IEEE Intelligent Transportation Systems Magazine, vol .6, no. 2, pp. 8-20, 2014

Making Bertha Drive — An Autonomous Journey on a Historic Route

Julius Ziegler, Thao Dang, Uwe Franke, Henning Lategahn, Philipp Bender, Markus Schreiber, Tobias Strauss, Nils Appenrodt, Christoph G. Keller, Eberhard Kaus, Christoph Stiller, Ralf G. Herrtwich, *et al.*

***Abstract*—125 years after Bertha Benz completed the first overland journey in automotive history, the Mercedes Benz S-Class S 500 INTELLIGENT DRIVE followed the same route from Mannheim to Pforzheim, Germany, in fully autonomous manner. The autonomous vehicle was equipped with close-to-production sensor hardware and relied solely on vision and radar sensors in combination with accurate digital maps to obtain a comprehensive understanding of complex traffic situations. The historic Bertha Benz Memorial Route is particularly challenging for autonomous driving. The course taken by the autonomous vehicle had a length of 103 km and covered rural roads, 23 small villages and major cities (e.g. downtown Mannheim and Heidelberg). The route posed a large variety of difficult traffic scenarios including intersections with and without traffic lights, roundabouts, and narrow passages with oncoming traffic. This paper gives an overview of the autonomous vehicle and presents details on vision and radar-based perception, digital road maps and video-based self-localization, as well as motion planning in complex urban scenarios.**

***Index Terms*—Autonomous driving, stereo vision, radar sensing, self-localization, motion planning, digital maps.**

I. INTRODUCTION

IN August 1888, Bertha Benz and her two sons began the first cross-country automobile journey in the world. Without telling her husband Carl Benz, she drove his *Benz Patentmotorwagen Number 3* from Mannheim to Pforzheim, a route with a one-way distance of more than 100 km (approx. 65 miles) through southern Germany. Today, this overland journey is received as a pioneering event in the history of automobiles. Not only did Bertha Benz demonstrate the maturity of Carl Benz's gasoline engine. The public reactions on her maiden voyage paved the ground for her husband's economic success and the acceptance of the automobile in society. 125 years later, a Mercedes Benz S-Class prototype vehicle revisited the Bertha Benz Memorial Route, yet this time in a fully autonomous manner.

et al. are: Clemens Rabe, David Pfeiffer, Frank Lindner, Fridtjof Stein, Friedrich Erbs, Markus Enzweiler, Carsten Knöppel, Jochen Hipp, Martin Haueis, Maximilian Trepte, Carsten Brenk, Andreas Tamke, Muhammad Ghanaat, Markus Braun, Armin Joos, Hans Fritz, Horst Mock, Martin Hein, and Eberhard Zeeb.

J. Ziegler, and P. Bender are with the Forschungszentrum Informatik (FZI), Mobile Perception Systems, 76131 Karlsruhe, Germany, e-mail: {ziegler, pbender}@fzi.de.

H. Lategahn, M. Schreiber, T. Strauß, and C. Stiller are with the Department of Measurement and Control Systems, Karlsruhe Institute of Technology, Germany, e-mail: {henning.lategahn, markus.schreiber, strauss, stiller}@kit.edu

The remaining authors are with Daimler AG, Research & Development, 71059 Sindelfingen, Germany, e-mail: firstname.lastname@daimler.com.

Manuscript received November 19, 2013; revised December XX, XXXX.

The last two decades have seen tremendous advances in autonomous driving and we can only give a non-exhaustive review of some important work here. Early European contributions started within the PROMETHEUS project in the 1990s, cf. [1]–[3]. The probably most renowned autonomous drive of the team was a tour in 1995 from Munich, Germany, to Odense, Denmark, at velocities up to 175 km/h with about 95% autonomous driving. In the U.S. similar research had been conducted. In the 'No hands across America' tour, Pomerleau and Jochem drove from Washington DC to San Diego with 98% automated steering yet manual longitudinal control [4].

Activities in this century have strongly been characterized by several public challenges. The Defense Advanced Research Projects Agency (DARPA) organized a first Grand Challenge for autonomous off-road ground vehicles in March 2004 and a second challenge in October 2005, e.g. [5]. The third DARPA Challenge was held in November 2007. In this *Urban Challenge*, vehicles had to drive through a mock up urban environment on a closed airfield in Victorville, California. Compared to the previous challenges this competition included some interaction with other vehicles, while other features like pedestrians, bicyclists, or traffic lights were still absent. Most successful teams in the DARPA challenges employed high-end laser scanners coupled with radars while computer vision played at most a secondary role (e.g. [6], [7]). High-precision GPS/INS was used for localization. The Grand Cooperative Driving Challenge 2011 (GCDC) was the first international competition to implement highway platooning scenarios of cooperating vehicles connected with communication devices, e.g. [8].

Several teams around the world are continuously advancing the field of autonomous driving. Among the publicly most noticed activities is the impressive work by Google that extends experience gained in the Urban Challenge. A roof-mounted high-end laser scanner and a detailed map, recorded in a prior manual drive, provide the main information about the driving environment. In July 2013, the team around Broggi [9] performed another impressive autonomous driving experiment in public traffic near Parma, Italy. Interesting work that aims for autonomous driving with close to production vehicles is presented in [10].

Compared to previous works on autonomous vehicles, we find the Bertha Benz Memorial Route is unique in difficulty and variability of encountered traffic scenarios. The route comprises overland passages, urban areas as e.g. Mannheim and downtown Heidelberg and 23 small villages, partly with narrow streets (Fig. 1). The autonomous vehicle handled traffic

Fig. 1. The Bertha Benz Memorial Route from Mannheim to Pforzheim (103km). The route comprises rural roads, urban areas (e.g. downtown Heidelberg) and small villages and contains a large variety of different traffic situations as e.g. intersections with and without traffic lights, roundabouts, narrow passages with oncoming vehicles, pedestrian crossings, etc.

