

# Lidar-based lane marker detection and mapping

Sören Kammel

Benjamin Pitzer

**Abstract**—The detection of lane markers is a prerequisite for many driver assistance systems as well as for autonomous vehicles. In this paper, the lane marker detection approach that was developed by Team AnnieWAY for the DARPA Urban Challenge 2007 is described. Based on current sensor technology, a robust real-time lane marker detection was developed and implemented. The system allows the robust estimation of a deviations between a digital map and the real world.

## I. INTRODUCTION

Autonomous driving requires substantiated knowledge in different domains like engineering, computer and cognitive sciences: A robust vehicle architecture and design plays a central role. On-board sensor technology, sensor data analysis including localization and sensor fusion techniques, are required for a consistent perception of the environment and the ego pose of the vehicle therein. Omnipresent measurement uncertainties as well as contradictory sensor data have to be handled consistently. This broad variety of tasks shows that the Urban Challenge connects interdisciplinary areas of research with relevance to science, industry and community.

Lane markers are important land marks for the orientation of humans and autonomous vehicles in traffic situations. They separate roads from the non-driveable environment and partly determine driving rules. In this paper, the lane marker detection approach that was developed by Team AnnieWAY for the DARPA Urban Challenge 2007 is described. The purpose of the algorithm originally was a robust estimation and correction of an offset between the map (RNDF = Road Network Definition File) and the real lane the autonomous vehicle was traveling on. It showed that not only the current lane could be detected robustly but also markings of different lanes are recognized. This makes it possible to create a topological map of the traversed street sections. Recent advances in sensor technologies enable simple new algorithms for a robust object/ground classification and especially lane marker detection. Using the reflectivity information recorded by modern laser scanners allows the detection of lane markers in the presence of shadows, against direct sunlight and even at night. In addition, those lidars consist of up to 64 laser beams scanning the environment simultaneously. This simplifies the detection of the road surface as described in Sec. IV. The possibility of registering sensor data of subsequent scans globally and with a high precision using a GPS/INS combination further improves the robustness of the lane marker detection.

This paper is organized as follows: In the next two sections, a brief overview of the hardware and the software architecture of the car developed by Team AnnieWAY for the

Urban Challenge is described. Sec. IV explains the mapping of the environment which is the base for the road surface detection used as a preprocessing step of the lane marker detection. Subsequently, the tracking of moving obstacles is described which are treated separately from the static map of the environment. The lane marker detection itself is described in Sec. VI.

## II. HARDWARE ARCHITECTURE

The basis of the AnnieWAY automobile is a 2006 VW Passat Variant B4 (Figure 1). The Passat has been selected for its ability to be easily updated for drive-by-wire use by the manufacturer.

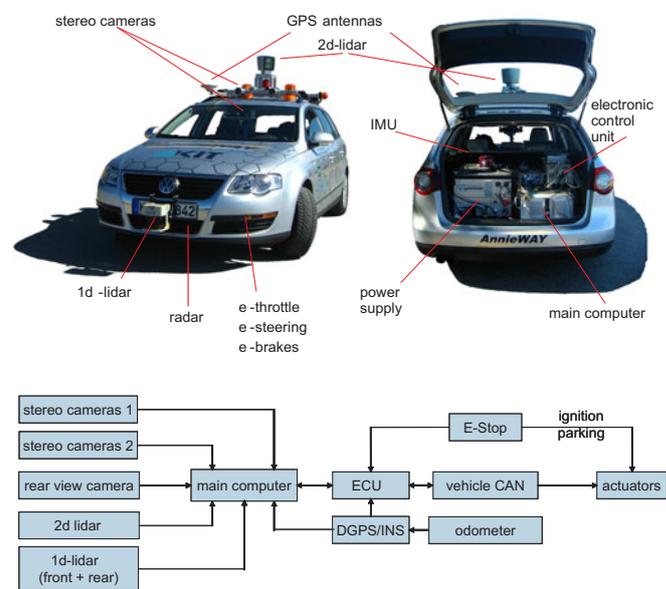


Fig. 1. Architecture and hardware components of the vehicle.

