Team AnnieWAY's entry to the Grand Cooperative Driving Challenge 2011

Andreas Geiger, Martin Lauer, Frank Moosmann, Benjamin Ranft,
Holger Rapp, Christoph Stiller, and Julius Ziegler

Abstract—In this paper we present the concepts and methods developed for the autonomous vehicle AnnieWAY, our winning entry to the Grand Cooperative Driving Challenge of 2011. We describe algorithms for sensor fusion, vehicle-to-vehicle communication and cooperative control. Furthermore, we analyze the performance of the proposed methods and compare them to those of competing teams. We close with our results from the competition and lessons learned.

Index Terms—autonomous vehicles, cooperative driving, V2X-communication

I. RESEARCH BACKGROUND AND TEAM COMPOSITION

In the following we give a brief review of the history of cooperative driving and introduce our team *AnnieWAY*, with which we entered the Grand Cooperative Driving Challenge.

A. Cooperative Driving and the GCDC

Driver assistance systems already help to make vehicle navigation safer and more comfortable. Nevertheless, one of the main challenges remains unsolved: An increasing amount of traffic on the streets causes congestion and environmental pollution. Traffic jams result from inhomogeneities in traffic flow, and consequently, longitudinal vehicle control plays an important role in avoiding them. However, human factors such as reaction time and perception constraints limit the possibilities to improve traffic homogeneity.

The technical basis for autonomous longitudinal control like electronic brake and throttle has been laid by the emergence of adaptive cruise control (ACC) systems [1], which employ radar for measuring distance and speed of a leading vehicle. However, standard ACC systems only control the vehicle's speed depending on distance and velocity of the vehicle directly ahead, neglecting the overall traffic situation. While these systems undoubtedly improve driving comfort, their influence on traffic homogeneity is still disputed [1], [2]. One idea to resolve these shortcomings and improve traffic homogeneity is to use vehicle-to-vehicle communication to provide the vehicle with information about the current traffic situation. If multiple vehicles ahead can be accounted for, more elaborated control approaches can be employed.

Most approaches for cooperative driving are based on the assumption of identical technical equipment and use of the

Fig. 1. One heat of the GCDC with six competing vehicles. AnnieWAY from Karlsruhe Institute of Technology is the silver vehicle directly in front of the truck.

same control strategy for all vehicles in a platoon. This assumption can't be made in the real world: different vendors will use different technical solutions. Older vehicles might use techniques different from those employed in newer ones. Furthermore, passenger cars, vans, trucks, and buses will be mixed on the same lane, and autonomous vehicles will share roads with manually driven cars.

The *Grand Cooperative Driving Challenge 2011* (GCDC) [3] was the first competition to implement such a realistic, heterogeneous scenario. It was organized by the *Netherlands Organisation for Applied Scientific Research* (TNO) in Helmond. Participating teams had to come up with strategies that were able to perform as good as possible without knowing the algorithms and technical equipment of other vehicles in the platoon. Control strategies had to cope with unexpected behavior of other vehicles, varying data quality, and sudden failure of communication, among others. Fig. 1 shows one heat of the GCDC, illustrating the large variety of vehicles and technical solutions in the competition.

B. State-of-the-art in Cooperative Driving

Cooperation among traffic participants plays an important role in everyday life to ensure traffic safety and traffic flow [4]. E.g., resigning one's right of way at a crossroads or allowing other vehicles to merge on one's own lane are frequent behaviors which are most beneficial to all traffic

participants and which may even resolve critical situations. While human drivers are still superior to automated vehicles in many situations, machines are able to negotiate cooperative driving maneuvers significantly faster and with fewer misunderstandings than humans.

Progress from intensive international research on automated cooperative driving has been demonstrated by numerous demonstrations. In August 1997, *Demo '97* took place in San Diego, USA, showing impressive results from the US National Automated Highway System Consortium (NAHSC) on self-driven vehicles. A platoon control demonstration showed cooperative platoon driving of up to eight identical vehicles on the instrumented freeway I-15 that was closed to the public. The vehicles were driving automatically at 6.5 meters spacing and at 60 mph (97 km/h). The key technologies were distance keeping using radar, lidar, video and intervehicle communications as well as lane following via roadway embedded magnets, roadway laid radar-reflective stripes, or existing visible lane markers detected with vehicle mounted cameras [5], [6]. In *Demo 2000* the National Institute of Advanced Industrial Science and Technology presented cooperative platoon driving of five vehicles on a test track in Japan, which included more advanced maneuvers such as stop-and-go, merging, and obstacle avoidance [7]. In May 2003 a platoon of three heavy trucks was presented by the European *CHAUFFEUR* project capable of cooperative cruise-control, lane keeping and concerted lane change and active obstacle avoidance maneuvers [8]. The German Karlsruhe-Munich Collaborative research center *Cognitive Automobiles* (2006-2010) has developed methods for ad-hoc group formation and joint overtaking and emergency maneuvers of automated vehicles [9], [10]. Small autonomous and cooperative passenger vehicles were presented by the European *Cybercar 2* consortium in September 2008 in France. The vehicles were designed for low speed autonomous cooperative city transportation capable of automated coordinated driving and cooperative intersection traversal [11]. Recently, in May 2011 the European *INTERSAFE 2* consortium demonstrated left turn warning, inhibition of acceleration, and automated braking in case of an imminent collision with oncoming traffic on an intersection closed to the public in Germany. The vehicles were equipped with laser sensors, cameras, DGPS, a map of the intersection, and a V2X communication system. Additional laser sensors, cameras and communication devices were mounted at the infrastructure [12], [13].

C. Team AnnieWAY

Team AnnieWAY is a group of researchers hosted at Karlsruhe Institute of Technology. Its overall goal is to develop and integrate new techniques for autonomous driving and to compare these techniques (e.g. on benchmarks and competitions) to other approaches. Based on the experiences made during the DARPA Grand Challenge 2005 [14] in a mixed team with Ohio State University, team AnnieWAY was formed to participate at the DARPA Urban Challenge 2007 [15] with its own vehicle, called *AnnieWAY*.

