Toward Automated Driving in Cities using Close-to-Market Sensors
an overview of the V-Charge project*

Paul Furgale[†1], Ulrich Schwesinger[1], Martin Rufli[1], Wojciech Derendarz[2], Hugo Grimmett[6], Peter Mühlfellner[2], Stefan Wonneberger[2], Julian Timpner[3], Stephan Rottmann[3], Bo Li[5], Bastian Schmidt[2], Thien Nghia Nguyen[2], Elena Cardarelli[9], Stefano Cattani[9], Stefan Brüning[7], Sven Horstmann[7], Martin Stellmacher[8], Holger Mielenz[4], Kevin Köser[5], Markus Beermann[4], Christian Häne[5], Lionel Heng[5], Gim Hee Lee[5], Friedrich Fraundorfer[5], René Iser[2], Rudolph Triebel[6], Ingmar Posner[6], Paul Newman[6], Lars Wolf[3], Marc Pollefeys[5], Stefan Brosig[2], Jan Effertz[2], Cédric Pradalier[1], and Roland Siegwart[1]

Abstract— **Future requirements for drastic reduction of CO_2 production and energy consumption will lead to significant changes in the way we see mobility in the years to come. However, the automotive industry has identified significant barriers to the adoption of electric vehicles, including reduced driving range and greatly increased refueling times.**

Automated cars have the potential to reduce the environmental impact of driving, and increase the safety of motor vehicle travel. The current state-of-the-art in vehicle automation requires a suite of expensive sensors. While the cost of these sensors is decreasing, integrating them into electric cars will increase the price and represent another barrier to adoption.

The V-Charge Project, funded by the European Commission, seeks to address these problems simultaneously by developing an electric automated car, outfitted with close-to-market sensors, which is able to automate valet parking and recharging for integration into a future transportation system. The final goal is the demonstration of a fully operational system including automated navigation and parking. This paper presents an overview of the V-Charge system, from the platform setup to the mapping, perception, and planning sub-systems.

I. INTRODUCTION

As part of their "Europe 2020" program, the European Commission has outlined a number of ambitious targets for Europe to meet by the year 2020 [1]. These targets address a wide range of social, environmental, and economic issues. Part of the strategy is to address the problem of climate change, to reduce greenhouse gas emissions, to move toward renewable sources of energy, and to increase energy efficiency.

One aspect of this challenge will be the reduction in reliance on fossil fuels and the move to electric motor vehicle transport. However, the automotive industry has identified significant barriers to the electrification of vehicles, including reduced driving range and increased refueling times [2].

Automated cars have the potential to reduce the environmental impact of driving, reduce traffic jams, and increase

Fig. 1. The initial experimental platforms for the V-Charge project—each VW Golf has been modified to support fully automated driving using only close-to-market sensors. A similar plug-in hybrid is being developed.

the safety of motor vehicle travel [3]. The current state-of-the-art in automated vehicle technology requires precise, expensive sensors such as differential global positioning systems, highly accurate inertial navigation systems and scanning laser rangefinders [3]. While the cost of these sensors is going down as robots become more ubiquitous, integrating them into electric cars will increase the price and represent yet another barrier to adoption.

The European V-Charge Project seeks to address these problems simultaneously by developing an electric automated car, outfitted with close-to-market sensors, which is able to automate valet parking and recharging for integration into a future transportation system. To provide a balance between individual and public transportation, such a system could be used to support coordinated parking, charging, and pickup of vehicles for park-and-ride public transit.

This implies three major fields of research: (i) *vehicle functionality*, onboard localization detection of static and dynamic obstacles, and on-board planning using only close-to-market sensors, (ii) *logistics*, optimal scheduling of charging stations and assignment of parking spots, and (iii) *infrastructure*, development of a secure and reliable communication framework to store and share a database of information about the parking area.

After a short section concerning the state of the art, this paper will present an overview of the V-Charge project and its goals, starting with a description of the platform and infrastructure setup, followed by a section about the mapping, perception and planning software components. Results from year one of the project are presented within each section.

[†] Corresponding Author, paul.furgale@mavt.ethz.ch, [1] Autonomous Systems Lab, ETH Zürich, [2] Volkswagen AG, Group Research, [3] IBR, Technische Universität Braunschweig, [4] Robert Bosch GmbH, Corporate Research, CR/AEV, CR/AEM, [5] Computer Vision and Geometry Group, ETH Zürich, [6] University of Oxford, [7] Carmeq GmbH, [8] IAV GmbH, [9] VisLab, Università degli Studi di Parma

A. State of the art

The DARPA Grand Challenge (2004, 2005) and Urban Challenge (2007) [4] competitions were instrumental in pushing research on automated driving out of the lab and into near real-world conditions. As such, they remain the de facto baseline for automated driving systems. In both cases, the vehicle had to drive fully autonomously in either an off-road environment or an urban environment. Among other competencies, these tasks required local obstacle perception and tracking, local path planning to avoid collisions as well as accurate global localization to enable progress toward high-level mission goals. All of the successful teams in these competitions utilized a highly sophisticated suite of expensive sensors, such as sweeping laser range finders, RADAR systems, and color cameras, most of them pointing ahead to detect the road parameters and potential obstacles (e.g. [5], [6], [7]). Most vehicles also used one GPS antenna in combination with a 6 degree of freedom inertial measurement unit (IMU) for localization as well as two additional GPS antennae to discern absolute heading. In addition, in the Urban Challenge, the cars had to select their own routes, perceive and interact with other traffic, execute lane changes, U-turns and parking maneuvers.