AnnieWAY relies on an off-the-shelf AMD dual-core Opteron multiprocessor PC main computer whose computing power is comparable to a small cluster, yet offers low latencies and high bandwidth for interprocess communication. All sensors connect directly to the main computer which offers enough processing capacity to run almost all software components. The main computer is augmented by a dSpace AutoBox that operates as electronic control unit (ECU) for low-level control algorithms. It directly drives the vehicle's actuators. Both computer systems communicate over a 1 Gbit/s Ethernet network. The drive by wire system as well as the car odometry are interfaced via the Controller Area Network (CAN) bus. The DGPS/INS system allows for precise localization and connects to the main computer

and to the low-level ECU (AutoBox). The chosen hardware architecture is supported by a real-time-capable software architecture described in the next section [1].

Since lidar units are active sensors and therefore produce their own light, low light conditions have no effect on this kind of sensor. The Velodyne HDL-64E lidar uses 64 fixed lasers to cover a 26.5-degree vertical field of view. The lasers are mounted on a spinning platform that rotates at a rate of 600 rpm. Therefore it provides a 360 degree field of view around the vehicle producing more than 1 million points per second at an angular resolution of 0.09 degrees horizontally and a distance resolution of 5cm with distances up to 120m. The result is highly accurate representation of almost the entire scene surrounding the vehicle. For each point, the sensor measures range and reflectivity. The reflectivity map is well suited for monoscopic image analysis tasks like lane marker detection. The inherent association of each reflectivity pixel with a range measurement alleviates information fusion of these data significantly.

For precise localization we use an OXTS RT3003 Inertial and GPS Navigation System which is an advanced six-axis inertial navigation system that incorporates a Novatel L1/L2 RTKGPS receiver for position and a second GPS Receiver for accurate Heading measurements. Odometry is taken directly from AnnieWAY's wheel encoders. The RT3003 delivers better than 0.02m positioning accuracy under dynamic conditions using differential corrections and 0.1° heading accuracy using a 2m separation between the GPS antennas. The RT3003 Inertial and GPS Navigation System includes three angular rate sensors (gyros), three servo-grade accelerometers, the GPS receiver and the required processing. It works as a standalone, autonomous unit and requires no user input for operation.

### III. SOFTWARE ARCHITECTURE

The core components of the vehicle are the perception of the environment, an interpretation of the situation in order to select the appropriate behavior, a path planning component and an interface to the vehicle control. Figure 2 depicts a block diagram of the information flow in the autonomous system. Spatial information from the lidar and the stereo cameras are combined to a static 3D map of the environment. Moving objects are treated differently. Such dynamic objects also include traffic participants that are able to move but have zero velocity at the moment. To detect moving objects, the spatial measurements of the lidar sensor are clustered and tracked with a multi-hypothesis approach. To detect possibly moving objects (which are standing still right now), a simple form of reasoning is used: If an object has the size of a car and is located on a detected lane, it is considered to be probably moving. Lane markers are detected in the reflectance data of the main lidar. Together with the road network map (RNDF), the absolute position obtained from the DGPS/INS system and the mission plan (MDF), this information serves as input for the situation assessment and the subsequent behavior generation. Most of the time, the behavior will result in a driveable trajectory. If

a road is blocked or the car has to be parked, modules for special maneuvers, like the parking lot navigation module, are activated.

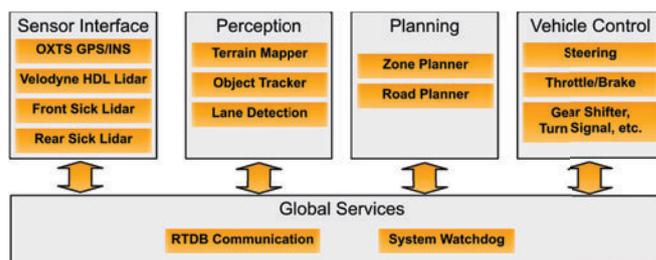


Fig. 2. Overview of the software architecture and the information flow.

### IV. ENVIRONMENTAL MAPPING



Fig. 3. Example for the evidence mapping of 3D lidar data onto a 2D grid. Darker spots correspond to high evidence for an obstacle while white cells correspond to drivable area. Unknown cells are marked as grey.

Accurate and robust detection of obstacles at a sufficient range is an essential prerequisite to avoid obstacles on the road and in unstructured environments like parking lots.

The basic idea is to maintain an evenly spaced 2D grid structure  $g$  where each cell  $g_i$  represents a random variable. Each random variable is binary and corresponds to the occupancy it covers. Therefore in the literature this approach is also called *Occupancy Grid Mapping* ([2], [3]) which has the ultimate goal to calculate the posterior over maps  $p(g|z, x)$  where  $z$  is the set of all measurements and  $x$  is the path of the vehicle defined through a sequence of poses.