The research focus of team AnnieWAY is mainly in mobile perception and scene understanding based on video and lidar
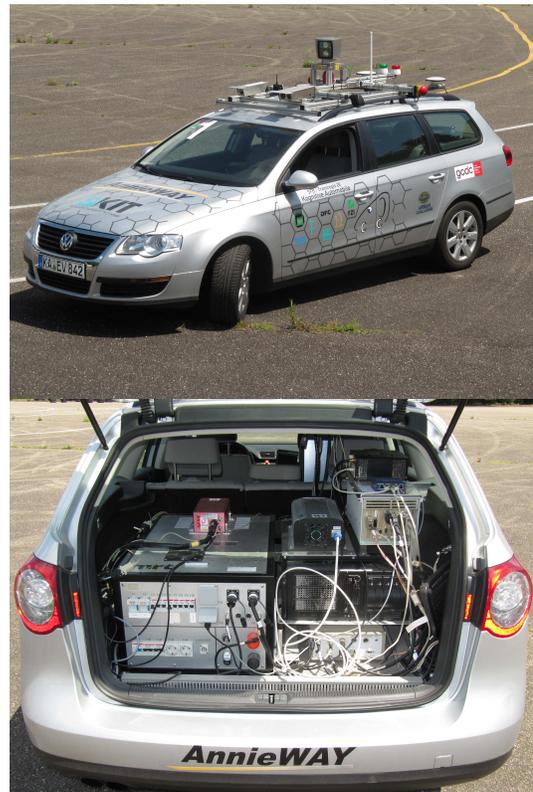
Fig. 2. The experimental vehicle (top) with its technical equipment in the trunk (bottom). The bottom image shows the power supply on the left, the GPS/INS unit on top (red box), and the control computers on the right.

sensors. This includes sensory processing techniques like real time stereo matching [16], [17], 3D scene reconstruction [18], and map generation from stereo image sequences [19] as well as scene segmentation [20] and scene understanding [21], [22]. In lidar data interpretation for autonomous vehicles the team works on efficient segmentation [23], object tracking [24], and map generation techniques [25].

In order to be able to run a vehicle fully autonomously, the team also develops methods for path and trajectory planning. This includes efficient collision checking [26], trajectory generation based on fast lattice search [27] as well as control strategies for path- and trajectory following [28], [29].

The remainder of this paper is organized as follows. In Sec. II we describe our experimental vehicle as well as the general software and hardware architecture of our system. In the subsequent sections, we discuss individual components like the communication modules (Sec. III), the environment representation (Sec. IV), and the control strategy (Sec. V). The final section wraps up our results and highlights lessons we have learnt during our participation in the GCDC.

II. EXPERIMENTAL VEHICLE AND SYSTEM ARCHITECTURE

Our experimental vehicle AnnieWAY (Fig. 2) is equipped with several modifications over the VW Passat base vehicle: Electronically controllable actuators for acceleration, brakes, transmission and steering have been added, each of which

can be enabled individually. A CAN gateway allows sending requests to these actuators and receiving selected signals like wheel speeds and status information. It additionally implements low-level safety components such as disengagement of autonomous functions in case the driver needs to interfere.

Fig. 3 sketches the technical components and data flow of our system employed at the GCDC. Gray boxes symbolize hardware devices, while white boxes illustrate software components. In the following, we briefly explain the purpose of the individual components within the categories sensors, computers and software.

a) GPS/INS: Self localization of the ego-vehicle is implemented by a combined inertial- and satellite-based navigation system[1], which can optionally be augmented by terrestrial reference stations. Using real time kinematics (RTK) correction, it provides precise position, velocity and acceleration of the host vehicle.

b) 24 GHz Doppler radar: Communication-based information on other vehicles is supplemented by the radar as part of the vehicle's standard ACC system. We decided to use it for robustness reasons, in case that transmitted positions of other vehicles become unreliable. The radar component is connected to the system via the vehicle's CAN bus.

In order to process sensor data, to plan and to control AnnieWAY's trajectory and to communicate with other vehicles, a total of three computers are installed in the vehicle's trunk:

c) AnnieWAY host computer: A Linux based server computer performs most higher level control- and data processing tasks. It is equipped with two six-core CPUs. A real-time database [30] serves as a virtual bus system for inter-process communication. It enables both synchronous and asynchronous queries as well as recording and replaying of data streams.

d) Real-time computer: The connection to the prototype vehicle itself is made through a modular rapid prototyping system[2], which can meet hard real-time requirements for critical tasks such as actuator control, driver intervention handling, fail-safe functionality and feedback trajectory stabilization. Especially the latter is important for the GCDC as it implements the low level acceleration controller described in Sec. V.

e) CALM-gateway: A separate x86-based Mini-ITX PC has been added for 802.11p-based communication. It runs the CALM daemon and dispatches incoming and outgoing data packages.

The architecture is completed by a set of software modules, each providing a building block to the actual GCDC system:

f) Vehicle Manager: This component receives vehicle information broadcast from other platoon members and augments it with radar data. The main purpose of the Vehicle Manager is to abstract from the latency of the received data: Through extrapolation and filtering it can provide an estimate of the platoon state at any given point in time.

g) Map Matcher: The Map Matcher is responsible for assigning vehicles to lanes, and hence decides which of them

are in the same platoon as the host vehicle. Furthermore, it computes geodesic distances between vehicles, which serve as inputs to the high level controller module.

h) pose server: This process interfaces to the GPS/INS hardware via UDP/IP.

i) radar server: This process interfaces to the radar sensor via CAN bus.

j) Low Level (LL) Controller: The low level controller receives a reference acceleration from the main computer, and stabilizes it based on readings from the GPS/INS. It is connected to all actuators through a CAN gateway.

k) High Level (HL) Controller: Based on the platoon state, the high level controller determines an optimal acceleration of the host vehicle to be passed downstream to the low level controller.

III. INTER-VEHICLE COMMUNICATION

The inter vehicle communication consists of a hardware and a software layer which will be discussed in the following sections.

A. Hardware & Drivers

The communication hardware is based upon the 802.11p standard, an amendment to the popular wireless LAN standard 802.11 [31] that is widely used in consumer devices. Besides defining the transmission frequencies, gains and ranges, the standard also specifies the basic addressing of devices using the MAC layer that is also used for wired Ethernet networks. The 802.11p standard broadcasts in the ITS band of 5.85-5.925 GHz and was specially derived for car-to-car communication. A good overview over the standard is given in [32].

Team AnnieWAY uses two different hardware modules in the form of mini-PCI plug-in boards. For the contest we settled on a Mikrotik RH52 card and an ECP12-5800 antenna. While testing, we also evaluated the UNEX DCMA-86P2 card together with an DM-5500S dome antenna. Equipping two cars with the combination of Mikrotik/ECP12-5800 allows for a stable communication up to roughly 800 m if an unimpeded line of sight is maintained. The roundtrip (ping) times are between 1 and 50 ms depending on surroundings, weather and distance. The UNEX/DM5500S combination only allows for 250-300 m communication range under the same conditions. The ping times are comparable.

The chips on the wireless LAN cards are very similar to comparable 802.11a cards. Still, kernel drivers had to be adapted in order to access all the required features. We based our implementation on current Atheros 5k drivers from the Linux kernel (ath5k) and on patches from older Atheros drivers provided by TNO, the organizers of the GCDC 2011.