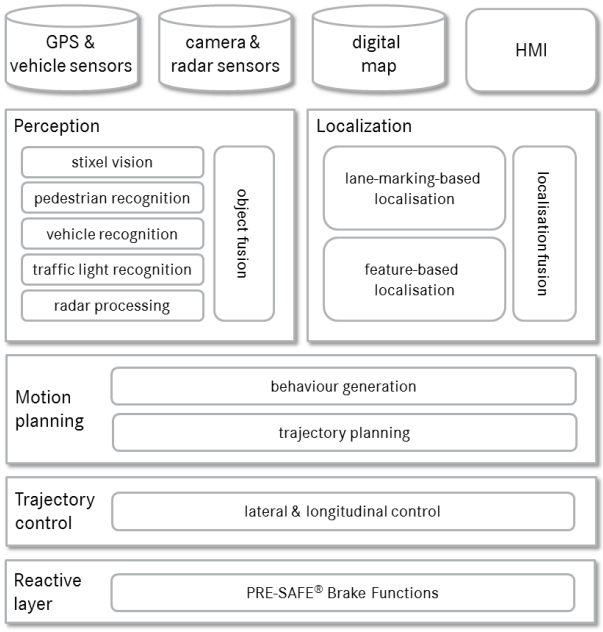


Fig. 2. System overview of the Bertha Benz experimental vehicle.

lights, pedestrian crossings, intersections and roundabouts in real traffic. It had to react on a variety of objects including parked cars, preceding and oncoming vehicles, bicycles, pedestrians and trams. Besides facing the challenges of the historic Bertha Benz Route, our second goal was to realize autonomous driving based on close-to-market sensors. Our robot relies solely on the sensor setup of a standard 2013 S-Class vehicle and additional radar and vision sensors for object detection and free-space analysis, traffic light detection and self-localization.

In the remainder of this paper, we will provide an overview of the experimental vehicle used for the Bertha Benz drive. The system architecture of our robot is outlined in Fig. 2. As stated earlier, the main sensing components are cameras and radar sensors. These will be reviewed in Sec. II. Another important source of information is a detailed digital map (cf. Sec. III). This map contains the position of lanes, the topology between them as well as attributes and relations defining traffic regulations (e.g. right-of-way, relevant traffic lights, and speed limits). An important prerequisite for using such digital maps is a precise map relative localization. In this work, we employ two complementary vision algorithms — point feature based localization and lane marking based localization — to accomplish this task (Sec. IV). The objective of the motion planning modules (cf. Sec. V) is to derive an optimal trajectory, i.e. the path of the vehicle as a function of time, from the given sensor and map information. This trajectory is transformed into actuator commands by respective lateral and longitudinal controllers (Sec. V-C). All standard emergency braking systems available in our Mercedes-Benz S-Class S 500 INTELLIGENT DRIVE are activated in our prototype vehicle and underlie our autonomous driving function, such that emergency braking need not be considered in the trajectory planning and control modules, cf. Secs. V and V-C. A specific human-machine-interface has been designed to inform the operator of the vehicle about current driving maneuvers. For limited space, however, the latter two components are not discussed here. Before concluding the paper, we summarize our experimental results on the Bertha Benz Memorial Route in Sec. VI.

II. PERCEPTION

Precise and comprehensive environment perception is the basis for safe and comfortable autonomous driving in complex traffic situations such as busy cities. As mentioned above, we modified the serial-production sensor setup already available in our S-Class vehicles as follows: Four 120° short-range radars were added for better intersection monitoring and two long range radar mounted to the sides of the vehicle in order to monitor fast traffic at intersections on rural roads. The baseline of the Bertha's existing stereo camera system was enlarged to 35 cm for increased precision and distance coverage. For traffic light recognition and pedestrian recognition in turning maneuvers, an additional wide angle-monocular color camera was mounted on the dash-board. A second wide-angle camera looking backwards was added for self-localization described in Sec. IV. The complete sensor setup is shown in Fig. 3. The main objectives of these sensors are free-space analysis (*Can Bertha drive safely along the planned path?*), obstacle detection (*Are there obstacles in Bertha's path? Are obstacles stationary or moving? How do they move?*), and object classification (*What is the type of obstacles and other traffic participants, e.g. pedestrians, bicyclists, or vehicles?*).

Although various perception systems are already on board for advanced driver assistance, including fully autonomous emergency braking for pedestrians, the existing algorithms had

to be improved significantly. Previous safety relevant assistance systems necessitate a minimum false positive rate while keeping the true positive rate sufficiently high. An autonomous system, however, requires the environment perception module to detect nearly all obstacles and — at the same time — to have an extremely low false-positive rate.

A. Stereo Vision — the Stixel Approach

The stereo camera used to understand the environment in front of the ego-vehicle covers a range of up to 60 m with a $45°$ field of view. The stereo processing pipeline consists of four main steps: the dense stereo reconstruction itself, the computation of super-pixels called stixels, their tracking over time to estimate the motion of each stixel, and the final object segmentation. The different processing steps are briefly illustrated in Fig. 4.

1) Stereo Matching: Given the stereo image pairs, dense disparity images are reconstructed using semi-global matching (SGM) [11], c.f. Fig. 4a and Fig. 4b. A real-time realization of this scheme was made available on an efficient, low-power FPGA-platform by [12]. The input images are processed at 25 Hz with about $400,000$ individual depth measurements per frame.

2) Stixel Computation: To cope with this large amount of data, we developed the so called stixel representation [13], [14]. The idea is to approximate all objects within the three-dimensional environment using sets of thin, vertically oriented rectangles. Each stixel is defined by its position, footpoint and height. All areas of the image that are not covered with stixels are implicitly understood as free, and thus, in intersection with the map of the route, as potentially driveable space. To consider non-planar ground surfaces, the vertical road slope is estimated as well. Altogether, the relevant 3D content of the scene is represented by an average of about 300 stixels only. Just like SGM, the stixel computation is performed on an FPGA platform.

3) Motion Estimation: Autonomously navigating through urban environments asks for detecting and tracking other moving traffic participants, such as cars or bicyclists. In our setup, this is achieved by tracking single stixels over time using Kalman filtering following the 6D-vision approach of [15], as described in [16]. The result of this procedure is given in Fig. 4c showing both the stixel representation and the motion prediction of the stixels.

4) Object Segmentation: Up to this point, stixels are processed independently, both during image segmentation and tracking. Yet, given the working principle of this representation, it is quite likely for adjacent stixels to belong to one and the same physical object. Thus, when stepping forward from the stixel to the object level, the knowledge which stixel belongs to which object is of particular interest, e.g. for collision avoidance and path planning.

For object segmentation, we rely on the approach presented in [17]. Besides demanding motion consistency for all stixels representing the same object, this scheme also makes strong use of spatial and shape constraints. The optimal segmentation is obtained by means of graph cuts that — thanks to the compact representation — runs in less than 1ms on a single CPU. The segmentation result for the depicted scenario is given in Fig. 4d.