However, the sensor setup in the previously mentioned projects is much too costly to consider inclusion into series automobiles. The research initiative PReVENT [8] is one step closer to market-ready automated vehicles. In this project low-cost sensors such as cameras or radio-based car to car communication were used. Similarly, [9] describes a full architecture for decision making under uncertainty during autonomous city driving and provides experiments showing the effectiveness of their approach in simulations of many real traffic situations. However, in both cases, long distance fully automated driving was not demonstrated.

The Park Assist system by Volkswagen[1] is an example of an automatic driving application already on the market. This system assists the driver with maneuvering the vehicle in parallel or head in parking spots. The automatic parking mode is enabled in collaboration with the driver but using stock sensors only.

Consequently, fully automated driving in dynamic urban environments using only close-to-market sensors and onboard computation remains an open research challenge.

II. PLATFORM OVERVIEW

This section gives an overview of the hardware and software platforms used in the V-Charge project.

A. Hardware

Currently, a modified conventional combustion engine VW Golf VI is used as the test platform of the project. The modifications include integration of a sensor array used as data source for environment perception and ego-localization, installation of a computer cluster responsible for the control

---

[1]http://www.volkswagenag.com/content/vwcorp/content/en/innovation/driver_assistance/parking_steering_assistance.html

---

of the vehicle, adaptation of vehicle ECU network enabling the drive-by-wire operation, and installation of additional safety elements. We are working on analogous modifications of a plug-in hybrid vehicle that will serve as the final test platform for the project.
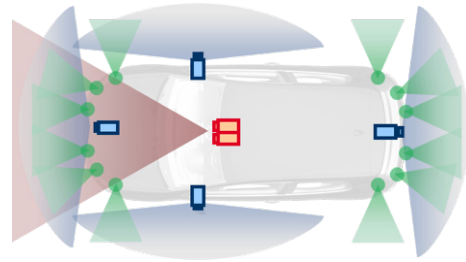

Fig. 2. The sensor setup of the V-Charge test vehicles including a schematic representation of the field of view of the individual sensors (green is used for sonar sensors, red for stereo-camera system, and blue for mono fish-eye cameras).

Fig. 2 gives an overview of the sensor system installed on the V-Charge test vehicles. Currently it consists of 12 sonar sensors responsible for obstacle detection in short range, a 45° field of view (FOV) front stereo camera used for obstacle perception, and a set of 4 fish-eye cameras providing a 360° imagery of the vehicle surroundings. A rear-facing stereo sensor will be integrated in the future. The vehicle is also equipped with a standard GPS receiver and the onboard sensor cluster including stock odometers, accelerometers, and gyroscopes.

Each of the mono-cameras has a nominal FOV of 185° and 1.3Mpx resolution. They are synchronously triggered at a 12.5Hz. The stereo sensor is described in Sec. V-B.1. Each sonar sensor has an aperture angle of 60° horizontally and 30° vertically and range of about 3–4.5m. The sonar system provides the first and second direct and indirect reflections for each sensor at about 10Hz.

The computer cluster consists primarily of 6 PCs installed in the trunk of the vehicle. It can be powered directly from the alternator or—if the engine is off—from the standard 12V DC battery.

The automated driverless operation of the V-Charge test vehicles is achieved by using stock actuators only. Even though the steering, throttle, brakes, and gearbox of a regular VW Golf VI are generally controllable from within the CAN network, some significant modifications to the vehicle were necessary. These included (i) modification of the network topology to enable two-way communication with the computer cluster, (ii) adaptation of the engine ECU to enable the availability of the Automatic Cruise Control (ACC) interface at low-speeds, and (iii) installation of an electronic parking brake.

In order to assure safe operation of the vehicle during the test phases and demonstrations a number of additional safety elements have been installed in the vehicle. The most important elements and principles include (i) a remote kill switch that will disconnect the V-Charge system from the ECU network and initiate an emergency stop, (ii) an onboard kill switch that will disconnect the V-Charge system from

the ECU network to enable safe manual driving, (iii) control inputs may be overridden by the driver at any time, and (iv) the control inputs are monitored for integrity and filtered by the CAN gateway.

In order to assure the highest possible safety level of the system, a Failure Mode and Effects Analysis will be performed with an independent organization.
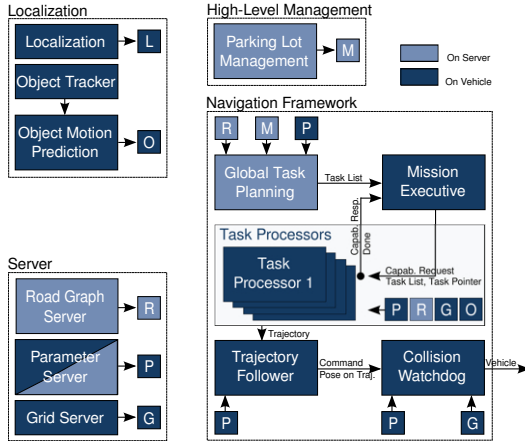
B. Software

Fig. 3. The V-Charge navigation architecture including adjacent modules to highlight the relevant interfaces. Modules colored in light blue operate on the central server side, those colored in dark blue on each automated car.

