Driver intent inference at urban intersections using the intelligent driver model," in *IEEE Intelligent Vehicles Symposium (IV 2012)*. IEEE, June 2012, pp. 1162–1167.

[13] A. Broadhurst, S. Baker, and T. Kanade, "Monte Carlo road safety reasoning," in *IEEE Intelligent Vehicles Symposium (IV 2005)*. IEEE, June 2005, pp. 319–324.

[14] N. Kaempchen, B. Schiele, and K. Dietmayer, "Situation Assessment of an Autonomous Emergency Brake for Arbitrary Vehicle-to-Vehicle Collision Scenarios," *IEEE Transactions on Intelligent Transportation Systems*, vol. 10, no. 4, pp. 678–687, Dec. 2009.

[15] A. Ceravola, F. Joublin, M. Dunn, J. Eggert, M. Stein, and C. Goerick, "Integrated Research and Development Environment for Real-Time Distributed Embodied Intelligent Systems," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2006)*. IEEE, Oct. 2006, pp. 1631–1637.

[16] T. Kuehnl, F. Kummert, and J. Fritsch, "Spatial ray features for real-time ego-lane extraction," in *IEEE Int. Conf. on Intelligent Transportation Systems (ITSC)*, Anchorage, Alaska, USA, 2012, pp. 288–293.

[17] S. Hasler, H. Wersing, S. Kirstein, and E. Körner, "Large-Scale Real-Time Object Identification Based on Analytic Features," in *19th International Conference on Artificial Neural Networks (ICANN 2009)*, ser. Lecture Notes in Computer Science, vol. 5769. Springer Berlin Heidelberg, 2009, pp. 663–672.

[18] N. Dalal and B. Triggs, "Histograms of Oriented Gradients for Human Detection," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 05)*, vol. 1. IEEE, 2005, pp. 886–893.

[19] M. Gabb, O. Lohlein, M. Oberlander, and G. Heidemann, "Efficient monocular vehicle orientation estimation using a tree-based classifier," in *IEEE Intelligent Vehicles Symposium (IV 2011)*. IEEE, June 2011, pp. 308–313.

[20] Y. Bar-Shalom, F. Daum, and J. Huang, "The probabilistic data association filter," in *IEEE Control Systems Magazine*, vol. 29, no. 6, 2009, pp. 82–100.

[21] D. Mušicki and B. La Scala, "Multi-target tracking in clutter without measurement assignment," in *IEEE Transactions on Aerospace and Electronic Systems*, vol. 44, no. 3, 2008, pp. 877–896.

[22] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? The KITTI vision benchmark suite," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2012)*. IEEE, June 2012, pp. 3354–3361.