

Design and Capabilities of the Munich Cognitive Automobile

Matthias Goebel, Matthias Althoff, Martin Buss, Georg Färber, Falk Hecker, Bernd Heißing, Sven Kraus, Robert Nagel, Fernando Puente León, Florian Rattei, Martin Russ, Michael Schweitzer, Michael Thuy, Cheng Wang, Hans Joachim Wuensche

Abstract—This paper presents the design of the Cognitive Automobile in Munich. The focus of the capabilities shown here is the navigation on highways and rural roads. The emphasis on higher speed requires early detection of far field objects, so a multi focal active vision with gaze control is essential. For increased robustness lidar range sensors are combined with vision using an object fusion approach. An elaborate safety concept and a verification stage ensure a safe behavior of the vehicle in all situations. A communication system enables the vehicle to perform cooperative perception and action together with similar intelligent vehicles.

I. INTRODUCTION

After a few weeks of driving lessons, an adult human being is able to drive a car on its own. Most of the time, humans drive remarkably prudent, closely watching the environment and continuously tracking the behavior of other traffic participants around. However their endurance is limited: As they get tired, they get inattentive, sometimes with serious consequences.

Technical systems can compensate for this disadvantage as they expose no persistence problems. Despite several decades of research, the common vehicle is still far from being able to drive on its own. A modern driver assistance system enhances comfort and supports the driver particularly in emergency situations: For example, the electronic stability program helps the driver to maintain control when skidding. Those achievements are greatly appreciated, as they reduce accidents significantly.

Cognitive Automobiles [1] are vehicles that cannot only react in certain situations but that have enough environment knowledge to be able to act on their own. Yet, this requires a

M. Goebel, F. Rattei and G. Färber are with Technische Universität München, Institute for Real-Time Computer Systems, Arcisstr. 21, 80333 München, Germany. {goebel, rattei, faerber}@rcs.ei.tum.de

M. Thuy and F. Puente León are with Technische Universität München, Fachgebiet Verteilte Messsysteme, Theresienstraße 90/N5, 80333 München, Germany. {m.thuy, f.puente}@tum.de

M. Althoff and M. Buss are with Technische Universität München, Institute for Automatic Control Engineering(LSR), Theresienstr. 90N5, 80333 München, Germany. {althoff, mb}@tum.de

R. Nagel is with Technische Universität München, Lehrstuhl für Kommunikationsnetze, Arcisstr. 21, 80333 München, Germany. robert.nagel@tum.de

C. Wang, S. Kraus and B. Heißing are with Technische Universität München, Institute for vehicle technology, 85748 Garching, Germany. {wang, kraus, heissing}@ftm.mw.tum.de

F. Hecker, M. Russ, M. Schweitzer, H.J. Wuensche are with University of Armed Forces, Neubiberg, Department of Aerospace Engineering LRT 13, Werner-Heisenberg-Weg 39, 85577 Neubiberg, Germany. {falk.hecker, martin.russ, misw, joe.wuensche}@unibw.de

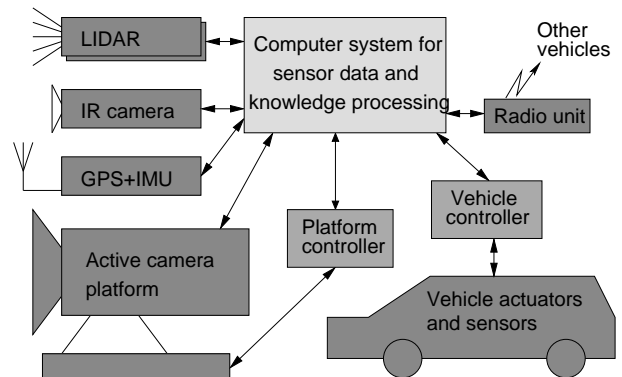


Fig. 1. Hardware architecture

major leap in system complexity and algorithmic robustness: In order to generate a safe behavior, the vehicle depends on a correct situation assessment, that in turn needs a robust perception of the vehicle's environment. Moving traffic requires all involved perception and control to be executed in real-time.

The Transregional Collaborative Research Center (TCRC 28) "Cognitive Automobiles" [2] aims to contribute significantly to the evolution of machine cognition for automotive environments. An interdisciplinary team of researchers in Munich and Karlsruhe works closely together to combine different methods, sensors and algorithms. For evaluation and tests we have vehicles in several locations. To ensure a successful cooperation we created unified interfaces, so that software modules can be exchanged between the vehicles and combined using fusion techniques for a even more powerful situation evaluation.