B. Software

1) Protocols: The GCDC is not using the IP protocol for communicating, instead the ISO Communications Access for Land Mobiles (CALM) protocol [33] was chosen. It uses MAC multicasting- or broadcasting packages, and only offers a limited addressing scheme for peer-to-peer communication. It also

---

[1]OXTS RT 3003

[2]dSPACE AutoBox with DS1005 PPC Board
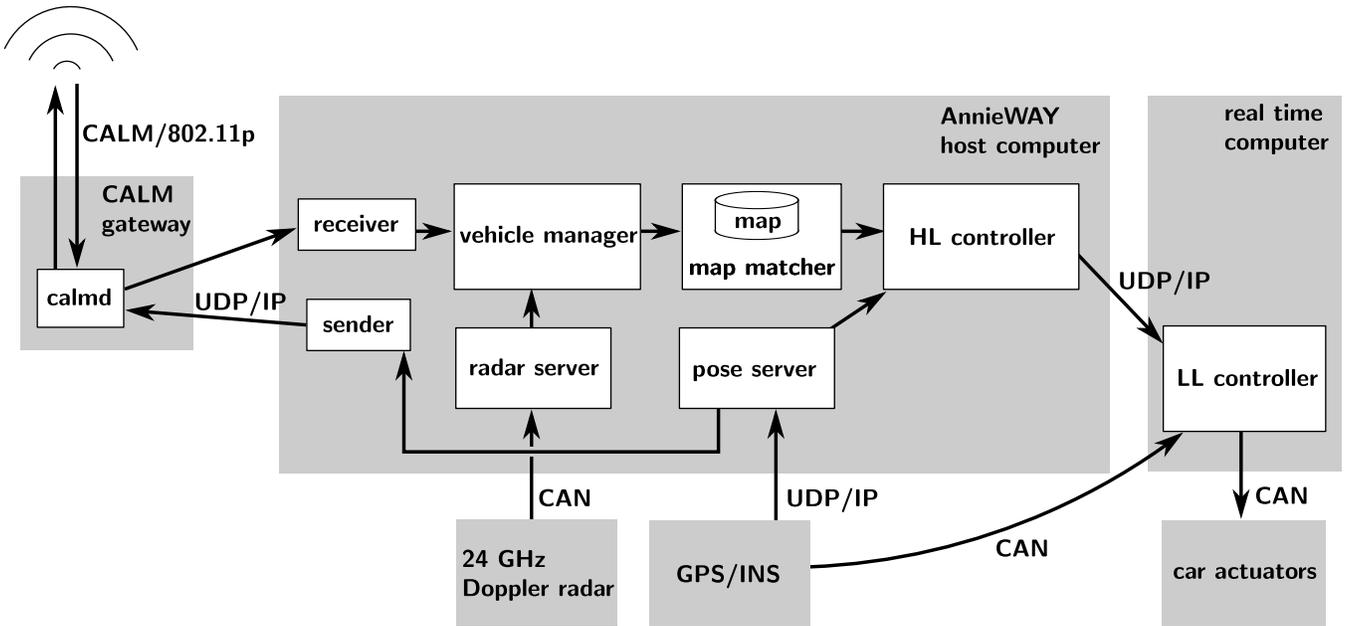
Fig. 3. GCDC system architecture. See Sec. II for explanation.

does not implement routing ideas, instead relying on important messages being passed on by higher level protocols. CALM is not natively supported by Linux, but can be implemented in user space using RAW sockets.

The CALM protocol is very complex and offers a rich feature set and therefore high implementation costs. For the GCDC, a small wrapper program called *calmd* was provided by TNO that essentially translates from incoming broadcast CALM messages to a TCP connection and vice-versa. We based our own calmd implementation on this version, but significantly improved upon the feature set and stability. We also added 64 bit compatibility. In our setup, the calmd is running on the ITX CALM gateway computer. Another process on the same computer gathers the packets from the calm daemon via TCP and communicates their content via UDP over a wired connection with the AnnieWAY host computer.

2) Receiver & Sender: Both, the receiver and sender processes are running on the AnnieWAY host computer. The receiver is handling all packets that are passed over from the CALM gateway via TCP/IP, tries to unpack the GCDC payload inside the packages and writes the data into the real time database. If no GCDC payload is found, the packet is discarded. As the CALM protocol does not offer error correction or checksumming, the receiver also implements a number of heuristics that reduce the risk of corrupt packages reaching the database.

The sender observes the database for changes and encodes corresponding GCDC packets which are then broadcast. As all sources on the host computer are trusted, this software is significantly less conservative in its error checking compared to the receiver.

3) Auxiliary Software: The combination of the new 802.11p standard, the new CALM protocol and the new cards with custom drivers proved to be unstable at first. As the communication is crucial for the GCDC, a lot of effort was put

into making it as stable as possible. During the implementation, bug tracking and network hardening steps, a number of tools proved useful. Their benefit and design intentions are discussed in the next paragraphs.

a) CALM Sender and Receiver: A pair of scripts have been implemented that check the number of lost or corrupted packages sent via the CALM protocol over wireless LAN. This information is vital to estimate the probability of receiving wrong information. As the CALM protocol does no error detection or correction, data that was received partly scrambled is directly passed on. These scripts also proved useful to detect buffer over- and under-runs in the kernel driver and the user-land libraries. Our system was hardened versus partly scrambled packages by adding feasibility checks of the data: we only accepted packages that were send with a timestamp of today and GPS position in our vicinity.

b) CALM Roundtrip Sender & Receiver: The roundtrip receiver is an echo server for the CALM protocol: it immediately rebroadcasts everything it receives. The roundtrip sender is sending packages with a fixed content and a defined delay between packages. It also listens for the echo replies and measures the roundtrip time for each package. Usually, networks are designed to value bandwidth over latency, e.g. by collecting many small send requests and combining them into one Ethernet frame. During the GCDC, however, low latency is more important than bandwidth. These scripts helped profiling and optimizing the roundtrip time. Overall, we achieved to reduce it by one order of magnitude and reached data roundtrip times which are comparable to ping times.

c) CALM Fuzzer: To maximize stability and security of network applications, all data received from the outside must be considered unsafe, potentially broken, or even maliciously crafted to exploit vulnerability in the receiving system. To stress test our communication framework, a CALM network fuzzer was written. It floods the network with either com-

(a) Kd-tree of map   (b) Geodesic vs. Euclidean distance
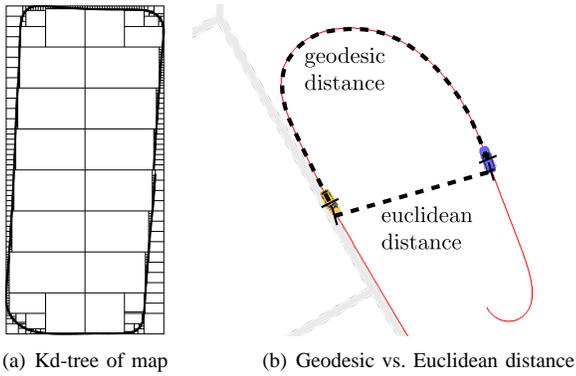
Fig. 4. Our maps are stored in kd-trees, as shown in (a) for one of our testing grounds. This allows for fast map matching and evaluation of geodesic distances between vehicles on the same lane (b).

pletely random data or slightly mutated GCDC payload packages with a very high frequency. This tool helped us in testing and improving the stability of the network stack tremendously. For example, it revealed a number of critical bugs like crashes and infinite loops in the CALM software stack.