Fig. 3 displays the conceptual layout of the overall V-Charge navigation framework. It shares key aspects with the architectures employed by the top DARPA Urban Challenge finishers discussed in Sec. I-A, including a behavioral layer that handles priority between a set of specialized planning instances (called *task processors*).

We distinguish modules operating on the *central server side* (light blue) from those on each of the *automated vehicles* (dark blue). The *parking manager* processes requests of incoming and outgoing vehicles. It will assign free parking spots and charging areas by considering charging needs and expected parking time. The *global task planner* operates on a regularly updated road graph (obtained from the road graph server). It is responsible for topological route planning and task assignment. The *mission executive* located on vehicle side is responsible for task assignments to the individual task processors, management of task processor exceptions and the overall correct execution of events.

III. INFRASTRUCTURE, COMMUNICATION AND MANAGEMENT

Since the number of charging stations at large parking areas, due to cost reasons, will be limited, the search for an available (and charging-capable) parking spot will be typically even more complicated and time-consuming for electric vehicle (EV) drivers than for drivers of internal combustion engine (ICE) cars. V-Charge therefore provides an automated parking and charging system, based on a central back-end server [10] which is in charge of an efficient
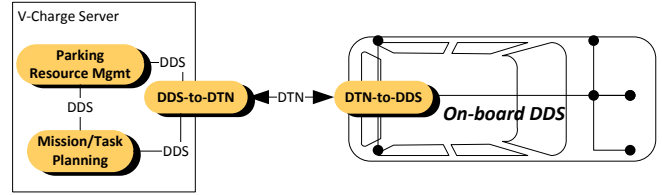

Fig. 4. Mission control architecture. A disruption tolerant network will be used for vehicle to infrastructure communication and the DDS middleware will be used for communication within individual components.

parking resource management. It also provides each vehicle with relevant mission information allowing it to navigate to its assigned target destination.

To enable this system functionality, two main contributions to the management and infrastructure part of the project are made. First, the above-mentioned concepts for efficient parking management are developed. Based on driver requirements, e.g., prospective parking time, current battery charging level and required travel distance, the Java EE-based V-Charge server assigns (schedules) available parking resources, such as regular parking spots and, in particular, scarce charging stations to connected vehicles. The resulting scheduling algorithms are being evaluated in a simulation environment with real-world parking statistics to learn their suitability for different usage scenarios (e.g., downtown vs. airport parking). Requirements for charging station scheduling as well as a short overview of first evaluation results are given in [11]. Second, a sophisticated Disruption Tolerant Networking (DTN) framework for vehicle-to-infrastructure (V2I) and vehicle-to-vehicle (V2V) (both terms are often subsumed as vehicle-to-x, or V2X) communications is developed. This framework enables the distribution of mission information to connected vehicles. Of course, state-of-the-art security and trust concepts are factored in. Driver interaction (status check, drop-off, pick-up) is realized via mobile user devices (smartphones). An overview is given in Fig. 4.

IV. MAPPING

In order to build a system suitable for both outdoor and indoor parking places, we have designed a layered topological/metric map to support both localization and planning. The sparse map (Sec. IV-A) is built from a state of the art Simultaneous Localization and Mapping pipeline. It defines the coordinate frames in which all other map data is expressed and provides the geometric and appearance data needed for vehicle localization. The dense map (Sec. IV-B) encodes a ground plane and height map representing the static structure of the scene. Finally, the road graph (Sec. IV-C) represents the abstract graph of connected lanes, parking spots, and other semantic annotations (Sec. IV-D).

A. Sparse map

The sparse map represents the world as a trajectory of vehicle poses and a set of sparse 3D map points. The 3D map points correspond to landmarks in the world, which are observed as keypoints in the camera images taken from its respective car poses. We generate the sparse map offline
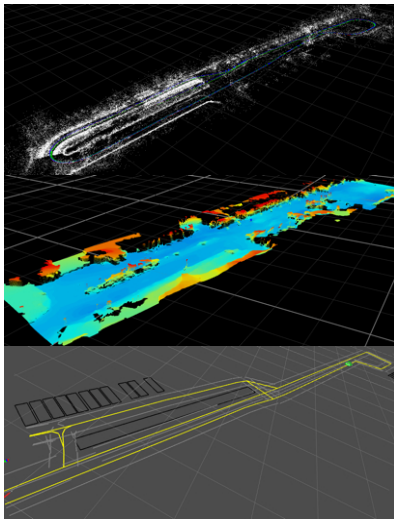
Fig. 5. The layered map concept used in the project. **Top:** Sparse Map—a graph of vehicle poses with three-dimensional landmarks encode the appearance and geometric information. The sparse map defines the coordinate system for all further mapping data and contains all of the geometric and appearance information needed for localization. **Middle:** Dense Map—a single layer height map encoding the static structure of the scene. **Bottom:** Road Network—an abstract graph of lanes used for mission planning and semantic annotations.