In January 2007, only one year after establishment, all newly acquired TCRC vehicles were able to perform visual lane keeping. In October 2007, a group of TCRC researches, among them also authors of this paper, demonstrated successful safe navigation in urban environments as "Team AnnieWAY" at the DARPA Urban Challenge [3]. This paper presents the design of the cognitive Audi Q7 in Munich, called "MUnich's Cognitive Car Innovation" (MUCCI).

II. ARCHITECTURE

A. Hardware Architecture

Fig. 1 shows the hardware architecture of the Munich cognitive automobile. On the left side it consists of a set of sensors for environment perception: An active camera platform is used for visual recognition and tracking of lanes,

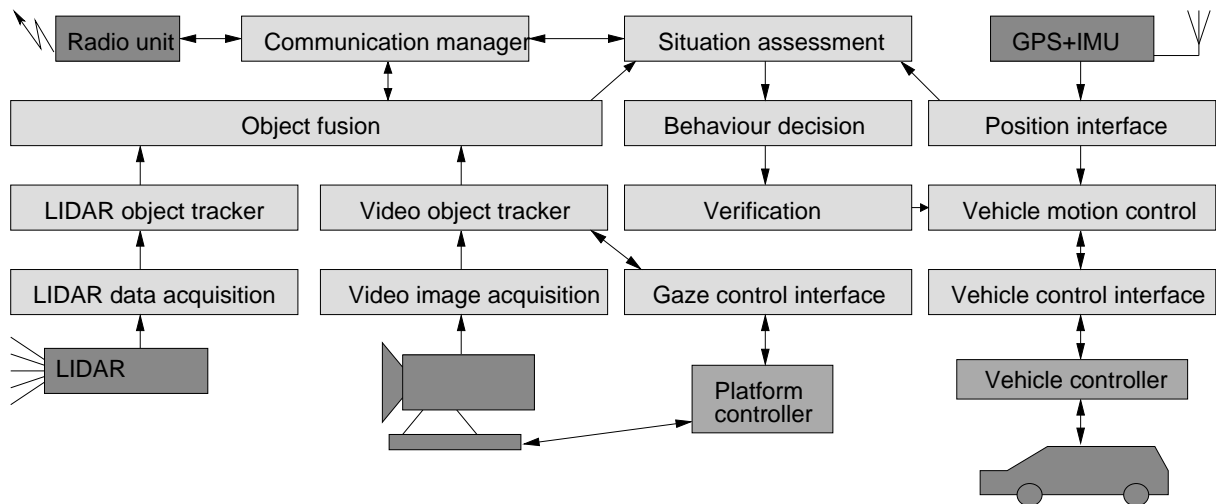


Fig. 2. Functional software architecture

vehicles and obstacles in the near range. Its two wide angle cameras, shown in [4], are also used for stereo-based depth estimation. The tele camera, shown in [5], is used for early recognition of obstacles. Two lidar sensors provide range data for obstacles and improve object detection in a data fusion stage as shown in Sec. IV. The combination of a GPS receiver with an inertial measurement unit (IMU) delivers precise position data for navigation.

All raw sensor data is fed directly into a powerful multi-core multiprocessor computer system as described in [6]. Within an Opteron system all processor cores are linked together by HyperTransport at a bandwidth of 3.2 GByte/s. Compared to an Ethernet-linked computer cluster this solution simplifies management and saves power, because computer infrastructure components (hard disks, console, power) are required only once. As every cognitive software module is executed on this platform, communication is performed with low latencies. Even large blocks of raw data can be passed between modules with very low effort, stimulating a tight cooperation between cognitive modules and their developers. Communication with other intelligent vehicles is achieved using a radio unit.

The actors and standard sensors of the vehicle itself are controlled by a dSpace AutoBox. The AutoBox serves also as a security element as shown in Sec. V. The camera platform is controlled by a dedicated platform controller. Its main task is the inertial stabilization of the tele camera for a stable fixation of far objects as outlined in Sec. III.

B. Software Architecture

Fig. 2 gives a rough overview of the software modules used in our cognitive automobile and the data flows between them. It is apparent that the data produced by one module is often used by several other ones. This includes raw data like video images that are needed for road tracking as well as object detection, and comprises also the current camera gaze direction from the gaze control interface.

For fast interprocess communication we use the real-time database for cognitive automobiles (KogMo-RTDB) [7]. It is capable of distributing raw sensor data streams to several processes and at the same time relay vehicle control commands at a rate of 1 kHz between the vehicle guidance module and the vehicle controller via CAN bus.