AnnieWAY uses a grid that is always centered at the vehicle position but aligned with a global coordinate system. The grid is shifted at each timestep to account for the new vehicle position. This restricts the size of the map to an area around the vehicle while the cells are bound to an absolute position. The size of each grid cell is 15cm×15cm. Figure 3 shows an example for our mapping algorithm.

AnnieWAY is equipped with two different types of sensors. Here, we will mainly discuss the integration of high resolution lidar data. Algorithms for the integration of low resolution lidar data can be found in [2], [3], [4], [5].

Processing the data of the HDL-64E lidar into an environmental map consists of three steps. In the first step the range measurements  $z_l$  of one revolution  $L$  are projected into a global coordinate system under consideration of the vehicle's motion  $x_l$ . In the second step different measures are extracted from the data for each cell  $g_i$ . Two straightforward measures are the number of measurements  $n_i$  and the number of different laser beams  $b_i$ . The most important measure we use is the elevation difference

$$e_i(g_i, z_l) = \max_l h(g_i, z_l) - \min_l h(g_i, z_l), \quad (1)$$

where  $h$  is the vertical component of each measurement.

In the third step we compute the evidence for each measure by using an inverse sensor model. E.g. the inverse sensor model for the elevation difference returns  $l_{occ}$  if  $e_i$  exceeds a threshold (e.g. 15cm) and  $l_{free}$  otherwise. The inverse models for  $n_i$  and  $b_i$  are slightly more complex since they are learned by a supervised learning algorithm. The result of the learning procedure is a forward model which accepts  $g_i$  and  $n_i$  or  $b_i$  respectively as parameters and returns the appropriate evidence.

Finally we can compute the combined occupancy evidence  $o_{i,t}$  as a weighted sum of the partial evidences

$$o_{i,t} = o_{i,t-1} + \alpha_1 \cdot n_i + \alpha_2 \cdot b_i + \alpha_3 \cdot e_i, \quad (2)$$

with

$$1 = \alpha_1 + \alpha_2 + \alpha_3, \quad (3)$$

and the estimated occupancy for a single cell

$$p(g_i | \mathbf{z}, \mathbf{x}) = 1 - \frac{1}{1 + \exp o_i} \quad (4)$$

AnnieWAY is equipped with different sensors and ideally one wants to integrate information from all sensors into a single map. A naive solution is to update the map for each sensor separately which neglects the different characteristics of each sensor, E.g. field of view, maximal range and noise characteristic. To ensure safe driving we use the most pessimistic approach to fuse the sensors. We compute the maximum

$$p(g_i) = \max_k p(g_i^k) \quad (5)$$

of all estimated occupancies, where  $K$  is the number of sensors. If any sensor will detect a cell as occupied, so will the combined map.

The standard *Occupancy Grid Mapping algorithm* suffers from one major drawback: it is only suitable for static environments. Driving environments are typically highly dynamic and the result is very poor without modifications. Moving objects create virtual obstacles with high evidence while

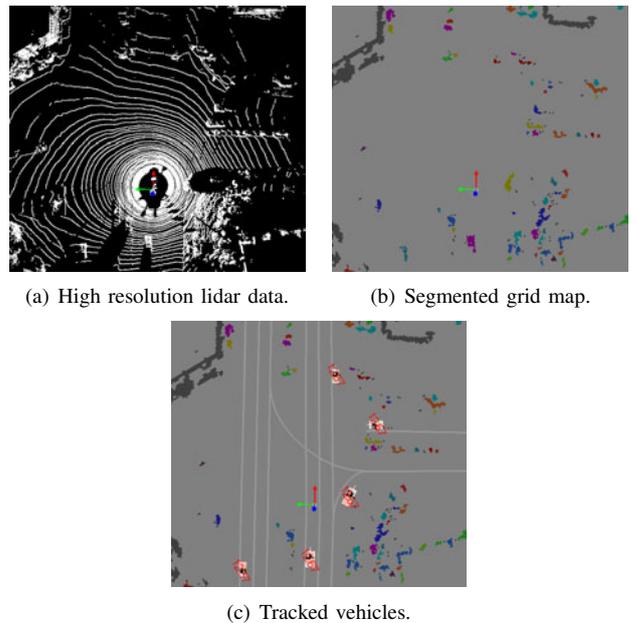


Fig. 4. Tracking of dynamic objects with occupancy grid map and linear Kalman Filter.