B. Vehicle and Pedestrian Recognition

The sketched spatio-temporal analysis is complemented by an appearance based detection and recognition scheme. This approach detects pedestrians up to 40 m in front of Bertha and oncoming vehicles up to 200 m. In doing so, we can exploit class-specific (pedestrian and vehicle) models and increase the robustness of the visual perception significantly.

Our real-time recognition system consists of three main modules: region-of-interest (ROI) generation, object classification and tracking. All system modules make use of two orthogonal image modalities extracted from stereo vision, i.e. gray-level image intensity and dense stereo disparity. This processing chain is described for the recognition of pedestrians as an example.

1) ROI Generation: The 3D road profile obtained from dense stereo vision constraints possible pedestrian locations regarding the estimated ground plane location, 3D position and height above ground. Regions of Interest (ROIs) are then computed in a sliding-window fashion at corresponding scales.

2) Classification: Each ROI is classified by means of a powerful multi-cue classifier. Here, we are using a Mixture-of-Experts scheme that operates on a diverse set of image features and modalities as described in [18]. In particular, we couple gradient-based features such as histograms of oriented gradients (HoG) [19] with texture-based features such as local binary patterns (LBP) or local receptive fields (LRF) [20]. Furthermore, all features operate both on gray-level intensity as well as dense disparity images to fully exploit the orthogonal characteristics of both modalities [18]. Classification is done using linear support vector machines. Multiple classifier responses at similar locations and scales are addressed by applying mean-shift-based non-maximum suppression to the individual detections, e.g. a variant of [21]. For classifier training, we use the public *Daimler Multi-Cue Pedestrian Classification Benchmark*, as introduced in [22].

3) Tracking: For tracking, we employ an Extended Kalman Filter (EKF) with an underlying constant velocity model of dynamics. As such, the state vector holds lateral and longitudinal position as well as corresponding velocities. Measurements are derived from the footpoint of detected pedestrians and the corresponding depth measurements from stereo vision.

Pedestrians in areas to the side of the vehicle are particularly relevant in turning maneuvers. Given our limited field-of-view in the stereo system, we additionally utilize a monocular variant of the pedestrian system described above, operating on the wide angle camera that is also used for traffic light recognition.

Vision-based vehicle detection follows a similar scheme except for the ROI generation. In the near-range (up to 40m), ROIs are found using the Stixel World as described in [23]. For higher distances up to 200m, stereo-based ROI generation cannot be applied. Thus, we rely on a fast monocular vehicle detector to create search regions for our subsequent Mixture-of-Experts classifiers as described above.

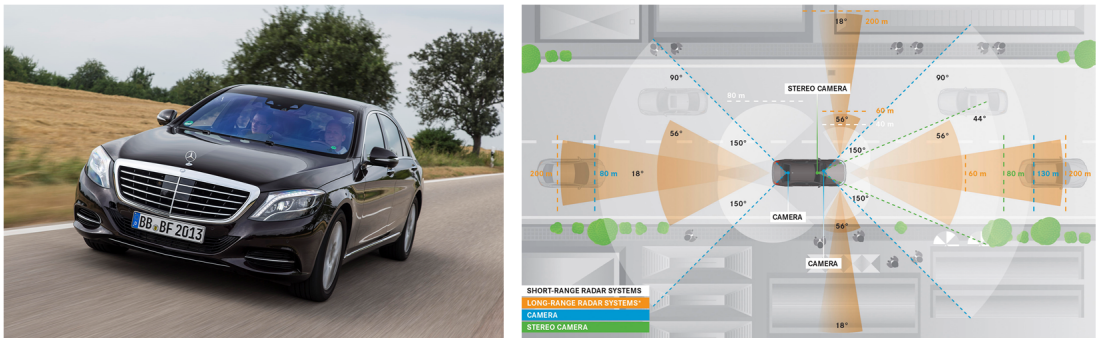
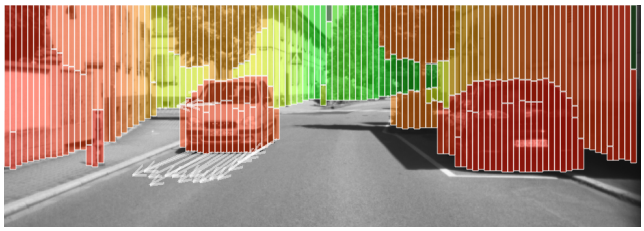


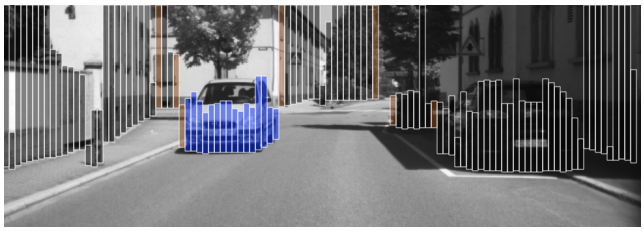
Fig. 3. The Bertha Benz experimental vehicle and it's sensors. Depicted in orange are the sensing fields of the long and mid range radar sensors. Marked in blue are range and field of view of the used wide angle cameras. The central stereo vision system is shown in green.

(a) Left input image of the stereo camera setup. The ego-vehicle drives through a narrow urban environment with static infrastructure (buildings, trees, poles), a parking car on the right as well as an approaching vehicle.

(b) Visualization of the SGM stereo matching result. Red pixels are measured as close to the ego-vehicle (i.e. $dist \leq 10$ m) while green pixels are far away (i.e. $dist \geq 60$ m).



(c) Stixel World representation of the disparity input. Objects are efficiently described using vertical rectangles. The arrows on the base-points of the stixels show the estimated object velocity. The color encodes the distance.



(d) Segmentation of the Stixel World into static background/infrastructure and moving objects. The color represents a group of connected stixels with similar motion. Brown stixels are flagged as potentially inaccurate.

Fig. 4. Visual outline of the stereo processing pipeline. Dense disparity images are computed from sequences of stereo image pairs. From this data, the Stixel World is computed, a very compact and efficient intermediate representation of the three-dimensional environment. Stixels are tracked over time for estimating the motion of objects. This information is used to extract both static infrastructure and moving objects for subsequent processing tasks.

C. Traffic Light Recognition

Stopping at a European traffic light requires a viewing angle of up to $120°$ to be able to see the relevant light signal right in front of the vehicle. At the same time, a comfortable reaction to red traffic lights on rural roads calls for a high image resolution. We chose a 4 MPixel color imager and a lens with a horizontal viewing angle of approximately $90°$ as a compromise between performance and computational burden.