Every software module (light gray) shown in Fig. 2 is connected to the KogMo-RTDB. Hardware components (dark gray) are connected with dedicated interface modules that stores all raw data in the RTDB. Every piece of data is organized within the RTDB in objects that can be created, updated and destroyed. Other modules can search for certain objects within the RTDB and retrieve their data. For process synchronization a module can wait for an object to change. The efficient implementation of the RTDB manages $1.2 \cdot 10^5$ update and $2.1 \cdot 10^5$ retrieve operations per second (see [8]), depending on the object size and access method being used.

III. VISUAL ENVIRONMENT PERCEPTION

The automatic generation of suitable behavior for automobiles in traffic is mostly based on a comprehensive internal description of the actual traffic-related environment. Machine vision is used to gather information about the most relevant parts of the environment such as the road itself as well as static and dynamic objects on the road.

The image processing methods used in our vehicle emphasize on robustness and real-time capability. The scene interpretation is based on recursive estimation techniques according to the 4D approach [9], [10].

Predicted 2D feature positions based on temporarily and spatially modeled and recursively estimated objects allow for an efficient image processing. Thus, processing time is reduced and detection is more robust.

An additional feedback loop, applied directly to the first link of the image processing chain, is used to adapt the cameras exposure control algorithm depending on the expected object positions. Shifting the limited dynamic range of a camera in a high dynamic traffic scene also contributes

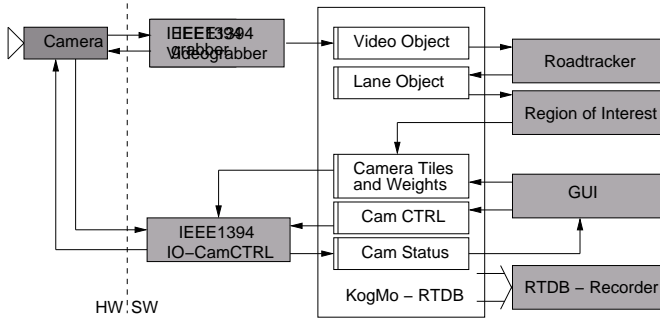


Fig. 3. Fundamental structure of the adaptive exposure control

to increased detection robustness [11]. Fig. 3 illustrates the principle structure of the feedback loop using the KogMo-RTDB and our application for road detection and tracking.

An inertially stabilized multifocal camera platform for active vision extends the limit of traditional vision based approaches for further far-field detection of traffic related objects. Miniaturized rotation rate sensors detect low frequency changes of the car’s pitch angle and mostly compensate image blur induced by vertical motion by active movement of the tele-optic camera. Precise mechanics and early feedback control loops as well as lightweight cameras are necessary to reduce latency on the one hand and to increase position accuracy on the other hand.

Two independently steerable wide-angle cameras enlarge the field of view by camera movement, extending the perception capabilities e.g. in sharp and hairpin curves. Assisted by the capability of very fast camera movements the 4D approach has proven robustness even over camera saccades neglecting the image frames during camera movement.

Within the visual perception, an important task is the modeling of an abstract description of the environment, on which an intelligent vehicle is able to act rationally. Currently this description contains three main classes: *lanes*, *static objects* and *dynamic objects*. Lanes are described by *width*, *curvature*, *derivation of the curvature*, *yaw angle* and *lateral offset* (last two with respect to the own vehicle). All lane parameters are directly estimated by predicted 2D edge features [12] with an UD-factorized, sequential, extended Kalman filter [13][14], as shown in Fig. 4. Static objects are described by 3D bounding boxes and a pose vector relative to the own position. Dynamic objects are represented in the same way but extended by a dynamic model. We currently just handle dynamic objects which conform to a *bicycle model*.

To increase robustness of the estimation of static (obstacles on the lane) and dynamic objects (other cars) we enhance the 4D approach by a *feature model*: More descriptive features than edge features are registered in an object fixed coordinate system. Tracking those registered features allows us to estimate the bounding box and pose of the object. Therefore we use a CUDA¹ implementation of U-SURF

features [15] to achieve real-time capabilities. Within the feature model we test the use of other scale space [16] and illumination invariant features, like [17], [18], [19], [20], [21]. We don’t use the rotational invariance.

IV. FUSION-BASED OBJECT DETECTION WITH LIDAR

A. Motivation and Sensor Principle

Since image processing suffers from ambient light conditions, the use of additional sensors can yield a drastic improvement of system robustness. Using active sensors like lidar scanners overcomes the limitation of the surrounding light condition using an active sensing principle. The sensor itself sends out a laser impulse at a specified angle and measures the time the light takes to come back to the sensor. Besides the very high independence on the light condition, the sensor delivers only a distance measurement if the light hits an object. Consequently, every given distance measurement is a potential part of an object. Furthermore, with the help of the distance values one can eliminate the ambiguity within camera pictures resulting from the 3D to 2D projection model.