IV. ENVIRONMENT REPRESENTATION

The GCDC took place on a normal highway with additional infrastructure, namely traffic lights and speed limits. Hence, it was sufficient to model the environment as a flat 2D world. The model was split into a static and dynamic part. The static environment corresponds to the road with its lanes. The dynamic environment comprises all vehicles, the traffic lights, and the speed limits (which might change over time).

A. Maps and Matching

The GCDC competition involves platooning in scenarios with multiple adjacent lanes. In order to join a platoon of other vehicles, the correct assignment of vehicles to lanes is important. Furthermore, the controller needs to be precisely informed about the distance to other cars on the vehicle's own lane. We handled this by recording a map of the road a priori.

To this end, we recorded the vehicle's GPS coordinates while driving on the right lane. In order to cope with metric distances, our map implicitly defines a local Mercator coordinate system with its origin anchored at the first point of the GPS track. Our map-creation algorithm subdivides a recorded trajectory into piecewise linear segments of length 0.5 meters. Further lanes can be added at a given offset, if required. In the case of the GCDC, a left lane was added 3.5 meters next to the right lane, corresponding to the standard highway lane width in the Netherlands. The Mercator coordinates of the vertices are stored in a 2-dimensional kd-tree [34], which is an efficient search structure under Minkowski metrics. Here, we compute exact nearest neighbors using the library *ANN*[3] with respect to the l_2-norm. The kd-tree structure reduces nearest neighbor search complexity from $O(n)$ for the naïve algorithm to $O(\log n)$. Time for constructing the tree, $O(n \log^2 n)$, can be neglected since this needs to be done only once, namely

[3]http://www.cs.umd.edu/~mount/ANN/

when the map is loaded from disc. An example of a kd-tree space decomposition is illustrated in Fig. 4(a) for one of our testing grounds, the Engler-Bunte-Ring in Karlsruhe.

In order to compute distances in between speed limits, traffic lights, or other vehicles on the own lane, we first assign these objects to their closest lane by retrieving the nearest neighbor GPS track vertex. Each center's coordinate is then projected onto the two connected line segments to obtain its *foot point*. The geodesic distance between two objects is readily obtained by summing the segment lengths falling in between those foot points, see Fig. 4(b). For efficiency, we pre-compute geodesic distances of all vertices with respect to the beginning of the respective lane.

B. Vehicles, Traffic Lights, and Speed Limits

All vehicles of the GCDC, including the GCDC leading vehicle, share the same dynamic properties. This allows to describe each vehicle at time t by a state vector

$$(l, w, \phi, \lambda, \psi, v, \dot{\psi}, a)^T$$

with l and w denoting the length and the width of the vehicle respectively, ϕ and λ the GPS position (latitude, longitude) of its geometric center, ψ the heading, v the velocity in direction of heading, and a the acceleration in direction of heading[4]. In the GCDC, each vehicle broadcasts its own state vector including the corresponding GPS time. According to the GCDC rules this information should be precise enough to completely abstain from using other sensors.

Unfortunately, we discovered during the testing weeks that not all teams were able to transmit precise data (the log of one representative run is illustrated in Fig. 6). This mainly led to the following two problems: First, if the position from vehicles physically driving on a neighboring lane were broadcasted to be close to our lane (e.g., due to sensor noise), those cars foot point would be projected onto our lane – in the worst case directly in front of us. Clearly, this could cause an emergency-brake or wrong platooning-behavior. We solved this issue by ignoring all vehicles from the neighboring lane by means of a blacklist, which we manually updated before each run in the competition. Note that this was compliant with the rules of the GCDC and almost all participants made use of it. Second, GPS outage under bridges froze the position of some vehicles, causing our vehicle to stop in cases where we directly followed that vehicle. We were able to solve this issue by using the built-in radar sensor. Once the vehicle ahead was within 50 meters range and tracked with high confidence by radar[5], we put full trust into the radar measurements and ignored any broadcasted position in between our car and the radar's detection.

C. Control Requirements

Our control strategy (see Sec. V) implements a model-predictive controller, which does not only need the current state of other vehicles but also predicted future-states. We

[4]The length and width of a vehicle could also be represented outside of the state vector since it does not change over time

[5]This was not the case at the beginning of a run, when the leading vehicle was standing still.

achieved this by employing a non-linear kinematic model that is based on the assumption of constant yaw-rate and acceleration, corresponding to the movement on a circle. To achieve a smoother behavior of the controller in the velocity limits, prediction is cropped at those values.

Supplementary to vehicle states, the coordinates and the current state of speed limits and traffic lights were broadcasted from road side units. Their coordinates were directly matched onto the map as described in Sec. IV-A and fed into the controller as additional constraints.

V. CONTROL

In the GCDC, performance of the controller would be judged by the following three criteria [35]:

- Speed: Of two competing platoons, the one which first crosses the finish line scores.
- Average platoon length: Should be as small as possible, without violating safety margins.
- Stability: A figure describing stability of the controller was derived from the H_∞ criterion.

Several distinct control related tasks can be identified:

- Low level control transfers desired acceleration into pedal actuation.
- A follow controller stabilizes the desired safety distance to a single leading car.
- A platooning strategy stabilizes a platoon of multiple cars.

After a quick recapitulation of requirements which the GCDC rules impose onto the control strategy, we will deal with each of these tasks in a separate section.

A. Problem Definition and Formalization

Let the platoon consist of N vehicles (only vehicles which are in front of the host vehicle are considered relevant to the platoon). The state of the i-th car in the platoon is described by the vector $\mathbf{x}_i(t) = (x_i(t), \dot{x}_i(t))$, which contains its position and velocity. Here, position is a scalar quantity which describes the distance traveled on a reference path, cf. Sec. IV-A. The system model of a single car is assumed to be a simple double integrator, i.e. it has a single input $u_i(t)$, which is its acceleration, $\ddot{x}(t)$. Cars are ordered by their position, i.e. $i < j \Rightarrow x_i < x_j$. Hence, the host car has index $i = 0$. Let the complete state of the platoon be the tuple $\mathbf{X}(t) = (\mathbf{x}_0, \mathbf{x}_1, \ldots, \mathbf{x}_{N-1})$.