Fig. 6. Depth map computation–perspective images (middle) are rendered from the fisheye images (left) and plane sweep stereo matching is used between three temporally consecutive images to produce a depth map (right) for each camera at each time.

from a sequence of images collected from the four fisheye cameras during a drive through the environment. It is used as a topometric map for online localization (Sec. V-A) and as the input for creating a dense map (Sec. IV-B). We use the OpenCV GPU implementation of SURF [12] to extract keypoints and descriptors from every image. The keypoints from each camera are tracked through the sequence of images by matching the descriptors over consecutive frames using K-nearest-neighbors matching to keep processing time low. Mismatched keypoints from two consecutive frames are rejected based on the essential matrices [13] computed from the wheel odometry readings and extrinsic values of the cameras. Remaining consistent keypoint correspondences over two consecutive frames are triangulated to infer the corresponding 3D map point's coordinates. Finally, the estimations of the sparse 3D map points and the trajectory of car poses are improved with a full Bundle Adjustment [13] where the total reprojection errors are minimized. Fig. 5 shows an example of the 3D map points (white points) and the trajectory of car poses (RGB color for the x, y and z axis of each frame) from the sparse map after full bundle adjustment.

B. Dense map

For path planning tasks such as the generation of the road graph or parking spot detection, the sparse map does not represent the world densely enough. Therefore our map also contains a dense height map layer. In this way the height profile in the areas where the car is moving can be described accurately, while avoiding the computationally expensive computation of a full dense 3D model.

As we do not have much overlap between the four fisheye cameras, we perform motion stereo on three consecutive

images for each of the four cameras, using the middle one as reference (Fig 6). For now, we first extract perspective images out of the fisheye images and subsequently run plane sweep stereo matching [14] to get depth maps. As an output we get four depth maps per vehicle pose which are used as input to a fusion procedure.

For the fusion of these local noisy depth maps, we closely follow the approach of [15], but instead of the two-layer height map useful for indoor scenarios, so far we compute a single layer height map of the environment.

The method consists of the following three steps. First, depth maps are computed and entered into a volumetric grid. Each voxel of the grid stores information about its occupancy likelihood. As a next step a raw height map together with information about the certainty of a given height are extracted out of the grid. The extraction of the heights from the grid happens point-wise, without looking at the neighboring heights. As a last step, a global convex optimization is run on the 2D height map to introduce spatial smoothness.

In [15] the total variation (TV) is used to penalize the height differences between neighboring places in the height map. To reduce the staircasing artifacts in the final height map we replace the TV with the Huber-TV. In Fig. 5 the final output of the fusion is depicted.

C. Road Network

The traffic infrastructure considered for local and global path planning is represented by an efficient data structure named RoadGraph [16], [17]. The RoadGraph is a directed graph comprising nodes connecting adjacent edges. Consequently, efficient graph search algorithms may be applied to find topological paths from the current vehicle pose to any final pose in the world.

Roads are represented by a set of edges subdivided according to the driving direction. Edges in the graph are placed at lane centers by assigning interpolation points consisting of sparse map coordinates to each edge. It is important to emphasize that a node in the RoadGraph is only a means to connect consecutive edges logically. Thus, a node does not constitute any place in the world. Intersections are modeled as sets of edges as well. Here the edges constitute either approaching lanes, lanes leaving the intersection, or lanes located on the intersection.

An example is illustrated in Fig. 5. It shows a RoadGraph of our test site at ETH Zürich. The yellow lines represent the edges (lanes) while the gray lines constitute the road borders. Parking spots are marked by the black rectangles. Note that the edges of the RoadGraph are very sparse, i.e.
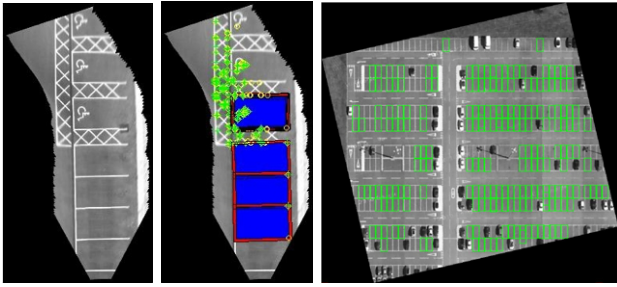
Fig. 7. **Left:** Overhead image obtained from single images of an onboard camera. **Center:** Parking spot detection results using template matching. **Right:** Parking spot detection results applied to an overhead image from Google Maps.

only very few interpolation points were used to model the lanes. Consequently, the edges are not driveable for non-holonomic vehicles and a global path planning operation smoothing the edges is required (cf. Sec. VI-A).

D. Semantic layer

As an essential requirement for the navigation and path planning process, a robust automated *semantic annotation* module is developed, which operates both on raw sensor input and on the dense map. The major aim of semantic annotation is to provide information about parking spaces, driveable areas, and obstacles such as curbs. To do this, we first obtain an overhead image of the environment, where the vehicle is deployed. This can be a satellite image from the area, or a compound image consisting of image frames taken previously from on-board cameras and projected onto the ground plane obtained from the dense map. Then, we apply template matching to find parking spots in the overhead images. Fig. 7 shows example results of this detection. We can see that the detection is in general very robust, with some slight exceptions where the template matcher did not find enough evidence for a parking spot. We plan to address these issues using an approach based on probabilistic graphical modals, which uses context information to improve the detection, in a similar way as was done by Spinello *et al.* [18]. Also, a classification method is currently under investigation, which provides uncertainty estimates (see, e.g. [19]). These can then be used to improve the classification and to obtain more detailed information for navigation and planning.