For the given task of combining object hypothesis coming from the processing of both camera pictures and lidar processing, firstly we have to describe the chosen sensor setup. The camera rig is located at the front window of the car. Consequently, it can only perceive objects which are located ahead of the ego-car. To complement this restricted perception view, we added a lidar scanner at the rear of the car. Additionally, we also place one scanner at the front bumper. Both lidar scanners are set up parallel to the ground plane. The sensors can observe 180° at an angular resolution of 0.25°.

According to the setup of the heterogeneous sensors, object hypotheses in the rear of the car can only be established by the lidar scanner. In contrast, objects located ahead of the car can be detected – assumed that the object is located in the perception field of the camera and the lidar – simultaneously by the camera and the front lidar scanner.

B. Data Preprocessing and Object Detection

Finding objects in lidar scanner data is mainly based on segmenting regions belonging together. For instance, if the laser rays hit a car, according to the distance between the scanner and the object one will get back the contour of the car represented by a certain number of lidar scan-points. Since it is very unlikely that objects are – in the sense of the lidar scanner – that close to each other, one object will be separated the next object by a distance step in the lidar points. So, the key of finding objects in the lidar data is the search for subsequent scan-points whose distance to each other is smaller than a certain well-defined threshold [22].

For calculating the threshold just mentioned, another fact is very important: as the scanner sends out the laser rays at equally spaced angles, the local distance between two consecutive rays grows with the distance between the sensor and the object. According to that relation, the point to point

¹CUDA is an API provided by NVidia for GPU programming <http://www.nvidia.com/cuda>

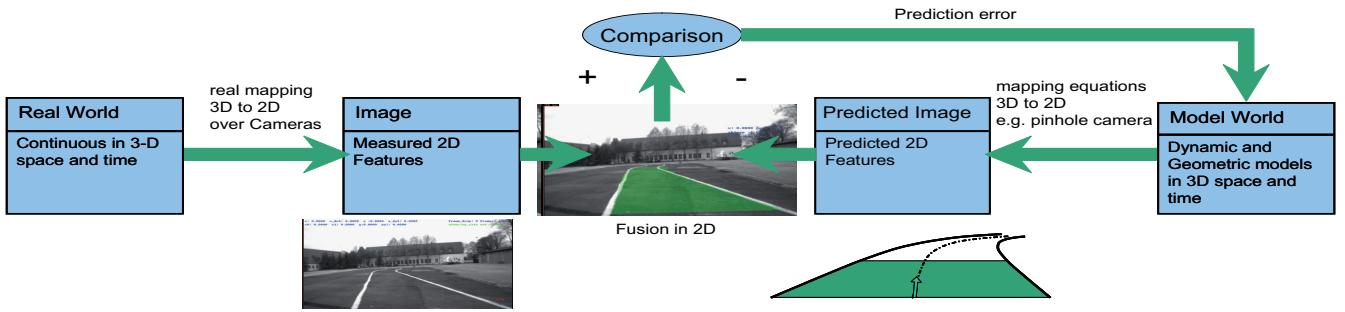


Fig. 4. Road tracking using the 4D approach

threshold is a distance dependent value, which grows with increasing distance.

After determining the objects, their dimensions and key parameters have to be extracted: the estimated object dimensions and the supposed center of gravity.

C. Kalman Filtering of Detected Objects

After segmenting the objects, an association and tracking stage has been implemented. Association means that one newly detected object has to be associated to an object from the last scan cycle. As the central association measurement, we take the center of gravity as the most important feature and determine the distance to every predicted object. If this distance is less than a certain threshold value, we have a positive object association: we have found the old object in the new scan. Otherwise, a new object hypothesis must be established.

In our contribution, the tracking is realized by a linear Kalman filter based on an uniform movement model [23]. The change of speed is modelled by a Gaussian noise source. Therefore, the filter is capable of predicting the object speed in addition to the object position. For the case of a positive object association, the corresponding Kalman filter has to be updated. For objects without any possible association, a new Kalman filter has to be set up.

D. Fusion of Object Hypothesis

As already mentioned, the object hypotheses coming from the lidar sensors have to be combined with the object hypotheses from the image processing stage. To this end, we use a maximum likelihood estimator for the object positions. The image processing stage delivers the specific variance for their object positions. The variance related to the lidar hypotheses can be inferred out of the Kalman filter. As we know all the key parameters of the assumed Gaussian likelihood functions, it is possible to multiply them and search for its maximum as the optimization criterion for the estimated position.