Since the arrangement of the GCDC implies a decentralized platooning strategy, we can only control the acceleration of the host car ($i = 0$). The task of the controller is now to determine the acceleration $u(t)$ for the host vehicle, such that the following conditions hold:

- keep safety distance: $x_0(t) < x_1(t) - r + t_h \dot{x}_1(t)$. Here, t_h is a constant headway time and r (*reserve*) is a constant distance. During the competition, requirements for headway time and reserve distance were 0.6 seconds and 20 meters, respectively.
- keep limits for acceleration a: $-4.5\frac{m}{s^2} < a < 2.0\frac{m}{s^2}$
- keep limits for velocity v: $0 < v < 100\frac{km}{h}$

B. Low Level Controller

Under the assumption that a low level controller is in effect, the host car can be controlled by a single input only, which is its acceleration. Our implementation of this low level controller consists of two feed-forward controllers translating set-point accelerations into virtual actuations for brake and throttle pedals, respectively. This subdivision is advantageous because feed-forward couplings differ largely between both pedals. An integral anti-windup feedback controller compensates for disturbances from wind, slope etc.

C. Follow Controller

The follow controller will determine an optimal acceleration for the host vehicle, based on its current state $(x_0^0, v_0^0) = (x_0(t_0), \dot{x}_0(t_0))$ and the trajectory $x_\text{lead}(t)$ of a single leading car. Indices placed to the upper right will designate discrete time indices in this section, 0 indicating current time, t_0. We assume that $x_\text{lead}(t)$ is given. In practice, we generate it under the assumption that the lead vehicle will drive at constant acceleration, except when velocity limits must be respected. Note that the current acceleration of the vehicles in the platoon is known, since it is part of the communication protocol. Hence, the control law which we derive has a single output, acceleration a, and receives the current position and velocity of the host car, and current position, velocity and acceleration of the leading car as inputs. For reasons which will become clear in the next section, the control law is furthermore parameterized with a specific headway time, $\tilde{t}_h$ and safety reserve, $\tilde{r}$. We will designate it as function k:

$$a = k(x_0^0, v_0^0, x_\text{lead}^0, \dot{x}_\text{lead}^0, \ddot{x}_\text{lead}^0, \tilde{t}_h, \tilde{r}). \tag{1}$$

To determine the optimal acceleration, we minimize the following functional

$$J[u(t)] = \int_{t_0}^{t_0+T} w_\text{dist}[\Delta d(t)]^2 + w_\text{acc}[u(t)]^2 + w_\text{vel}[\Delta v(t)]^2 \mathrm{d}t, \tag{2}$$

where $\Delta d(t) = x_\text{lead}(t) - \tilde{r} + \tilde{t}_h \dot{x}_\text{lead}(t) - x_0(t)$ is the error of the safety distance, $\Delta v(t) = \dot{x}_\text{lead}(t) - \dot{x}_0(t)$ is the velocity difference to the leading car and $u(t) = \ddot{x}_0(t)$ is the sought-after acceleration. The functional is evaluated up to the time horizon T (currently 10 seconds). The functional integrates a weighted sum of the square of these terms, using the weighting factors w_dist, w_acc and w_vel. The first, w_dist-weighted term asserts that the goal of the controller, i.e. reaching the required safety distance, is met. The second, w_acc-weighted one incorporates dampening, by penalizing excessive accelerations. The last, w_{vel}-weighted term can be tuned to avoid overshoot. All weights were tuned during many experimental runs to give a good balance of comfort and speed.

The functional in equation (2) can be minimized in closed form by means of the EULER-LAGRANGE-equation, which leads to a system of RICCATI type equations. This, however, does not allow accounting for the limits of both velocity and acceleration explicitly. We therefore discretize equation (2) by sampling $x_0(t)$ at m equidistant time steps: $x_0^j = x_0(t_0+j\Delta t)$,

$j \in \{0, \ldots m-1\}$. Furthermore, we approximate derivatives $\dot{x}_0$ and $\ddot{x}_0$ at time index j by central finite differences:

$$\dot{x}_0^j \approx \Delta_{\mathrm{c}}^2 x_0^j \quad = \quad \frac{x_0^{j+1} - x_0^{j-1}}{2\Delta t}$$

$$\ddot{x}_0^j \approx \Delta_{\mathrm{c}} x_0^j \quad = \quad \frac{x_0^{j+1} - 2x_0^j + x_0^{j-1}}{[\Delta t]^2}.$$

The functional (2) then becomes a finite sum

$$J_{\mathrm{d}}(x_0^0, x_0^1, ..., x_0^{m-1}) = \sum_{j=1}^{m-2} w_{\mathrm{dist}}[\Delta d_j]^2 + w_{\mathrm{acc}} u_j^2 + w_{\mathrm{vel}}[\Delta v_j]^2 \tag{3}$$

with

$$\Delta d_j \quad = \quad x_{\mathrm{lead}}^j - \tilde{r} + \tilde{t}_h \dot{x}_{\mathrm{lead}}^j - x_0^j$$

$$u_j \quad = \quad \Delta_{\mathrm{c}}^2 x_0^j$$

$$\Delta v_j \quad = \quad \dot{x}_{\mathrm{lead}}^j - \Delta_{\mathrm{c}} x_0^j.$$

Minimization of (3) can be treated as an ordinary extremum problem. Equation (3) is a positive definite quadratic form, and both velocity- and acceleration limits can be expressed as linear inequalities. Hence, the extremum problem is a quadratic program (QP), which can be solved exactly in a finite number of iterations, e.g. using Goldfarb and Idnani's active set method [36]. The desired acceleration can now be reconstructed from the extremum point, again by using finite differencing: $a = \Delta_{\mathrm{c}}^2 x_0^1$.

D. Platooning

Our basic strategy of building a controller which is capable of stabilizing a platoon can be described informally like this:

- For each vehicle in the platoon
  - Consider this vehicle as a single leading vehicle. Using the control law (1), calculate an acceleration based on the state of this vehicle, using a multiple of the safety distance required between adjacent vehicles (the safety distance is multiplied by the integer index of the leading vehicle).
- Out of all these accelerations, choose the *smallest* one (which is the most conservative one).

To assure stable behavior, a small *loose* or *slack* l is added to the safety distance, multiplied by $i-1$, where i is the vehicle index. This assures that, in the steady state, a stable *lock* is established on the leader of the platoon, as can be seen in Fig. 5(a). This lock will only change if one vehicle deviates from its optimum position by an amount greater than l, as has happened in Fig. 5(b). Without the slack, the lock would, in the presence of noise, change very quickly near the steady state, a behavior which could possibly induce oscillation.