moving. To overcome this problem we introduce a temporal evidence decay. The evidence is reduced at each timestep by a factor  $\epsilon_t$  for cells which are not updated. The intuition is that the uncertainty increases for cells not augmented by any sensor. Equation 2 turns now into

$$o_{i,t} = \operatorname{argmax}(0, o_{i,t-1} + \alpha_1 \cdot n_i + \alpha_2 \cdot b_i + \alpha_3 \cdot e_i - \epsilon_t), \quad (6)$$

where the *argmax* operator enforces positive evidences.

## V. TRACKING OF DYNAMIC OBJECTS

Driving in urban environments requires to capture and estimate the dynamics of other traffic participants in real time. AnnieWAY uses a processing pipeline that takes in raw sensor data (from different lasers) and generates a list of dynamic obstacles, along with their estimated locations, sizes, and relative velocities, that accurately describes the relevant dynamic environment surrounding. This pipeline consists of a number of parts, including

- 1) **Data preprocessing:** removing irrelevant readings: noise, ground readings, readings from obstacles outside the road, etc.
- 2) **Obstacle detection:** creating a list of obstacles raw readings...includes segmentation for laser
- 3) **Obstacle tracking:** corresponding obstacles timestep with those of another timestep in order determine their headings, relative velocities, etc.
- 4) **Obstacle postprocessing and publishing**

The data preprocessing step used for tracking was discussed earlier as part of the environmental section. The result of this part is a grid map with occupancy probabilities attached to each cell.

The first stage of dynamic object tracking is the identification of object hypothesis. AnnieWAY uses occupancy grid

maps which are segmented using a connected components approach. We treat each grid cell as a node in a graph  $G$ . Two points are connected if and only if the distance between them is within a threshold  $d$  (e.g. 0.5m). We then find all the connected components in the graph and assign cells that are in the same connected component to the same group. To filter out noise, we may discard any connected component with less than a minimum number of cells. Figure 4(b) depicts the result of the connected components segmentation.

The connected components are analysed in a second step for their probability of being other traffic participants. Here we use several heuristics based on their shape and location relative to the road network. Only 'good' candidates are augmented in the following tracking step. Figure 4(c) displays the resulting objects after postprocessing.

With this procedure we still detected far more objects than we wanted to track and often tracked more objects than we wanted to publish to other modules. This is due to noisy observations, occlusion, dynamic objects passing out of our sensors fields of view, etc. In order to decide when to track and publish obstacles, we define a notion of confidence that works similarly to log-likelihood updates in an occupancy grid map as mentioned earlier. When an obstacle is observed, we increment its confidence, and it goes unobserved in our field of view, we decrement it. Thus defined, confidence allows us to set minimum thresholds for the tracking and publishing obstacles.

Tracking of dynamic objects serves two purposes. First, it aids the correspondence of obstacles detected in one sensor frame with those in subsequent sensor frames. This can be done simply with distance-based methods or more sophisticated 3D fitting and registration methods like iterative closest point (ICP). However, these methods don't take into account the noise and uncertainty of our sensors. The second and equally important purpose of tracking is to return estimates of other vehicles relative velocities and headings.

AnnieWAY uses a *Linear Kalman Filter* [6] to model a simplified dynamic obstacle state as  $[x \ y \ \Delta x \ \Delta y]^T$ . Obviously this model ignores completely the nonlinear dynamics of cars, but the frequency of sensor updates (10Hz) means that cars move very little between them which allows us to assume linear dynamics. Transition updates are linear with a linear covariance

$$T = \begin{bmatrix} 1 & 0 & \Delta t & 0 \\ 0 & 1 & 0 & \Delta t \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, Q = \begin{bmatrix} \sigma_{q_x}^2 & 0 & 0 & 0 \\ 0 & \sigma_{q_y}^2 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad (7)$$

and we use a simplified observation model that is also linear, in which we do not directly observe the velocities ( $\Delta x$  and  $\Delta y$ )

$$O = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}, R = \begin{bmatrix} \sigma_{r_x}^2 & 0 \\ 0 & \sigma_{r_y}^2 \end{bmatrix} \quad (8)$$

We do not know a priori which observed obstacles are in fact observations of which tracked obstacles or whether they are observations of previously unseen obstacles. Thus, we are required to solve a problem of correspondence between observations and dynamic obstacles. This is a nontrivial problem, requiring that we define both a measure of distance and a procedure for finding the optimal correspondence. AnnieWAY uses a maximum-likelihood matching algorithm to find the optimal assignment of observations to existing kalman filters. This matching is a one-to-one function from filters to observations.