V. VEHICLE CONTROL IN COMBINATION WITH SAFETY CONCEPT

A. Motivation

Autonomous driving including accelerating, braking, steering and other advanced maneuvers, such as lane change

or platooning, should be exactly executed in the cognitive vehicle. To realize these objectives, besides re-fitting the adequate actors in the vehicle, path planning based on the object recognition and maneuver generation with respect to the vehicle's capabilities is necessary.

Because this control process completely takes the vehicle's motion in hand and can cause fatal damage to the vehicle and people in case of an erroneous implementation, it should be accomplished precisely, in bounded delays and without errors by means of variable control algorithms. Furthermore it is also required to implement a safety concept to detect the emergency situation and in this case bring the vehicle back to a safe status.

B. Experimental Vehicle

An Audi Q7 (Diesel 3.0l; YOC: 2006) with automatic transmission was employed as experimental vehicle. The e-gas module is integrated between the gas pedal and its control unit. The steering function is realized by a brushless Megaflex motor, which is directly mounted on the steering rod. To apply the brakes, two solutions are implemented because of the safety demand on the redundancy. The first uses the normal brake booster, which allows directly building the braking pressure in the master cylinder. The second is a pneumatic actor to push the brake pedal, which is powered by the off-the-shelf air suspension. A powerful dSpace AutoBox takes the overall motion control of the vehicle. In the parallel mode the PXI Box from National Instruments works as the watching and safety machine. In addition, the vehicle is equipped with the monitoring and emergency stop as Fig. 5 shows.

C. Vehicle Motion Control

The knowledge about one's own performance is a prerequisite for independent decision making and action selection. To execute an action that is represented by a corresponding capability, you need to know which conditions for its execution must apply and whether they are satisfied. Complex maneuvers are composed from different time-coordinated driving action capabilities, which depend on lower-skill driving basics; the associated performance can only be provided if all the necessary conditions are met. The representation of vehicle capabilities can be found in so-called capability-networks [24]. The concept represents an approach to abstract essential skills to a general extent. An ability node

consists of an organizational, operational and monitoring component to perform the encapsulated functionality. The modeling of this behavior is carried out in state machines; within the operational component execution models of the required function are deposited. The controller topology is not fixed; control parameters can be adapted to the respective conditions on runtime, complete sets of parameters can be changed or different controller structures can be used. Apart from the purely static representation of the availability of skills, there is a dynamic component. Each capability node takes the current availability of the required components into account. The dynamic availability depends on the status of necessary components, which will be given from part of the safety concept. Due to the fact that accumulated knowledge is available on the encapsulated functionality in capabilities in form of execution models, availability modes, progress dimensions, etc. across the system via the capability network, this knowledge can also be used to predict the ability of an expected performance. Prediction is particularly useful for decision making and planning of autonomous systems.

D. Decision Making and Path Planning

The object recognition modules like image processing or the communication between cooperative vehicles are generating the scene tree in the KogMo-RTDB. Based on the object characteristics, their positions, and their predicted trajectories an ideal path can be calculated.

A set of rudimentary and adjustable commands is pre-calculated and updated continuously by the vehicle control interface as shown in Fig. 2. This assures that the planned path is drivable within the vehicle's capabilities as described in paragraph V-C. The stopping of the vehicle at a defined braking distance and the adjustment of the current acceleration are the skills in longitudinal direction. The ability to follow a path and to change the lateral offset of the vehicle to the lane according to a clothoid model are the skills in lateral direction. Currently the lateral maneuvers and the longitudinal maneuvers are successively executed.

This set of maneuvers, the estimated objects and their corresponding trajectories are the input variables of a finite state machine. We are using fuzzy logic as proposed in [25] to reduce the amount of possible states. The maneuver progress is continuously observed and compared to the desired trajectory. So the emerge of new obstacles, time lags or deviations can lead to a reconfiguration or even to a termination of the current maneuver.

E. Safety Concept

The safety concept as shown in Fig. 5 focuses on the online safety warranty of the cognition units. Cognition units include all the hardware and some software programs which are related to the vehicle motion control. The basic principle is that the driver is capable of taking back the control of the vehicle at any time in case an error situation is detected. Firstly the PXI BOX acts as the watching machine, integrating all the information from sensors, actors, motion controller

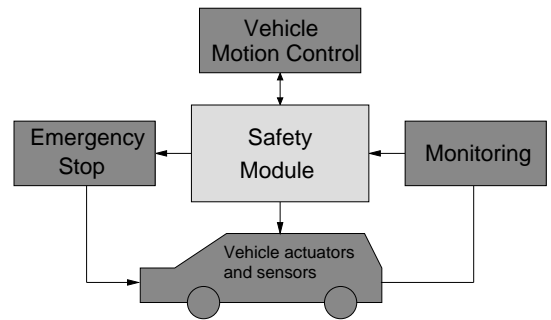


Fig. 5. Safety concept

and vehicle via CAN bus. There the collected signals will be analyzed in the watchdog program, which makes the check of the actor status and signal plausibility. Subsequently the status of the components will be sent back to the motion controller for the dynamic availability determination. If an error is detected in the actor or software program from the mode-based diagnose, it would call the warning or emending method depending on the error classification. Meanwhile it should indicate its location and deliver the necessary advice to the driver. If the error exists in a certain time threshold, an emergency program will automatically bring the vehicle into the fail-safe mode with the redundant hardware/software. The redundant system is already implemented in the cognitive vehicle design and the knowledge base of the diagnose system will be established via the SIL, HIL simulation method.