V. PERCEPTION

This section describes the V-Charge perception system, which provides both localization within the sparse map (Sec. V-A), and situational awareness (Sec. V-B) using only close-to-market sensors.

A. Localization

The localization module provides the pose of the vehicle within the coordinate frame defined by the sparse map (Sec. IV-A). Fig. 8 shows the localization pipeline.

For robust data association between 2D points from the images and 3D points from the map, SURF keypoints and
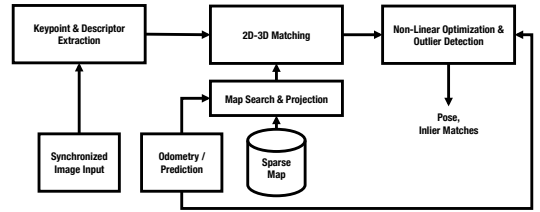

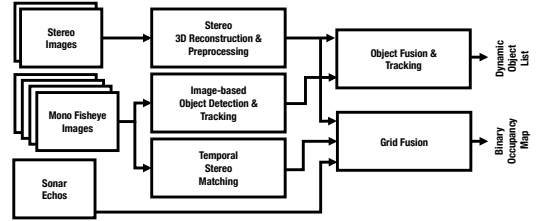Fig. 8. The pipeline used for online data-association with the sparse map and localization.

Fig. 9. The perception framework used to generate a detailed reconstruction and consistent view of the scene

descriptors are extracted from each currently observed image. Then, a pose predicted based on wheel odometry is used to select keypoints from close-by topological nodes in the sparse map. The chosen keypoints are projected into the image plane, resulting in a second set of sparse image-points. The two sets of points—from the current observation, and from the projected map—are matched based on their distance in image- and descriptor-space, providing a set of correspondences between the image and the map. The pose is estimated by minimizing the total reprojection error in all images. A robust cost function is used to deal with outliers.

To initialize the system, we allow for more extensive data associations between the sparse map and the observed frame by using NP3P-fitting (the Non-Perspective 3-Point Problem) combined with RANSAC [20], followed by a linear NPnP step for refinement. Both problems are solved using the gP3P/gPnP method presented by Kneip et al. [21].

B. Situational Awareness

For automated driving in rapidly changing and dynamic environments, a robust and accurate scene reconstruction and situation analysis is mandatory. Therefore a complete 360 degree sensor coverage is required and realized with the sensor system shown in Fig. 2. Fig. 9 sketches the workflow of the perception framework. In detail, each module provides the following functionality:

Fig. 10. The Bosch stereo video camera detects objects' height and distance as well as capturing standard video images.

1) Dense Stereo : With a 12-centimeter baseline distance, the Bosch stereo camera may well be the smallest system of its kind currently available in the field of automotive solutions. Each of the two CMOS image sensors has a resolution of 1.2 megapixels. Thanks to its high-quality lens system, the camera is able to capture an angle of view of 25 degrees vertically and 45 degrees horizontally, and offers a 3D measurement range in excess of 50 meters. The highly light-sensitive image sensors are capable of processing very high contrasts and cover the full spectrum of light visible to the human eye.

For V-Charge, internal distance data from the Bosch stereo camera system is made available to the map fusion module. The data is represented in a fashion similar to a laser scanner: for each direction horizontally, distances to obstacles—if any—are provided. Beyond the ability of a laser scanner, the stereo camera is robust against changes in the vehicle's pitch angle or a rising road surface. In addition to their distance, obstacles are associated with attributes like motion, height, and visual appearance. Based on this information, static obstacles can be aggregated in a map, dynamic obstacles clustered and tracked, and their visual appearance used for classification.

2) Temporal Stereo Matching : To extend the FOV of the front and rear facing stereo camera we run temporal stereo matching on the side-facing fisheye cameras. The stereo matching is done the same way as in the offline mapping phase (Sec. IV-B), using our GPU plane sweep implementation. Camera poses are provided by the online localization pipeline. We can currently compute depth maps with a 129.8° FOV horizontally and 106.3° vertically with a 640×400 resolution. For the final online system we plan to preprocess the data to provide the extracted data in a similar way as the stereo camera does.

3) Object Detection and Tracking : The obstacle detection and tracking phase is focused on pedestrian and vehicle recognition, through the application of vision based algorithms. An AdaBoost classifier is applied on specific region of the input images to determine the presence of a vehicle or pedestrian: the areas on which to apply the classifiers are determined according to typical objects sizes and calibration parameters. Thus, for each input image, the area that a specific obstacle at a specific distance occupied in the image is determined. Obstacle tracking between images is performed to determine static and dynamic elements.