## VI. LANE MARKER DETECTION

In the context of this paper, lane markers can be either painted markings or curbs. Painted lane markings are detected from the intensity readings of the lidar whereas curbs cause small height changes in the range data of the lidar. A combined intensity/range plot is depicted in the Fig. 7(a). Both kind of lane markers form one dimensional structures that can be approximated by line segments locally. In contrast to camera images, the laser reflectivity and range data is insensitive to background light and shadows while producing only a sparse images. In order to increase the density of the lane marker information, subsequent scans are registered and accumulated employing GPS/INS information. The first step in order to obtain a dense map of lane marker features is a classification of the data points in each scan into obstacle and ground according to the algorithms described in the last sections. Lane markers are expected to occur on the road surface (painted markers) or at their borders (curbs) only. Therefore, the points of each individual laser labeled as ground are searched for large continuous chunks (chunks that do not exhibit height changes exceeding the height of curbs) representing the road. Only in those large chunks high intensity gradients are detected. In addition, only points that exhibit an absolute intensity larger than the median intensity of each laser scan are taken into account. Both types of features –painted markings and curbs– are mapped into a feature grid  $g(\mathbf{x})$  similar to the evidence grid described in Sec. IV, see Fig. 6(a) and Fig. 6(c). Features are detected first in the single scans and mapped afterwards (instead of creating a dense map first and extracting the features then) to minimize the effect of errors in the vehicle localization. A summary of the detection algorithm is shown in Fig. 5.

Lane segments are detected by applying the Radon transform

$$g(\theta, r)_{RT} = \mathcal{R} g(\mathbf{x}) = \int \int g(\mathbf{x}) \delta(\mathbf{x}^T \mathbf{e}(\theta, r)) d\mathbf{x} \quad (9)$$

with

$$\mathbf{e}(\theta, r) = (\cos \theta, \sin \theta)^T \quad (10)$$

to the accumulated feature map data (see Fig. 6(b) and Fig. 6(d)). Since the Radon transform is a global algorithm (regarding the map) it proved to be robust against occlusions, noise and outliers. Compared to the Hough transform, the

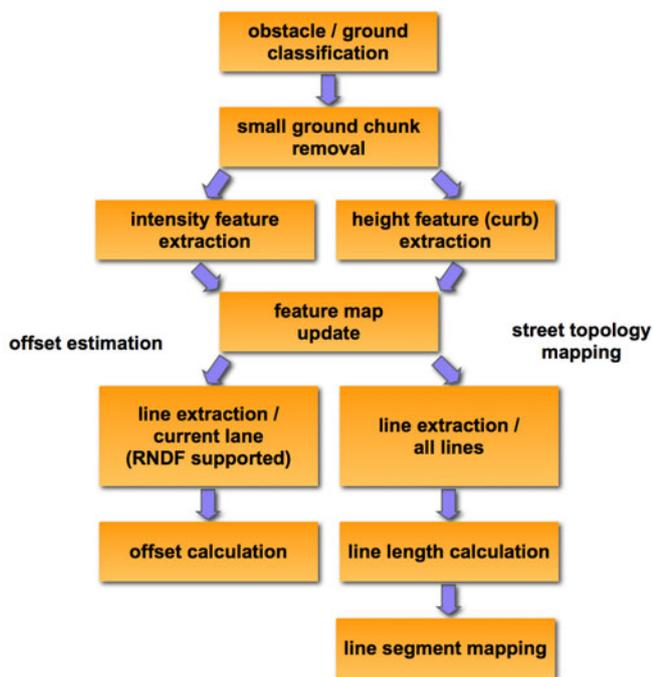


Fig. 5. Overview of the offset estimation and street topology mapping

Radon transforms exhibits the advantage of a calculation time independent of the numbers of lane markers and the capability to handle gray-scale images efficiently and without thresholding. For a real-time calculation in the car, an implementation exploiting the central-slice theorem was used[7]. The position and direction of lane boundaries can be calculated by locating their corresponding maxima in the Radon plane. To determine deviations of a digital map from the real street network, the lane markers of the digital map are first projected into the Radon plane. Assuming that the offset between map and reality does not exceed one lane width, the deviation is obtained in a second step from the distances to the maxima in the Radon plane closest to the predicted positions. Assuming further that predicted and estimated lane boundaries are close to parallel, the vertical distance is sufficient to determine the offset.