VI. SAFETY ASSESSMENT BY VERIFICATION

The verification module decides if planned trajectories are safe using methods of hybrid verification. Safety assessment is performed based on the regions of positions, that traffic participants surrounding the autonomous car can reach within future time intervals. These possible positions are also referred to as the reachable sets of the traffic participants. Reachable sets are computed based on the initial state (position, velocity) and an uncertain dynamic model of the traffic participants. Uncertain models are regarded as dynamic models, where inputs and parameters are uncertain within certain sets. This allows to compute reachable sets for classes of traffic participants, such as cars, trucks or bicycles.

Given the reachable sets of traffic participants, one can state the following: if the reachable sets of the autonomous car following its planned trajectory, do not intersect with any reachable set of other traffic participant (for all time intervals within a certain prediction horizon), the trajectory of the autonomous car is guaranteed to be safe. In addition, the probability distributions of possible positions of traffic participants are computed within their reachable sets, in order to compute a probability of safety for the planned trajectory. An exemplary situation that is verified using probabilistic reachable sets is presented in Fig. 6(a). This situation is not safe, as the probabilistic reachable sets for the second time interval of both vehicles intersect. The computation

of probabilistic reachable sets of traffic participants is performed online in the autonomous car. As mentioned above, the prediction is limited to a time horizon t_f . After a certain time increment $\Delta t = t_{i+1} - t_i$, the computation of reachable sets is reset regarding the updated measurements of the traffic situation. The procedure is illustrated in Fig. 6(b).

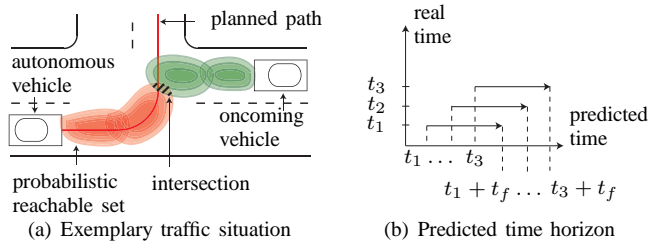


Fig. 6. Illustration of reachable sets and the resulting prediction horizon

A. Computing Probabilistic Reachable Sets of Traffic Participants

The motion of traffic participants is modeled in two stages. First, the paths that a vehicle can follow are generated. They do naturally arise from the possible routes that the vehicle can take in the examined road network segment. Besides the possible paths of a vehicle, the deviation from these paths is modeled by a static piecewise constant probability distribution that varies between road user types. In a second step, the longitudinal dynamics of the vehicles along the paths is considered. The dynamics is modeled by a hybrid automaton [26] which combines discrete and continuous dynamics. Within each discrete mode, a continuous dynamics model is defined, capturing all possible behaviors within the discrete mode, e.g. in mode *acceleration*, all behaviors ranging from no to full acceleration are considered. A more detailed description on the modeling of traffic participants concerning path generation and path following is given in [27].

In order to be able to efficiently compute probabilistic reachable sets of the hybrid dynamics of traffic participants, their mixed discrete/continuous dynamics is abstracted to Markov chains. A detailed description of the abstraction to Markov chains and the computation of probabilistic reachable sets using Markov chains is given in [28], [27]. Besides computing probabilistic reachable sets for each traffic participant individually, a more sophisticated computation of reachable sets encountering the interaction of traffic participants has been proposed [29].

B. Numerical Example

The numerical example considering interaction between traffic participants shows an intersection scenario as depicted in Fig. 7(a) illustrating the initial setting. Cars *A* and *B* approach the crossing from a street that has no right of way and cars *C* and *D* have right of way. In Fig. 7(b)-7(d), one can see the probabilistic reachable sets for selected time intervals (the darkness of the color indicates the probability). It is clearly recognizable that cars *A* and *B* decrease their velocity when they approach the crossing and that they wait

until the crossing cars have passed. Note that the reachable sets of cars overlap to a certain degree, such that individual cars cannot be distinguished after a certain prediction time.