Different AdaBoost classifiers has been trained and tested, using samples directly obtained from the acquired images. Thousands of samples of front and rear vehicle faces have been collected and used to train the classifier. The obstacle detector is based on the trained classifier and is able to detect vehicles from frontal and rear faces. Some preliminary results relative to rear and front vehicles detection are shown in Fig. 11. The work has been performed using the Parma GOLD framework using perspective images rendered from the fisheye images.

The use of rendered perspective images introduces noise and imprecision in vehicles detection (Fig. 11, right). Further

Fig. 11. Front and rear vehicles detection on the acquired images. In the right example, the image distortion affects the recognition of a vehicle.
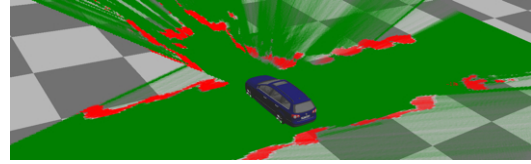

Fig. 12. Occupancy grid for a typical parking lot scene after vehicle has passed several occupied spaces and positioned itself in front of an empty one.

training sessions and tracking procedure are needed to improve the reliability of the results and cope with the presence of inaccurate detections.

4) Map Fusion and Dynamic Objects : In order to reconstruct a consistent model of the environment, the pre-processed sensor measurements are aggregated over time in an ego-referenced occupancy grid. Such approaches go back to the early works of e.g. [22]. In the current implementation, each sensor output is first processed in an individual layer to keep sensor specific details. For a computational efficient fusion, cell and grid size of each layer are defined by a factor of an abstract base layer. In addition, the Bayesian logic is used with a logit representation with saturation allowing faster processing of updates. Currently, the sonar sensors and stereo camera are fused and binarized to provide a single representation. Fig. 12 shows a visualization of a reconstructed map of a parking scene using color intensities for obstacle and free space probabilities. For dynamic objects the perception framework includes a separate tracking and object fusion module. For objects in the scene classified as dynamic, obstacles in a sensor measurement associated to these objects are not processed by the grid map update logic which results in a consistent map of static obstacles and list of dynamic objects. Finally both scene views—grid map and object list—are provided to the mission control and path planning modules.

VI. Path Planning and Motion Control

Path and motion planning is split in a hierarchical approach. A mission planner (Sec. VI-A) produces a sequence of tasks through a graph-search on the topological Road-Graph. It assigns these tasks to specific task processors. The current implementation consists of specific processors for on-lane driving (Sec. VI-B) and parking maneuvers (Section VI-C). Trajectories from the task processors are sent to a motion control module (Sec. VI-D).

A. Global Path Planning

A fundamental requirement for vehicle navigation is a global driveable path on the parking lot from the drop-off
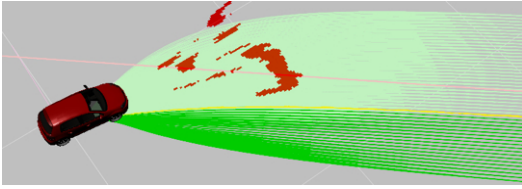
Fig. 13. Set of trajectories produced by the reactive planner. Collision-free trajectories in dark green are rated by their lateral offset to the reference path. The closest collision-free trajectory in yellow is forwarded to the motion controller.

zone to the target pose. The global path planning routine does not exploit any dynamic obstacles but relies only on the static information stored in the RoadGraph. The algorithm employed for global planning exhibits three stages.

The first stage is a traditional A* search along the edges of the RoadGraph. The result is a sequence of edges the vehicle is supposed to pass during navigation. The edges of the RoadGraph are not necessarily driveable. Instead the graph may be only a very rough and sparse approximation of the traffic infrastructure.

The second stage performs an edge smoothing operation using a fourth-degree polar-polynomial function with continuous curvature as proposed by Nelson [23]. The path is checked for its curvature maximum and if a threshold is exceeded a second smoothing step is triggered. This secondary smoothing involves a conjugate gradient descent where the error function is closely related to the one presented by Dolgov et al. [24]. We combine both smoothing stages in order to stabilize the final path output but in many standard scenarios the application of either the first stage or the second stage is sufficient to compute driveable paths.

B. Local on-lane Planning in Dynamic Environments

The local reactive planning stage has to safely navigate the car on the path obtained by the topological planning on the RoadGraph. The reactive planning layer has to generate feasible trajectories for the underlying trajectory controller. These trajectories have to maneuver around newly sensed obstacles and have to be compliant with the kinodynamic constraints of the platform.

We follow the idea presented in [25] and prefer trajectories that are aligned with the reference path. This criterion reduces situation in which we are unable to return to the reference path because of infeasible heading offsets. It also helps the visual localization system to detect the same features seen in the offline mapping process. The reactive planner is implemented in a receding horizon manner. Unlike in [25], we directly include nonholonomic constraints of our platform—which have a major influence at low speeds— in the design process of our motions. A dense set of trajectories is created by applying discrete Euler-integration of the unicycle motion equations in conjunction with the nonlinear feedback controller presented in [26]. The shape of the trajectories is varied by applying a set of longitudinal velocity profiles and lateral offsets as shown in Fig. 13.

To test a trajectory for collision, we apply a fast distance transform on the binary occupancy grid. The rectangular

shape of our vehicle is approximated by a set of circles which can be tested efficiently for collision by a single look-up per circle on the distance map. Collision-free trajectories are rated by their lateral offset to the reference path at their end points. The offset is calculated with a fast distance transform of the reference path.