On the other hand, when sufficient slack is used, platoon stability follows directly from the stability of the follow controller. Note that Fig. 5(a) and Fig. 5(b), for the sake of clarity, convey the impression that accelerations are determined only based on the distance to vehicles. However, as has been shown in the preceding section, both accelerations and velocities of the vehicles are taken into account as well. If, e.g., in Fig. 5(a) the vehicle with index $i = 2$ was braking very
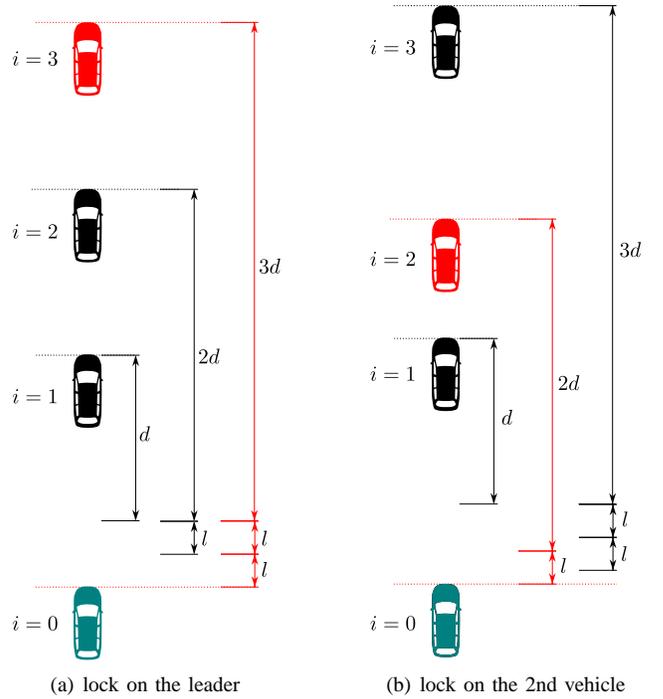
Fig. 5. Platooning strategy for host vehicle (i=0). See text for explanations.

hard, while the others would move uniformly, the lock would switch immediately to the braking vehicle, since it would be the vehicle which enforces the most conservative action, i.e. the highest deceleration of the host vehicle.

VI. RESULTS AND LESSONS LEARNED

Participating in a competition like the GCDC is a highly motivating experience. We focused on making our vehicle run reliably during the competition. This meant that all components like communication, sensor fusion, and control had to work properly and that the vehicle and hardware had to be ready in time. During the competition 15 runs were driven with vehicles assigned randomly to two neighboring lanes. In summary, our vehicle drove very reliably throughout the whole competition and all components worked as expected. In the end, we were awarded first place, just barely beating runners-up team Halmstad. The ranking was based on a set of criteria measuring the contribution of a team to the formation of short and stable platoons, and measuring the ability to follow the lead vehicle as precise as possible. The criteria are described in more details in [3] and [35]. To analyze the system performance in more detail we will describe the performance of the major components in the subsequent paragraphs.

Our V2V- and V2I-communication worked trouble-free throughout the competition. We were able to receive messages up to distances of 800 meters. Although this sounds promising for future applications we have to consider that the test bed of the GCDC on a straight highway is not representative. In a more realistic scenario, problems with occlusions due to buildings and trees next to the road or due to large trucks are to be expected. The bridges over the road that were part of the GCDC test bed already provided ample problems of this sort.
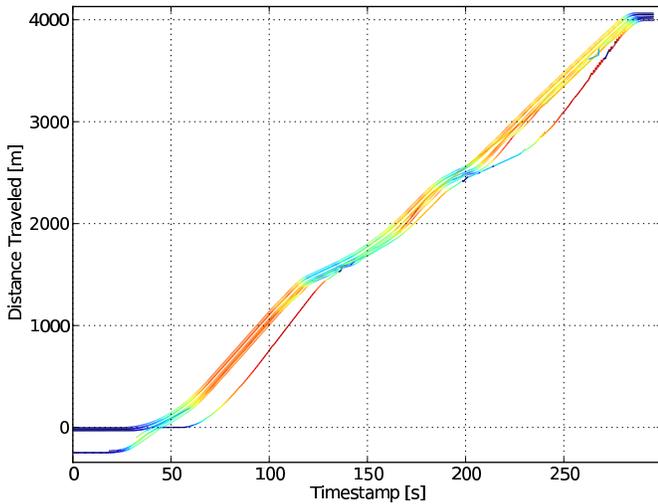
(a) the whole run
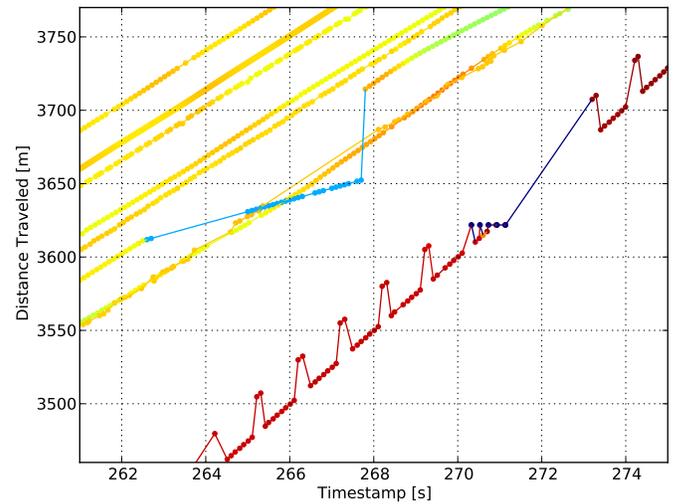
(b) a detail view

Fig. 6. A path-time diagram of a complete run as received by team AnnieWAY. Each line illustrates the distance of one vehicle over time. Travel distance 0 is defined as the starting position of the lead vehicle. The colors encode the velocity of the cars. Warm colors indicate high velocity, cool colors slow velocity. The right plot shows a zoomed in version. Here, some problems become apparent, e.g. partly wrong data sent by the last vehicle. The consequences of a highway bridge are also highlighted, as signal loss is poorly handled by some of the GPS/INS systems.

Additionally, the communication protocol's aim for simplicity led to dropping all security concerns. The CALM protocol offers no encryption or source verification and standard network attacks (man-in-the-middle) are trivial to perform and potentially lethal when wrong data is relied upon by controller strategies. During our preparations for the GCDC we found that, with the current reference implementation of the CALM protocol, a maliciously crafted broadcast package is able to put all clients that were not modified to work with garbage input into an infinite loop or a crash.

One major issue during the competition was the quality of data concerning the position of other vehicles in the heat. Since the GCDC addresses a multi vendor scenario all teams used different GPS/INS systems. Although accuracy requirements were specified in the GCDC rules, the reliability and accuracy of those systems was very different and some systems created position estimates which were very noisy over time. Moreover, in some situations some teams sent outdated data or did not sent anything at all. This behavior has been often observed below bridges, where satellite reception was interrupted. Hence, these vehicles disappeared in our world model or they were mapped to a wrong place. Fig. 6(b) provides an impression of the quality of the data received. The bridge problem becomes apparent at distance 3625 m: some participants only provided a position estimate and constant velocity while others delivered constant velocity and constant position.