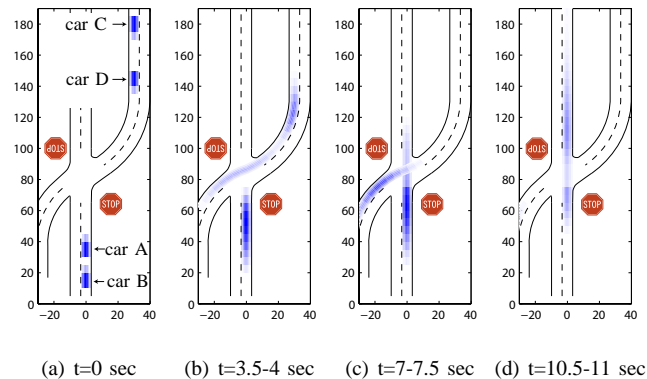


Fig. 7. Reachable sets of the intersection scenario

VII. COMMUNICATION FOR INTER-VEHICLE COOPERATION

A. Motivation

For the realization of cognitive capabilities, the vehicles need not only to be able to perceive and understand their environment and act appropriately, they also have to be able to communicate with each other to support functions such as cooperative perception, distributed object fusion and coordinated behavior decision. The key requirements in this context are bandwidth and real-time capability. Furthermore, the communication design has to be decentralized, self-organizing and fault-tolerant.

B. Implementation

On the network layer, the data streams from and to the data object communication module are multiplexed with maintenance messages that handle network topology detection, uni- and multicast routing as well as security signaling [30]. A loopback interface allows the connection of external security modules, such as a secure socket wrapper. All of these functions are implemented transparently so that no modification for controlling these modules are necessary to any application.

The data object communication module itself integrates seamlessly with other applications, the central interface being the real-time database. Through a simple API, other applications notify the communication module about data objects that should be sent (*exported*) to other vehicles. Received (*imported*) data objects are automatically stored in the real-time database, organized hierarchically under an informational object that contains data about the sending vehicle. Also, information about the networks radio topology is available to other applications, aiding, for instance, setting up cooperative groups based on the connectivity of vehicles.

Applications that depend on data objects from other vehicles can simply search the database for the presence of an object of the specific data type and will in return get a list of

relevant objects. From the hierarchical organization in parent and child objects, the application can deduce the origin of a data object, if necessary. Whenever a data object is updated in the senders database, it is exported to other vehicles, imported by those vehicles that receive it and updated in the receiving vehicles' databases. Applications waiting for an update of that data object can then retrieve the object from the database and resume processing.

VIII. CONCLUSIONS AND OUTLOOK

In this work the design of the Cognitive Automobile in Munich is presented. It features all characteristic components with emphasis on robust situation assessment and safe behavior. The communication link between two vehicles is tested, cooperative behavior has been shown in simulation [31]. One of our next steps is to demonstrate cooperative perception and maneuvers in real-world.

To increase the speed and robustness of the visual perception we focus first on a scene-dependent best parameter estimation on the physical sensor level and second, on improving the prediction step in [9] by a visual enhanced ego-motion estimation and by focusing on keypoint descriptors solving the correspondence problem more reliable.

IX. ACKNOWLEDGMENTS

The authors gratefully acknowledge support of this work by the Deutsche Forschungsgemeinschaft (German Research Foundation) within the Transregional Collaborative Research Center 28 "Cognitive Automobiles". We also thank our colleagues for valuable discussions.