Predictions of dynamic objects will be included in this reactive planning approach in a later stage of the project.

C. Automated Parking

The automated parking module is activated by the mission executive when the vehicle approaches the target parking spot or the target charging station. It provides a collision-free trajectory to reach the intended position. The trajectories are evaluated if they cause collision with newly detected objects and renewed if necessary until the vehicle reaches the target pose.
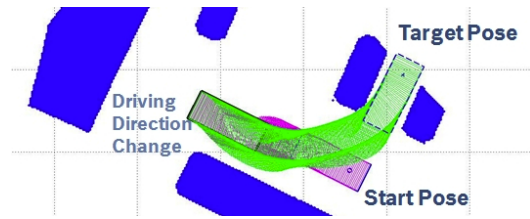

Fig. 14. An example of forward (red) and backward (green) trajectories while parking backwards

The path planning algorithm is a *State Lattice Planner* [27] which executes the A^* path search on the discretized state space. In order to reduce the processing time, the motion primitives, the driving swath (grid point list passed over by the vehicle), and a heuristic look-up table [28] are precalculated. An example of the trajectories provided by the automated parking module is shown in Fig. 14.

As the vehicle needs to park in a charging station with high accuracy and also needs to react to the newly detected objects or any dynamic objects, higher accuracy at the target pose and shorter processing time are required. The possible countermeasures are the multi fidelity state space [29] or a geometric path planning algorithm focusing on the parking maneuver. These possibilities will be analyzed as the next step.

D. Motion Control

The control parameters for trajectory execution are steering angle and acceleration. To this end, the electric power steering (EPS) interface as well as the ACC interface are employed. The input of the controller is a local reference trajectory. Lateral offset control is performed as follows: given a certain point on the reference trajectory, electric field lines of an electric dipole are simulated guiding the vehicle back to the reference trajectory. The active steering angle results from computing the difference of the vehicle orientation and the direction of the current field line. An example is depicted in Fig. 15.

The active desired velocity is given by considering the the maximal velocity on the parking lot and some physical
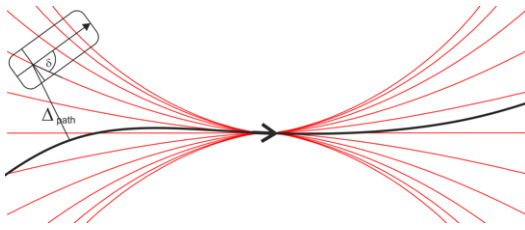
Fig. 15. The red lines indicate the simulated electric field lines. δ reflects the difference angle between the vehicle heading and the slope of the field line. Δ_{path} is the current distance to the reference trajectory.

restrictions. A traditional P-controller is applied to control the vehicle velocity via the ACC interface.

VII. CONCLUSIONS

After one year of development, the V-Charge consortium is confident that all the bricks required for the successful implementation of an automated valet-charging solution have been laid out. Relying on a reliable and capable hardware platform, the project partners have shown that vision-only localization, mapping, navigation, and control of an automated car is possible.

Obviously, the project is just at its beginning and an intense research effort is currently underway regarding perception, situational awareness, localization, environment representation, and planning among dynamic obstacles.

Additionally to these research objectives, the project will keep a strong focus on deploying its system and evaluating it in realistic environments and scenarios with the final goal of fully automated driving in urban environments using only close-to-market sensors.

REFERENCES

[1] "Europe 2020 targets," http://ec.europa.eu/europe2020/europe-2020-in-a-nutshell/targets/index_en.htm, accessed: 2012/12/18.

[2] E. C. for Automotive R&D, "The electrification of the vehicle and the urban transport system," http://www.eucar.be/publications, accessed: 2012/12/18.

[3] C. Urmson and W. Whittaker, "Self-driving cars and the urban challenge," *Intelligent Systems, IEEE*, vol. 23, no. 2, pp. 66–68, 2008.

[4] G. Seetharaman, A. Lakhotia, and E. Blasch, "Unmanned vehicles come of age: The darpa grand challenge," *Computer*, vol. 39, no. 12, pp. 26 –29, dec. 2006.

[5] S. Kammel, J. Ziegler, B. Pitzer, M. Werling, T. Gindele, D. Jagzent, J. Schröder, M. Thuy, M. Goebl, F. Hundelshausen, *et al.*, "Team annieway's autonomous system for the 2007 darpa urban challenge," *Journal of Field Robotics*, vol. 25, no. 9, pp. 615–639, 2008.

[6] M. Montemerlo, J. Becker, S. Bhat, H. Dahlkamp, D. Dolgov, S. Ettinger, D. Haehnel, T. Hilden, G. Hoffmann, B. Huhnke, *et al.*, "Junior: The stanford entry in the urban challenge," *Journal of Field Robotics*, vol. 25, no. 9, pp. 569–597, 2008.

[7] C. Urmson, J. Anhalt, D. Bagnell, C. Baker, R. Bittner, M. Clark, J. Dolan, D. Duggins, T. Galatali, C. Geyer, *et al.*, "Autonomous driving in urban environments: Boss and the urban challenge," *Journal of Field Robotics*, vol. 25, no. 8, pp. 425–466, 2008.