Since this problem did not allow safe autonomous operation of our vehicle we decided to integrate the radar sensor into our system and to merge the communicated position of vehicles with the radar targets. Hereby, our policy was to put more trust in the on-board sensors than in communicated positions (see Sec. IV-B). From these experiences we can conclude that additional on-board sensors are indispensable for communication based autonomous driving. Moreover, this shows that certified, high-quality GPS/INS sensors are required to enable
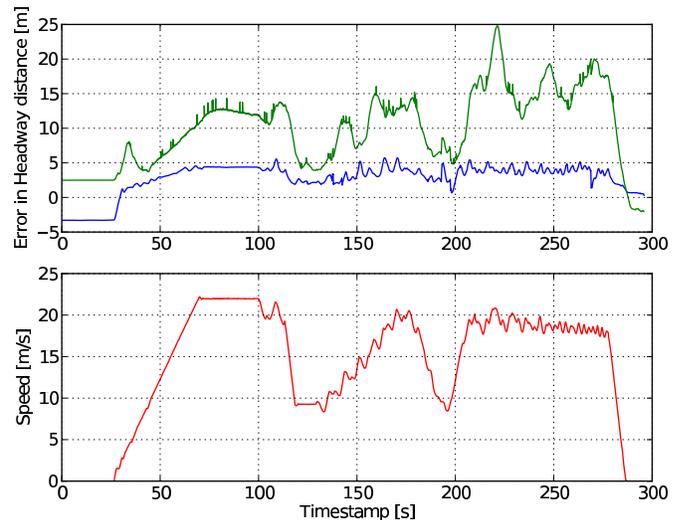
Fig. 7. The top plot shows desired headway distance minus real headway distance for AnnieWAY (blue) and Halmstad (green) in a run where both were directly behind the lead vehicle. The bottom plot shows the speed profile of the lead vehicle in the same run.

safe operation.

Longitudinal control of our vehicle worked satisfactory. Fig. 7 shows the performance of our vehicle following the lead vehicle during one heat of the GCDC. The controller reacted smoothly with small latencies to changes of the lead vehicle. The cooperative platooning control also worked well. Since the performance of a platoon depends on all vehicles belonging to the platoon it is hard to measure the contribution of a single vehicle. However, the overall result of the GCDC, which was obtained by averaging over 15 heats and distributing participants in various ways, indicates that our platooning controller contributed to compact platoons on average.

Since our controller was designed in a conservative way, it did not assume any properties of the controllers in other

vehicles. This fits very well to the multi vendor scenario of the GCDC. Certainly, knowing the control policies of other vehicles would offer great potential for further improvements. However, such an assumption would be far from being realistic considering real traffic applications.

Another concept implemented in the GCDC was explicit platoon joining, i.e. platoons are arranged explicitly sending join and confirmation messages between the vehicles. Our vehicle supported these messages to comply with the rules. However, we did not make use of the information whether or not a vehicle formally joined our platoon. In the light of our experiences during the GCDC, the concept of explicit joining a platoon seems to be of limited use for highway scenarios.

Our participation in the GCDC has been one further step on our way towards fully autonomous driving. Our next activities will again focus more on improved environment perception as it turned out that only vehicles with reliable on-board sensors can act safely. In our view, communication-based strategies are only useful as supplement to local perception.

References

[1] G. Marsden, M. McDonald, and M. Brackstone, "Towards an understanding of adaptive cruise control," *Transportation Research Part C*, vol. 9, no. 1, pp. 33–51, 2001.

[2] A. Kesting, M. Treiber, M. Schönhof, F. Kranke, and D. Helbing, "Jam-avoiding adaptive cruise control (ACC) and its impact on traffic dynamics," in *Traffic and Granular Flow*, 2005, pp. 633–643.

[3] E. van Nunen, M. R. Kwakkernaat, J. Ploeg, and B. D. Netten, "Cooperative competition for future mobility," *IEEE Transactions on Intelligent Transportation Systems*, vol. *same issue*, 2012.

[4] S. Shladover, "Cooperative (rather than autonomous) vehicle-highway automation systems," *Intelligent Transportation Systems Magazine, IEEE*, vol. 1, no. 1, pp. 10 –19, spring 2009.

[5] C. Thorpe, T. Jochem, and D. Pomerleau, "The 1997 automated highway free agent demonstration," in *Proc. IEEE Intelligent Transportation Systems Conference*, Boston, USA, 9–12 November 1997, pp. 496–501.

[6] S. Shladover, C. Desoer, J. Hedrick, M. Tomizuka, J. Walrand, W.-B. Zhang, D. McMahon, H. Peng, S. Sheikholeslam, and N. McKeown, "Automated vehicle control developments in the path program," *IEEE Transactions on Vehicular Technology*, vol. 40, no. 1, pp. 114 –130, Feb. 1991.

[7] S. Kato, S. Tsugawa, K. Tokuda, T. Matsui, and H. Fujii, "Vehicle control algorithms for cooperative driving with automated vehicles and intervehicle communications," *IEEE Transactions on Intelligent Transportation Systems*, vol. 3, no. 3, pp. 155 – 161, Sep. 2002.

[8] C. Bonnet, "Chauffeur 2 final report," Deliverable D24, Version 1.0, Contract IST-1999-10048, Jul. 2003.

[9] C. Stiller, G. Färber, and S. Kammel, "Cooperative cognitive automobiles," in *Proc. IEEE Intelligent Vehicles Symposium*, Istanbul, Turkey, Jun. 2007, pp. 215–220.

[10] C. Frese, J. Beyerer, and P. Zimmer, "Cooperation of cars and formation of cooperative groups," in *Proc. IEEE Intelligent Vehicles Symposium*, Istanbul, Turkey, Jun. 2007, pp. 227–232.

[11] V. Milanés, J. Alonso, L. Bouraoui, and J. Ploeg, "Cooperative maneuvring in close environments among cybercars and dual-mode cars," *IEEE Transactions on Intelligent Transportation Systems*, vol. 12, no. 1, pp. 15–24, 2012.

[12] O. Aycard, Q. Baig, S. Bota, F. Nashashibi, S. Nedevschi, C. Pantilie, M. Parent, P. Resende, and T.-D. Vu, "Intersection safety using lidar and stereo vision sensors," in *Proc. 2011 IEEE Intelligent Vehicles Symposium (IV)*, Jun. 2011, pp. 863 –869.

[13] B. Roessler, "Status of european project intersafe-2 on cooperative intersection safety," in *Proc. 2010 IEEE International Conference on Intelligent Computer Communication and Processing (ICCP)*, Aug. 2010, pp. 381 –386.

[14] Ü. Özgüner, C. Stiller, and K. Redmill, "Systems for safety and autonomous behavior in cars: The DARPA Grand Challenge experience," *IEEE Proceedings*, vol. 95, no. 2, pp. 1–16, Feb. 2007.