From an algorithmic point-of-view, traffic light recognition involves three main problems: detection, classification and selection of the relevant light at complex intersections. To avoid a strong dependency on the map, we apply an image based localization method consisting of an off-line and an on-line step, as follows.

Off-line, an image sequence is recorded while driving towards the intersection of interest. For these recorded images, we compute highly discriminative features in manually labeled regions around the relevant traffic lights. These features are stored in a data base.

While driving in on-line mode, the features in the actual image are matched against this data base. The resulting matching hypotheses allow for both the identification of the best-matching image in the data base and the determination of the location of the relevant traffic light in the current image. The correspondent image regions serve as input for the subsequent classification step. Classification follows the principle introduced in [24]. The detected regions of interest are cropped and classified by means of a Neural Network classifier. Each classified traffic light is then tracked over time to improve the reliability of the interpretation.

The classification task turned out to be more complex than expected. While roughly $2/3$ of the 155 lights along the route were as clearly visible, the rest turned out to be very hard to recognize (please note that most intersections are equipped with two or more traffic lights for each lane). Some examples are shown in Fig. 5. Red lights in particular are

Fig. 5. Examples of hard to recognize traffic lights. Note, that these examples do not even represent the worst visibility conditions.

very challenging due to their lower brightness. One reason for this bad visibility is the strong directional characteristic of the lights. While lights above the road are well visible at larger distances, they become invisible when getting closer. Even the lights on the right side, that one should concentrate on when getting closer, can become nearly invisible in case of a direct stop at a red light.

D. Radar Sensors

Monitoring of crossing scenarios in rural roads and the all-around perception for lane merges, vehicle side surveillance and round-about monitoring forced us to extend the radar platform. Three additional long-range radars (left, right, and rear) and a set of four short-range radars were selected to fulfill these perception tasks.

While the side long-range radars are mainly used to monitor crossing traffic in urban and rural intersections up to 200 m, the two frontal near-range radars have to fulfill various perception tasks at the same time. The reliable detection and tracking of vulnerable road users like pedestrians and bicycles in the complete frontal region up to 40 m is one of the most challenging tasks. Another important scenario is the robust monitoring of roundabout traffic taking into account the various topology and infrastructure constraints that were present on the route. For the side and rear surveillance of the vehicle the frontal near-range radars are complemented by the radars integrated at the rear bumper to deliver data of all moving objects in that area. The near-range radar signal processing is based on the untracked detection level of the sensors, called targets. A signal processing chain starting with an extensive pre-processing of each single sensor followed by a radar networking stage, clustering algorithm and multi-object target tracking had been realized to deliver a comprehensive object list as output of the radar processing unit.

III. Digital Road Map

In this project, a detailed digital road map was used to support motion planning. Such a map contains significantly more information than today's navigation maps. We store all those static properties of the environment that are necessary for driving, but cannot be reliably detected by sensors. For example, we explicitly store the layout of drivable lanes which is especially hard to detect within intersections.

Our map was created in a semi-automated process based on imagery from a stereo camera: For each stereo image pair, a dense disparity image and a 3D reconstruction of the vehicle's close environment are computed. These 3D points are projected onto the world plane and accumulated based on a reference trajectory (see background of Fig. 6). To ensure congruency, the same stereo images are also used for extracting the point feature map and the map containing visible lane markings, cf. Sec. IV. The reference trajectory was recorded by a DGPS-aided inertial navigation system whereas online localization during autonomous driving does not require such a costly system. For mapping and map maintenance we employed tools from the OpenStreetMap project (OSM) [25].

The road map consists of lane segments, which in the following we will refer to as *lanelets*. A lanelet is a driveable section of a lane, which — with the exception of lateral lane changes — has exactly one entry and one exit. Such a segment is described by two polylines, representing the left and right margins, respectively. Within the OSM formalism, we define such a drivable section as a relation containing the two polylines as members with roles `left` and `right`. Nodes shared by two road sections at the beginning and end define the predecessor-successor relationship of lane sections. This establishes a routable, directed graph which represents the topology of the road network. Fig. 6 shows an example of our roadmap: Six lanelets are shown (identified with numbers 1–6) as well as a graph representing the respective lane topology. To map the Bertha Benz Memorial Route, about two thousand lanelets were annotated manually.

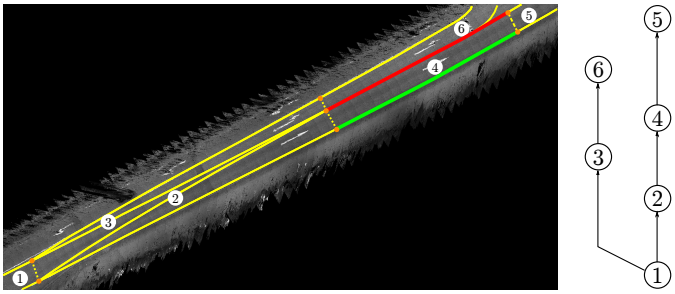


Fig. 6. Geometries of lanes and resulting lane topology. Lanelet 4 is highlighted: the right border is displayed in green, the left in red. Lane segments are interconnected at the orange dots.

Additional information required to make driving decisions is supplied in the form of *relations*. For example the two basic maneuvers *merge* and *yield*, see Sec. V-A5, are both expressed using the same type of relation: They contain a stop line, a reference line, and references to the prioritized lanelet for the *yield* maneuver or to the lanelet to merge with for the *merge* maneuver. The relation has two event points, i.e. `approach` and `complete`, that trigger when the vehicle passes them. Events are processed by the behavioral state chart which will be introduced in Sec. V-A. In a similar way, traffic lights are modeled.

IV. Map Relative Localization

One cornerstone of the presented autonomous system is the precise localization of the vehicle. The rich information

from digital road maps as introduced in Sec. III can only be exploited if a high-precision ego-localization solution is available. In fact, the sought localization solution is required to yield *map-relative* localization estimates with an accuracy up to 20cm. We developed two complementary map relative localization algorithms. The first system detects point-shaped landmarks in the immediate vicinity of the ego vehicle and is specifically effective in urban areas with large man-made structures. The other system exploits lane markings and curbstones as these are reliably detectable in rural areas and translates observations of these objects into a map-relative localization estimate.