While Fig. 7 illustrates that the algorithm works well in real environments, it also clearly reveals the significant offset between the road network provided and the visible lane markers.

Automatic generation of street maps is a further application of the lane marker detection. An important step on the way to assess traffic situations is a proper detection and mapping of lanes and intersections. For this purpose, the lane marker detection and offset correction system is extended to handle all visible lane markers and not only the ones belonging to the current lane. Therefore, all major peaks in the Radon plane have to be detected. This is achieved by first filtering the Radon image with a morphological filter (dilatation) and then thresholding the resulting image relative to its median. The maxima in the original Radon image that

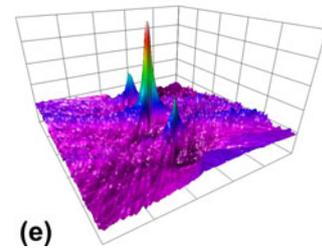
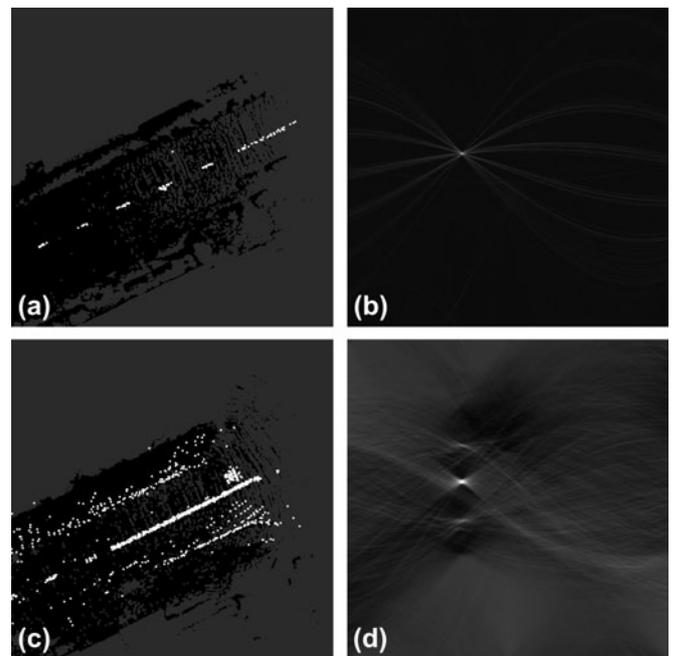


Fig. 6. Lane marker map created on intensity data only (a) with its corresponding Radon transform (b); lane marker map created on intensity data and height data (curbs) (c) with its corresponding Radon transform (d) and a 3D plot of a region of (d) to visualize the robust line estimation of the Radon transform (e).

are inside the corresponding regions in the thresholded image are considered as candidates for lane boundaries. To further refine this result, the number of points contributing to this possible lane boundary are determined. This can be achieved by summing the vertical neighbors in the Radon plane. After this, the remaining lane boundaries are projected back into the real world and associated with their closest neighbors (position and orientation) of the previous time steps.

## VII. CONCLUSIONS

New sensors enable new possibilities for fast and robust object detection, object/ground classification and lane marker detection. In this paper, the lane marker detection approach that was developed by Team AnnieWAY for the DARPA Urban Challenge 2007 is described. Based on a 64 beam lidar, a robust real-time lane marker detection was developed and implemented. The system allows the robust calculation of a possible offset between a digital map (RNDF) and the real street markings working in the presence of shadows, against direct sunlight and even at night. The promising