REFERENCES

- [1] C. Stiller, G. Färber, and S. Kammel, "Cooperative Cognitive Automobiles," in *Proc. IEEE Intelligent Vehicles Symposium*, 2007.
- [2] Transregional Collaborative Research Centre 28. [Online]. Available: <http://www.kognimobil.org>
- [3] DARPA Urban Challenge. [Online]. Available: <http://www.darpa.mil/grandchallenge/>
- [4] T. Dang, C. Hoffmann, and C. Stiller, "Self-calibration for Active Automotive Stereo Vision," in *Proc. IEEE Intelligent Vehicles Symposium*, 2006, pp. 364–369.
- [5] W. Günthner, P. Wagner, and H. Ulbrich, "An inertially stabilised vehicle camera system - hardware, algorithms, test drives," in *IEEE Industrial Electronics, IECON 2006*, 2006.
- [6] M. Goebel, S. Drössler, and G. Färber, "Systemplattform für videobasierte Fahrerassistenzsysteme," in *Autonome Mobile Systeme 2005*. Springer-Verlag, 2006, pp. 187–193.
- [7] M. Goebel and G. Färber, "A Real-Time-capable Hard- and Software Architecture for Joint Image and Knowledge Processing in Cognitive Automobiles," in *Proc. IEEE Intelligent Vehicles Symposium*, 2007, pp. 734–740.
- [8] —, "Eine realzeitfähige Softwarearchitektur für kognitive Automobile," in *Autonome Mobile Systeme 2007*. Springer-Verlag, 2007, pp. 198–204.
- [9] E. D. Dickmanns, "4D-Szenenanalyse mit integralen raumzeitlichen Modellen," in *9. DAGM-Symposium Mustererkennung*, ser. Informatik Fachberichte Nummer 149. Springer-Verlag, 1987.
- [10] —, *Dynamic Vision for Perception and Control of Motion*. London: Springer Verlag, 2007.
- [11] F. Rattei, M. Goebel, and G. Färber, "Beitrag zur Robustheitssteigerung videobasierter Fahrerassistenzsysteme durch frühe Rückkopplungen zur Sensorebene," in *Bildverarbeitung in der Mess- und Automatisierungstechnik*. VDI-Berichte, VDI Verlag, Düsseldorf, 2007.
- [12] E. D. Dickmanns, "Efficient Computing of Intensity Profiles for Realtime Vision," *Lecture Notes in Computer Science*, pp. 131–139, 1998.
- [13] G. J. Bierman, "Measurement updating using the u-d factorization," in *Proceedings of IEEE conference on control and decision*, 1975, pp. 337–346.
- [14] D. Simon, *Optimal state estimation*. Wiley-Interscience, 2006.
- [15] H. Bay and L. V. Gool, "Surf: Speeded up robust features," *ECCV*, 2006. [Online]. Available: <http://www.vision.ee.ethz.ch/~surf/>
- [16] T. Lindeberg, *Scale Space Theory in Computer Vision*. Kluwer Academic Publishers, 1994.
- [17] C. Schmid and R. Mohr, "Local grayvalue invariants for image retrieval," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, no. 5, pp. 530–535, may 1997.
- [18] D. G. Lowe, "Object recognition from local scale-invariant features," in *Proc. of the International Conference on Computer Vision ICCV, Corfu*, 1999, pp. 1150–1157.
- [19] A. Baumberg, "Reliable feature matching across widely separated views," 2000, pp. 774–781.
- [20] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in *Proceedings IEEE Conf. on Computer Vision and Pattern Recognition*, 2001.
- [21] K. Mikolajczyk and C. Schmid, "A performance evaluation of local descriptors," in *In Proc. IEEE Conf. Comp. Vision Patt. Recog., volume 2, pages 257–263, 2003*, 2003.
- [22] A. Mendes, L. Conde Bento, and U. Nunes, "Multi-target Detection and Tracking with a Laserscanner," in *Proc. IEEE Intelligent Vehicles Symposium*, 2004, pp. 796–801.
- [23] D. Steller and K. Dietmayer, "Multiple Hypothesis Classification with Laser Range Finders," in *Proceedings of the IEEE Intelligent Transportation Systems Conference*, 2004, pp. 195–200.
- [24] K.-H. Siedersberger, "Komponenten zur automatischen Fahrzeugführung in sehenden (semi-)autonomen Fahrzeugen," Ph.D. dissertation, Fakultät für Luft- und Raumfahrttechnik, Universität der Bundeswehr München, 2003.
- [25] M. Pellkofer, "Verhaltensentscheidung für autonome Fahrzeuge mit Blickrichtungssteuerung," Ph.D. dissertation, Fakultät für Luft- und Raumfahrttechnik, Universität der Bundeswehr München, 2003.
- [26] T. Henzinger, *The theory of hybrid automata*, ser. NATO ASI Series F: Computer and Systems Sciences. Springer, 2000, vol. 170, pp. 265–292.
- [27] M. Althoff, O. Stursberg, and M. Buss, "Online verification of cognitive car decisions," in *Proc. of the 2007 IEEE Intelligent Vehicles Symposium*, 2007.
- [28] —, "Safety assessment of autonomous cars using verification techniques," in *Proc. of the American Control Conference*, 2007.
- [29] —, "Stochastic reachable sets of interacting traffic participants," submitted to the IEEE Intelligent Vehicles Symposium 2008.
- [30] R. Nagel, S. Eichler, and J. Eberspächer, "Intelligent Wireless Communication for Future Autonomous and Cognitive Automobiles," in *Proceedings of the IEEE Intelligent Vehicles Symposium*, 2007.
- [31] S. Vacek, R. Nagel, T. Batz, F. Moosmann, and R. Dillmann, "An Integrated Simulation Framework for Cognitive Automobiles," in *Proc. IEEE Intelligent Vehicles Symposium*, 2007, pp. 221 – 226.