A. Feature-based Localization

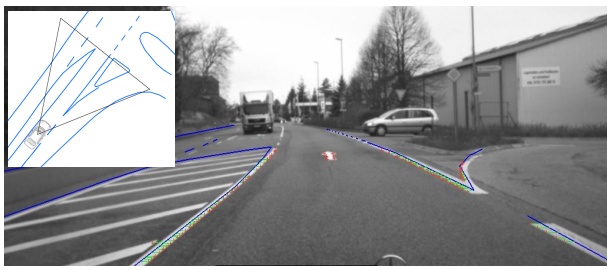
The principle underlying feature based localization is illustrated in Fig. 7a. The top image shows one frame of a stereo image sequence recorded in a mapping run. The image at the bottom of Fig. 7a has been acquired during an autonomous test drive from a rear facing monocular camera. Clearly both images have been obtained from approximately the same position and angle, yet at a different time of the year. The two images are registered spatially by means of a descriptor based point feature association: salient features of the map sequence (so-called *landmarks* shown in blue in Fig. 7a) are associated with detected features (red) in the current image of the vehicle's rear facing camera. Given the 3D positions of these landmarks have been reconstructed from the stereo image map sequence, it is possible to compute a 6D rigid-body transformation between both camera poses that would bring associated features in agreement. Fusing this transformation with the global reference pose of the map image and the motion information from wheel encoders and yaw rate sensors available in the vehicle, an accurate global position estimate can be recovered. More details on this feature-based localization can be found in [26]–[28].

In feature-rich environments like urban areas the proposed method yields excellent map-relative localization results achieving centimeter accuracy in many cases. In suburban and rural areas, however, the required landmark density may drop below a required reliability level and needs to be complemented by the method of Section IV-B. In fact, both of these methods always run in parallel and are fused in a filter framework using unscented Kalman filters. The framework handles out-of-sequence measurements thereby avoiding issues related to latency.

B. Lane-Marking-Based Localization

In rural areas often the only static features along the road are the markings on the road itself. Thus, we extend our localization algorithms with a localization system relative to the marked lanes. In a first step, a precise map containing all visible markings is built. To obtain congruent maps, the same tools and image data as described in Section III are used. In addition to the road markings (solid and dashed) and stop lines, also curbs and tram rails are annotated in the map. For the online localization step, a local section of this map is projected into the current image. Road markings

(a)



(b)

Fig. 7. (a) Landmarks that are successfully associated between the mapping image (top) and online image (bottom) are shown. (b) Detected lane markings (red), sampled map (blue) and corresponding residuals (green).

are detected in the image and their position is compared to the map (Fig. 7b). The matching is done with a nearest neighbor search on the sampled map and the resulting residuals are minimized iteratively using a Kalman filter. In suburban areas, the boundary lines of the road are often substituted with curbs. In this case, we support the measurements with a curb classifier described in [29]. The complete process of lane-based localization is described in [30].

V. Motion Planning and Control

Our approach to motion planning was to separate it into two distinct tasks (cf. Fig. 2). At the top level is a module we call *behavior generation*. It is responsible for translating perceived objects, information from the digital map, give-way rules, etc. into geometric constraints. Subsequently, the *trajectory planner* computes the desired path of the vehicle as a function of time. This trajectory is obtained by solving a geometric problem that has been posed as a nonlinear optimization problem with nonlinear inequality constraints. Trajectory planning provides an input to the *trajectory controller*. It stabilizes the vehicle and guides it along the planned trajectory. These three components are addressed in the following subsections.

A. Behavior Generation

Behavior generation can be modeled elegantly using a *state chart* notation. Depending on the current driving situation, behavior generation formulates constraints that stem from the current *driving corridor*, *static obstacles*, *dynamic objects*, and *yield and merge* rules. These constraints will be discussed below.

1) State chart: Behavior generation is defined as a hierarchical, concurrent state machine. The notation used is also known as a *Harel state chart* and was first described in [31]. The state chart notation allows for clear and comprehensible modeling of reactive systems, i.e. systems that are driven by processing a stream of *events*. The notation allows for specification of *concurrent* states, i.e. setting up multiple state charts in parallel, which react to the same events, but transition independently. States can be nested in a hierarchy of super- and substates, enabling a top-down design of complex reactive systems.

Fig. 8 shows a part of the state chart that was used in the project. State names are prefixed with `St`. The left part of the figure illustrates the concept of concurrency. When active, the system is simultaneously running four state charts, `StPathPlanning`, `StAnalyseObjects`, `StManageTrafficLights` and `StManageGiveWay`. The right part provides a detailed view of the substates of `StManageGiveWay`, which defaults to be in state `StApproach` (the default state of a state chart is indicated by a black dot). If the vehicle passes the trigger point `approaching` of a right-of-way relation (cf. Sec. III), event `T` is triggered and the substate chart transitions to state `StGiveWay`. This state contains another nested state chart, `StDriveAutomatically`, that will initially remain in state `StSituationUnclear` (which always generates a stop line constraint at the entry of the intersection) until the vehicle has approached close enough so that its sensors can cover the intersection completely. For testing purposes, the driver can overrule the vehicle's decisions by pulling a lever switch (event `D`). The state machine will then transition in state `StDriveManually` that will enter an intersection upon confirmation of the operator.

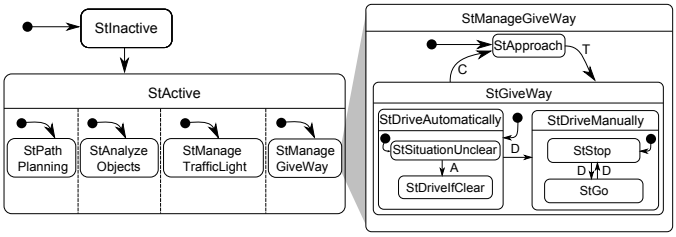


Fig. 8. Excerpt from the behavioral state chart.

2) Driving corridor: The sequence of lanelets that form the route to the destination, or a local section of it, is called the driving corridor. To determine the driving corridor, the vehicle pose is matched to exactly one lanelet of the map, considering the distances to the segment boundaries and the angle of deviation between the vehicle's orientation and the centerline of the segment. Starting from this initial segment, a shortest path search is expanded within the lane topology (Sec. III) to determine the complete driving corridor that leads to the destination. The trajectory planner will use the driving corridor as a constraint, and asserts that the vehicle stays in its bounds.