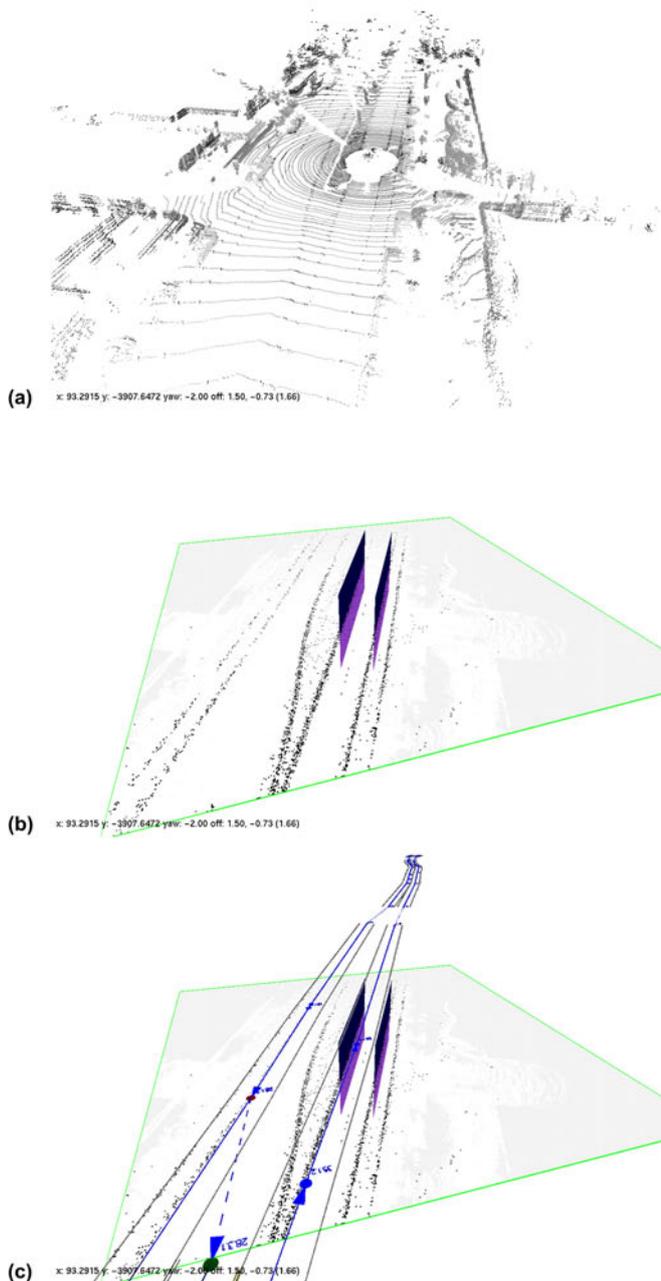


Fig. 7. Intensity readings of the lidar (a), lane marker map and the estimated current lane segment (b), overlay of a part of the original road network map (c).

results will be used in combination with a knowledge base to automatically map the inner city traffic environment.

#### VIII. ACKNOWLEDGEMENTS

The authors gratefully acknowledge the fruitful collaboration of their partners from Universität Karlsruhe, Technische Universität München and Universität der Bundeswehr München. Special thanks are directed to Annie Lien for her instant willingness and dedication as our official team leader. This work had not been possible without the ongoing research of the Transregional Collaborative Research Centre 28

Cognitive Automobiles'. Both projects cross-fertilized each other and revealed significant synergy. The authors gratefully acknowledge support of the TCRC by the Deutsche Forschungsgemeinschaft (German Research Foundation).

#### REFERENCES

- [1] Matthias Goebel and Georg Färber. A real-time-capable hard- and software architecture for joint image and knowledge processing in cognitive automobiles. In *Proc. IEEE Intelligent Vehicles Symposium*, pages 734–739, Istanbul, Turkey, June 2007.
- [2] Sebastian Thrun. Robotic mapping: A survey. In G. Lakemeyer and B. Nebel, editors, *Exploring Artificial Intelligence in the New Millennium*. Morgan Kaufmann, 2002.
- [3] S. Thrun. Learning occupancy grid maps with forward sensor models. *Auton. Robots*, 15(2):111–127, 2003.
- [4] P. Biber and W. Strasser. nscan-matching: Simultaneous matching of multiple scans and application to slam. In *IEEE International Conference on Robotics and Automation*, 2006.
- [5] M. Bosse, P. M. Newman, J. J. Leonard, M. Soika, Wendelin Feiten, and Seth J. Teller. An atlas framework for scalable mapping. In *Proceedings of the 2003 IEEE International Conference on Robotics and Automation*, pages 1899–1906, Taipei, Taiwan, September 14–19 2003.
- [6] R.E. Kalman. A new approach to linear filtering and prediction problems. *Journal of Basic Engineering*, 82(1):35–45, 1960.
- [7] R. N. Bracewell. Numerical transforms. *Science*, 248(4956):697–704, 1990.