[15] S. Kammel, J. Ziegler, B. Pitzer, M. Werling, T. Gindele, D. Jagzent, J. Schröder, M. Thuy, M. Goebl, F. von Hundelshausen, O. Pink, C. Frese, and C. Stiller, "Team AnnieWAY's autonomous system for the 2007 DARPA urban challenge," *Journal of Field Robotics*, vol. 25, no. 9, pp. 615 – 639, 2008.

[16] A. Geiger, M. Roser, and R. Urtasun, "Efficient large-scale stereo matching," in *Asian Conference on Computer Vision*, 2010, pp. 25–38.

[17] B. Ranft, T. Schoenwald, and B. Kitt, "Parallel matching-based estimation – a case study on three different hardware architectures," in *IEEE Intelligent Vehicles Symposium*, 2011, pp. 1058–1065.

[18] A. Geiger, J. Ziegler, and C. Stiller, "Stereoscan: Dense 3d reconstruction in real-time," in *IEEE Intelligent Vehicles Symposium*, 2011, pp. 963–968.

[19] H. Lategahn, A. Geiger, and B. Kitt, "Visual SLAM for autonomous ground vehicles," in *IEEE International Conference on Robotics and Automation*, 2011, pp. 150–156.

[20] P. Lenz, J. Ziegler, A. Geiger, and M. Roser, "Sparse scene flow segmentation for moving object detection in urban environments," in *IEEE Intelligent Vehicles Symposium*, 2011, pp. 926–932.

[21] A. Geiger, M. Lauer, and R. Urtasun, "A generative model for 3d urban scene understanding from movable platforms," in *Computer Vision and Pattern Recognition (CVPR)*, 2011, pp. 1945–1952.

[22] A. Geiger, C. Wojek, and R. Urtasun, "Joint 3d estimation of objects and scene layout," in *Neural Information Processing Systems (NIPS)*, 2011.

[23] F. Moosmann, O. Pink, and C. Stiller, "Segmentation of 3d lidar data in non-flat urban environments using a local convexity criterion," in *IEEE Intelligent Vehicles Symposium*, 2009, pp. 215–220.

[24] F. Moosmann and T. Fraichard, "Motion estimation from range images in dynamic outdoor scenes," in *IEEE International Conference on Robotics and Automation*, 2010, pp. 142–147.

[25] F. Moosmann and C. Stiller, "Velodyne SLAM," in *IEEE Intelligent Vehicles Symposium*, 2011, pp. 393–398.

[26] J. Ziegler and C. Stiller, "Fast collision checking for intelligent vehicle motion planning," in *IEEE Intelligent Vehicles Symposium*, 2010, pp. 518–522.

[27] J. Ziegler and C. Stiller, "Spatiotemporal state lattices for fast trajectory planning in dynamic on-road driving scenarios," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2009, pp. 1879–1884.

[28] J. Ziegler, M. Werling, and J. Schröder, "Navigating car-like vehicles in unstructured environment," in *Proceedings of the IEEE Intelligent Vehicles Symposium*, 2008, pp. 787–791.

[29] M. Werling, J. Ziegler, S. Kammel, and S. Thrun, "Optimal trajectory generation for dynamic street scenarios in a frenet frame," in *IEEE International Conference on Robotics and Automation*, 2010, pp. 987–993.

[30] M. Goebl und Georg Färber, "A real-time-capable hard- and software architecture for joint image and knowledge processing in cognitive automobiles," in *IEEE Intelligent Vehicles Symposium*, 2007, pp. 734–740.

[31] "IEEE Standard for Information technology-Telecommunications and information exchange between systems-Local and metropolitan area networks-Specific requirements - Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications," *IEEE Std 802.11-2007 (Revision of IEEE Std 802.11-1999)*, pp. C1–1184, 2007.

[32] D. Jiang and L. Delgrossi, "IEEE 802.11p: Towards an international standard for wireless access in vehicular environments," in *IEEE Vehicular Technology Conference Spring 2008*, 2008, pp. 2036–2040.

[33] ISO, "Intelligent Transport Systems - Communications access for land mobiles (CALM) - Architecture, work in progress," *ISO Technical Committee 204 WG16, Tech. Rep. ISO/DIS 21217*, 2009.

[34] S. Arya, D. M. Mount, N. S. Netanyahu, R. Silverman, and A. Wu, "An optimal algorithm for approximate nearest neighbor searching," in *Fifth annual ACM-SIAM symposium on Discrete algorithms*, 1994, pp. 573–582.

[6]http://www.fzi.de

[35] GCDC organization, "GCDC 2011. rules & technology document," available from: `http://www.gcdc.net/mainmenu/Home/technology/Rules_and_Technology`, 2011.

[36] D. Goldfarb and A. Idnani, "A numerically stable dual method for solving strictly convex quadratic programs," *Mathematical Programming*, vol. 27, pp. 69–88, 1983.

Andreas Geiger received a Diploma degree in computer science from the University of Karlsruhe in 2008. Currently, he is a Ph.D. candidate at the Institute of Measurement and Control at Karlsruhe Institute of Technology. His research interests include computer vision and scene understanding.

Martin Lauer studied computer science at Karlsruhe University and received his Ph.D. in computer science from University of Osnabrück. As a senior research fellow he works in the fields of machine vision, robotics, and machine learning at Karlsruhe Institute of Technology.

Frank Moosmann received a Diploma degree in computer science from the University of Karlsruhe in 2006. Currently, he is a Ph.D. candidate at the Institute of Measurement and Control at Karlsruhe Institute of Technology. His research interests include computer vision, machine learning and autonomous vehicles. As there is plenty of free space, he is happy to greet all his friends.

Benjamin Ranft received a Diploma degree in mechatronics from Karlsruhe Institute of Technology in 2009 and is currently working towards his PhD at FZI Research Center for Information Technology. His thesis' topic is auto-adaption to heterogeneous parallel hardware for real-time computer vision. Other research interests include autonomous vehicles and advanced driver assistance systems.

Holger Rapp studied Physics at the University of Heidelberg and is currently pursuing his Ph.D. at the Institute for Measurement and Control at Karlsruhe Institute of Technology (KIT). The topic of his thesis is 3-dimensional metrology of highly reflecting surfaces. Other research interests include time-of-flight cameras, real-time robotics, image processing, rapid prototyping and autonomous vehicles.

Christoph Stiller is professor at the Karlsruhe Institute of Technology and head of the Institute of Measurement and Control. At the same time he is director at the FZI Research Center for Information Technology heading the mobile perception research group. His research focus is in stereo vision, camera based scene understanding, driver assistance systems, and autonomous vehicles.

Julius Ziegler received a Diploma degree in computer science from the University of Karlsruhe in 2006. Currently, he is a Ph.D. candidate with the Institute of Measurement and Control at Karlsruhe Institute of Technology. He is a founding member of Team AnnieWAY, with which he participated in the DARPA Urban Challenge of 2007. His research interest is autonomous driving in general, and trajectory planning and control in particular.