3) Static obstacles: As described in Sec. II, static obstacles are represented as stixels. For all stixels within bounds of the driving corridor, it is decided before trajectory planning whether the vehicle is supposed to pass them on the left or right. For this decision, a minimum vertex graph cut is used (cf. [32]). The structure of the graph to be cut is illustrated schematically in Fig. 9a. Each individual stixel corresponds to a node in the graph. The two larger nodes represent the left and right bound of the driving corridor. Two nodes are connected if it is geometrically infeasible to pass between the corresponding stixels, or between the corresponding stixels and the respective corridor bound. The graph will now be cut into two sections. This is done in a minimal way, i.e. by removing the smallest possible amount of nodes required to separate the left and right corridor bounds. In Fig. 9a, the graph is cut by removing one node (orange). If the cut set is not empty, passage at this point is not possible. Behavior generation will put an active stopline at a suitable position (orange). After having assigned all stixels to either the left or right corridor bound, a polygonal hull is computed that defines the geometric constraint for trajectory planning (Fig. 9b).

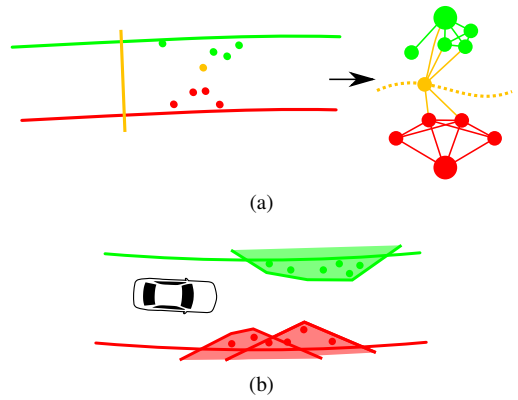


Fig. 9. Preprocessing of obstacle data, respecting the run of the driving corridor.

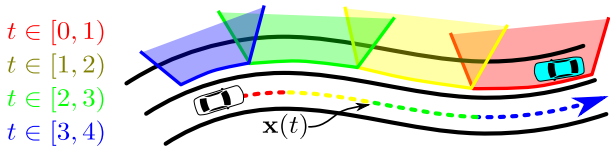


Fig. 10. Constraints for an oncoming Object (cyan). The trajectory is only constrained by polygons of corresponding color.

4) Objects: For reacting correctly towards other traffic participants, it is essential to anticipate their behavior in the near future. To achieve this, every object provided by the sensor system (Sec. II) is associated with one or more lanelets. For each of these lanelets (Sec. III), a shortest path search yields all corridors that the object can reach within a limited time horizon. For each of these corridors, a trajectory for the object is predicted, assuming that the vehicle follows the lane and maintains its distance to the right bound. Similar to the static obstacles, polygons are created for each of the trajectories. However, because the object is in motion, each polygon is active for a certain period of time only. In Fig. 10, these polygons are illustrated schematically for the case of an oncoming object with colors indicating different time intervals.

5) Yield and merge: Crossing any intersection can be expressed in terms of two basic maneuvers: One in which the own lane is crossed by a prioritized lane (*yield* type), and one in which two lane segments converge into one (*merge* type). Assume that the ego vehicle is approaching a T-junction with the intent of turning left and does not have the right of way. Its path will intersect the lane of traffic approaching from the left, which it will have to *yield* to. Immediately after this, it has to *merge* into the traffic approaching from the right. The two maneuvers cannot be treated strictly one after the other, but they must be treated simultaneously. It is the concurrent nature of the behavioral state chart (Sec. V-A1) that allows for modeling the two sub-problems separately, but to treat them simultaneously. Note that driving through a roundabout can be expressed as a merge maneuver and a subsequent, prioritized, right turn.

To formulate yield and merge type constraints, objects are considered in a space-time plane (cf. [33]), which is spanned by time on the abscissa and distance from the vehicles front bumper on the ordinate. A yield maneuver implies a space-time constraint in form of an axis aligned rectangle. A merge maneuver calls for a more general shape of constraints as illustrated in Fig. 11b.

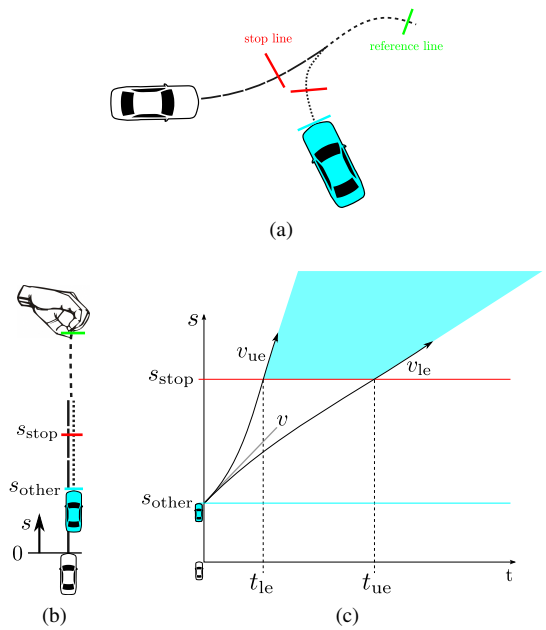


Fig. 11. Merging into traffic: (a) Top view, white indicates the ego vehicle, cyan the object vehicle. (b) Top view converted into a 1D arc length representation. (c) Space-time constraint computed by assuming a lower estimate v_{low} and upper estimate v_{up} for the object vehicle's speed, respectively.

B. Trajectory Planning

The trajectory planner computes an optimal trajectory $\mathbf{x}(t) = (x(t), y(t))^{\mathrm{T}}$ that minimizes the integral

$$J[\mathbf{x}] = \int_{t_0}^{t_0+T} j_{\text{offs}} + j_{\text{vel}} + j_{\text{acc}} + j_{\text{jerk}} + j_{\text{yawr}}\,\mathrm{d}t$$

subject to non-linear inequality constraints of the general form $\mathbf{c}(\mathbf{x}(t)) \leq 0$ provided by the behavior generation module. We now discuss the individual summands of the integrand. All summands contain a weighting factor w_{offs}, w_{vel} etc.