[8] M. Schulze, G. Nocker, and K. Bohm, "Prevent: A european program to improve active safety," in *Proc. of 5th International Conference on Intelligent Transportation Systems Telecommunications, France*, 2005.

[9] A. Furda and L. Vlacic, "Enabling safe autonomous driving in real-world city traffic using multiple criteria decision making," *Intelligent Transportation Systems Magazine, IEEE*, vol. 3, no. 1, pp. 4–17, 2011.

[10] J. Timpner and L. Wolf, "A back-end system for an autonomous parking and charging system for electric vehicles," in *2012 IEEE International Electric Vehicle Conference (2012 IEVC)*, Greenville, South Carolina, USA, Mar. 2012.

[11] ——, "Efficient charging station scheduling for an autonomous parking and charging system (poster)," in *Proceedings of the ninth ACM international workshop on Vehicular inter-networking, systems, and applications*, ser. VANET '12. New York, NY, USA: ACM, 2012, pp. 145–148.

[12] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool, "Speeded-up robust features (surf)," *Comput. Vis. Image Underst.*, vol. 110, no. 3, pp. 346–359, June 2008.

[13] R. I. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, 2nd ed. Cambridge University Press, ISBN: 0521540518, 2004.

[14] M. Pollefeys, D. Nistér, J. M. Frahm, A. Akbarzadeh, P. Mordohai, B. Clipp, C. Engels, D. Gallup, S. J. Kim, P. Merrell, C. Salmi, S. Sinha, B. Talton, L. Wang, Q. Yang, H. Stewénius, R. Yang, G. Welch, and H. Towles, "Detailed real-time urban 3d reconstruction from video," *Int. J. Comput. Vision*, vol. 78, no. 2-3, 2008.

[15] C. Häne, C. Zach, J. Lim, A. Ranganathan, and M. Pollefeys, "Stereo depth map fusion for robot navigation," in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, 2011.

[16] K. Homeier and J. Knaup, "Roadgraph - graph based environmental modelling and function independent situation analysis for driver assistance systems," in *13th International IEEE Conference on Intelligent Transportation Systems*, 9 2010, pp. 428–432.

[17] K. Homeier and L. Wolf, "Roadgraph - high level sensor data fusion between objects and street network," in *14th International IEEE Conference on Intelligent Transportation Systems*, 10 2011.

[18] L. Spinello, R. Triebel, D. Vasquez, K. Arras, and R. Siegwart, "Exploiting repetitive object patterns for model compression and completion," in *European Conference on Computer Vision (ECCV)*, 2010.

[19] R. Paul, R. Triebel, D. Rus, and P. Newman, "Semantic categorization of outdoor scenes with uncertainty estimates using multi-class Gaussian process classification," in *Proc. of the International Conference on Intelligent Robots and Systems (IROS)*, 2012.

[20] R. Bolles and M. Fischler, "A ransac-based approach to model fitting and its application to finding cylinders in range data," in *Proceedings Seventh International Joint Conference on Artificial Intelligence*, 1981, pp. 637–643.

[21] S. Kneip, Furgale, "Using multi-camera systems in robotics: Efficient solutions to the npnp problem," 2013, iEEE Conference on Robotics and Automation 2013 Conference (submitted).

[22] H. Moravec, "Sensor fusion in certainty grids for mobile robots," *AI magazine*, vol. 9, no. 2, p. 61, 1988.

[23] W. Nelson, "Continuous-curvature paths for autonomous vehicles," in *Robotics and Automation, 1989. Proceedings., 1989 IEEE International Conference on*, may 1989, pp. 1260 –1264 vol.3.

[24] D. Dolgov, S. Thrun, M. Montemerlo, and J. Diebel, "Practical search techniques in path planning for autonomous driving," in *Proceedings of the First International Symposium on Search Techniques in Artificial Intelligence and Robotics (STAIR-08)*. Chicago, USA: AAAI, June 2008.

[25] M. Werling, S. Kammel, J. Ziegler, and L. Groll, "Optimal trajectories for time-critical street scenarios using discretized terminal manifolds," *The International Journal of Robotics Research*, vol. 31, no. 3, pp. 346–359, Dec. 2011.

[26] C. Samson and K. Ait-Abderrahim, "Feedback control of a nonholonomic wheeled cart in cartesian space," in *Robotics and Automation, 1991. Proceedings., 1991 IEEE International Conference on*, apr 1991, pp. 1136 –1141 vol.2.

[27] M. Pivtoraiko, R. R. A. Knepper, and A. Kelly, "Differentially constrained mobile robot motion planning in state lattices," *Journal of Field Robotics*, vol. 26, no. 3, pp. 308–333, 2009.

[28] R. Knepper and A. Kelly, "High Performance State Lattice Planning Using Heuristic Look-Up Tables," in *Proceedings of the IEEE-RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2006.

[29] M. Pivtoraiko and A. Kelly, "Differentially constrained motion replanning using state lattices with graduated fidelity," in *Intelligent Robots and Systems, 2008. IROS 2008. IEEE/RSJ International Conference on*. IEEE, 2008, pp. 2611–2616.