$$j_{\text{offs}}(\mathbf{x}(t)) = w_{\text{offs}} \left| \frac{1}{2}(d_{\text{left}}(\mathbf{x}(t)) + d_{\text{right}}(\mathbf{x}(t)) \right|^2$$

is the term to make the trajectory pass in the middle between the two edges of the corridor. The functions d_{left} and d_{right} are the signed distance functions towards the bounds of the driving corridor, the distance being positive for all points left of the bound, and negative for all the points to the right. The term

$$j_{\text{vel}}(\mathbf{x}(t)) = w_{\text{vel}} \left| \mathbf{v}_{\text{des}}(\mathbf{x}(t)) - \dot{\mathbf{x}}(t) \right|^2$$

represents the quadratic error of the velocity vector of the trajectory compared to a reference velocity vector $\mathbf{v}_{\text{des}}$. The absolute value v_{des} of $\mathbf{v}_{\text{des}}$ corresponds to the current speed limit from the digital map. The direction of the velocity vector is orthogonal to the gradient of the distance functions of the corridor, such that the target direction is parallel to the bounds of the corridor. The two terms described so far specify the desired behavior of the trajectory: it should follow the middle of the driving corridor at a specified velocity. They have to be balanced against the following smoothness terms, which are motivated by driving dynamics and comfort. The term

$$j_{\text{acc}}(\mathbf{x}(t)) = w_{\text{acc}} \left| \ddot{\mathbf{x}}(t) \right|^2$$

penalizes strong acceleration in the transverse and longitudinal directions, and thus the forces acting on the passengers. The jerk term

$$j_{\text{jerk}}(\mathbf{x}(t)) = w_{\text{jerk}} \left| \dddot{\mathbf{x}}(t) \right|^2$$

imposes smoothness of the trajectory by dampening rapid changes in acceleration. The suppression of acceleration and jerk alone will not prevent rapid changes of direction that occur when driving along the trajectory. For this purpose, we introduced a term into the functional which attenuates high yaw rates:

$$j_{\text{yawr}}(\mathbf{x}(t)) = w_{\text{yawr}} \dot{\psi}(t)^2 \,,$$

where the yaw rate is given as the derivative of the tangent angle $\psi(t) = \arctan \frac{\dot{y}(t)}{\dot{x}(t)}$. The optimal trajectory must minimize the described energy functional, but, at the same time, obey constraints that assure freedom of collision and containment in the driving corridor. These constraints were described in the previous section, and we refer to them as *external* constraints. Furthermore, there are *internal* constraints which result from limits of the vehicle kinematics and dynamics. At low speeds, the curvature of the trajectory is limited by the steering geometry of the vehicle, so

$$|\kappa(t)| < \kappa_{\max} \quad \text{with} \quad \kappa(t) = \frac{\dot{x}(t)\ddot{y}(t) - \dot{y}(t)\ddot{x}(t)}{\sqrt[3]{\dot{x}^2 + \dot{y}^2}} \,.$$

At higher velocities, this the driving limit usually becomes dominated by the friction limit of the tires. This limit can be thought of as a circle of forces [34], and

$$\|\ddot{\mathbf{x}}(t)\| < a_{\max}$$

must hold.

For minimizing $J[x]$, this variational problem is transformed to an ordinary constrained extremum problem by applying the method of finite differences [35]. In the extremum problem, both the objective function and the constraints are described via non-linear equations and inequations. As an optimization method, therefore, the method of sequential quadratic programming (SQP) [36] is used.

C. Vehicle Control

The trajectory control module feeds back the pose estimate of the vision-based localization to guide the vehicle along the planned trajectory. In this section, we will only discuss lateral control of the vehicle, as longitudinal control was implemented using the model predictive controller already described in [8]. The lateral controller is similar to the path tracking controller of [37] or the lane keeping controller described in [38], but has enhanced precision and a wider operating range, that covers tight turns at inner-city intersections as well as driving on highways.

The lateral control has a feed-forward part for disturbance compensation and a feed-back control part for stabilization of the lateral position. In a first step, a desired yaw rate command $\dot{\psi}_{des}$ is calculated. In a subsequent step, a stationary inverse single track model is used to transform the desired yaw rate in a desired steering angle δ_{des} as command input to the actuator. To ensure steady state accuracy, a steering offset compensation is adapted during run time using measured yaw rate and measured steering angle. The full structure is shown in Fig. 12.

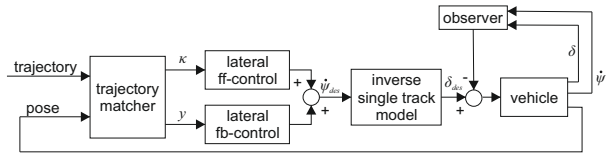


Fig. 12. Schematic structure of the lateral control loop.

The feed-forward control part calculates a desired yaw rate of the vehicle from the curvature κ of the trajectory at a near look-ahead point. The look-ahead distance is velocity dependent to compensate the reaction time of the vehicle and the steering system. The feed-back control part aims to minimize the lateral displacement of the vehicle to the desired trajectory. The two states lateral displacement y_e and its temporal derivative $\dot{y}_e$ generated by a Luenberger observer are used for stabilization. The displacement y provided by the observer is directly feed back but used for the determination of a velocity dependent desired track angle θ_{des}. The second observer state $\dot{y}_e$ is used to compute the track angle θ_{meas}. A P-controller is employed to obtain the feed-back component of the yaw rate (see Fig. 13).

In the last step, an inverse stationary single track model is used to compute the required wheel steering angle from the desired yaw rate. For steady state accuracy, a steering angle offset observer is used to compensate both deviations in the inverse single track model as well as a steering angle offset in the steering actuator. By measuring yaw rate, steering angle and velocity, this observer inherently introduces an integrator behavior within the closed loop system for steady state accuracy.

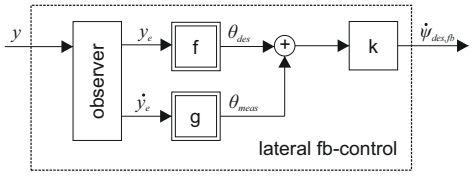


Fig. 13. Schematic structure of the feed-back control for the lateral position. f and g are nonlinear functions, k is a constant factor.

VI. Results

In August 2013, the Mercedes-Benz S-Class S 500 Intelligent Drive successfully completed the Bertha Benz Memorial Route in fully autonomous mode. The route was divided into six intervals that were driven several times in real traffic at various hours of the day. Splitting the course was necessary to ensure that operating times of the safety driver never exceeded 45 minutes. The speed limits along the route ranged from 30km/h or 50km/h in villages and cities to 100km/h on country roads. About 54km of the route passes through urban areas. The autonomous vehicle had to handle 155 traffic lights and traversed 18 roundabouts in various traffic conditions. Numerous pedestrians and bicyclists were encountered along the road. Some of the most challenging situations for perception and trajectory planning included narrow passages where parked vehicles forced our robot to wait for oncoming traffic.

Sudden human intervention of the safety driver was not required. In two occasions on our first drive, the vehicle came to a safe stop behind an obstacle and did not proceed without intervention of the human operator. In the first situation, the lane was blocked by a construction site. In the second case, Bertha stopped behind a delivery van parking in second row. Since we prohibited the car from entering the opposite lane by more than one meter, the situation could not be resolved automatically and the operator had to take back control to proceed.

The Bertha Benz vehicle has been tuned for a defensive driving style, e.g. the robot will generally wait for clear gaps between preceding vehicles before entering a roundabout. Some passengers thus compared the robot to a human learner taking driving lessons. Detailed video footage of our experiments can be reviewed on our websites,[1] respectively.

VII. Conclusion

The experiment presented in this article shows that autonomous driving is feasible — not only on highways but even in very complex urban areas such as the Bertha Benz memorial route. The key features that enabled this result are radar and stereo vision sensing for object detection and free-space analysis, monocular vision for traffic light detection and

[1] Available at http://www.youtube.com/Daimler, keyword *S 500 Intelligent Drive* and http://www.fzi.de/forschung/projekt-details/S_500_Intelligent_Drive.

object classification, digital maps complemented with vision-based map-relative localization, versatile trajectory planning and reliable vehicle control. Compared to other autonomous vehicle, we believe the S 500 INTELLIGENT DRIVE advances the state of the art in the variability of handled traffic situations and in the employed sensor setup. The chosen sensor configuration is closer to current automotive serial production in terms of cost and technical maturity than in many autonomous robots presented earlier.

Although the autonomous vehicle successfully completed the 103 km of the historic route from Mannheim to Pforzheim, the overall behavior is far inferior to the performance level of an attentive human driver. A further improvement of the robot's ability to interpret a given traffic scenario and to obtain a meaningful behavior prediction of other traffic participants is imperative to achieve comparable behavior. We found that the recognition of traffic lights needs to be improved in terms of recognition rates, especially at distances above $50m$. For traffic light generation and the generation of digital maps, not only the technical performance but also the scalability of the chosen solutions in terms of a commercial roll-out is pivotal. Future work on autonomous driving should also focus on reducing the requirements on the accuracy and update frequency of digital maps. This can only be achieved by improving the sensor setup regarding robustness, availability and redundancy.

REFERENCES

[1] E. Dickmanns, B. Mysliwetz, and T. Christians, "An integrated spatio-temporal approach to automatic visual guidance of autonomous vehicles," *Systems, Man and Cybernetics, IEEE Transactions on*, vol. 20, no. 6, pp. 1273–1284, 1990.

[2] U. Franke, S. Mehring, A. Suissa, and S. Hahn, "The Daimler-Benz Steering Assistant - a spin-off from autonomous driving," in *IEEE Intelligent Vehicles Symposium*, Paris, Oct 1994.

[3] E. Dickmanns, R. Behringer, D. Dickmanns, T. Hildebrandt, M. Maurer, F. Thomanek, and J. Schiehlen, "The seeing passenger car 'vamors-p'," in *Intelligent Vehicles '94 Symposium, Proceedings of the*, 1994, pp. 68–73.

[4] D. Pomerleau and T. Jochem, "Rapidly adapting machine vision for automated vehicle steering," *IEEE Expert*, vol. 11, no. 2, pp. 19–27, 1996.

[5] S. Thrun, M. Montemerlo, H. Dahlkamp, D. Stavens, A. Aron, J. Diebel, P. Fong, J. Gale, M. Halpenny, G. Hoffmann, K. Lau, C. Oakley, M. Palatucci, V. Pratt, P. Stang, S. Strohband, C. Dupont, L.-E. Jendrossek, C. Koelen, C. Markey, C. Rummel, J. van Niekerk, E. Jensen, P. Alessandrini, G. Bradski, B. Davies, S. Ettinger, A. Kaehler, A. Nefian, and P. Mahoney, "Stanley: The robot that won the darpa grand challenge," *Journal of Field Robotics*, vol. 23, no. 9, pp. 661–692, 2006.

[6] C. Urmson, J. Anhalt, D. Bagnell, C. Baker, R. Bittner, M. N. Clark, J. Dolan, D. Duggins, M. Gittleman, S. Harbaugh, Z. Wolkowicki, J. Ziglar, H. Bae, T. Brown, D. Demitrish, V. Sadekar, W. Zhang, J. Struble, M. Taylor, M. Darms, and D. Ferguson, "Autonomous driving in urban environments: Boss and the urban challenge," *Journal of Field Robotics*, vol. 25, pp. 425–466, 2008.

[7] 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, Sep. 2008.

[8] A. Geiger, M. Lauer, F. Moosmann, B. Ranft, H. H. Rapp, C. Stiller, and J. Ziegler, "Team annieway's entry to the 2011 grand cooperative driving challenge," *IEEE Transactions on Intelligent Transportation Systems*, vol. 13, no. 3, pp. 1008–1017, 2012.

[9] (2013). [Online]. Available: vislab.it/proud/

[10] J. Wei, J. Snider, J. Kim, J. Dolan, R. Rajkumar, and B. Litkouhi, "Towards a viable autonomous driving research platform," in *Intelligent Vehicles Symposium (IV), 2013 IEEE*, June 2013, pp. 763–770.

[11] H. Hirschmüller, "Accurate and efficient stereo processing by semi-global matching and mutual information," in *Proc. CVPR*, 2005, pp. 807–814.

[12] S. Gehrig, F. Eberli, and T. Meyer, "A real-time low-power stereo vision engine using semi-global matching," in *Proc. ICVS*, 2009.

[13] H. Badino, U. Franke, and D. Pfeiffer, "The stixel world — a compact medium level representation of the 3d-world," *Proc. DAGM*, 2009.

[14] D. Pfeiffer and U. Franke, "Towards a global optimal multi-layer stixel representation of dense 3d data," *Proc. BMVC*, 2011.

[15] U. Franke, C. Rabe, H. Badino, and S. Gehrig, "6D-Vision: Fusion of stereo and motion for robust environment perception," *Proc. DAGM*, pp. 216–223, 2005.

[16] D. Pfeiffer and U. Franke, "Efficient representation of traffic scenes by means of dynamic stixels," *IEEE IV Symp.*, 2010.

[17] F. Erbs, B. Schwarz, and U. Franke, "Stixmentation - Probabilistic stixel based traffic scene labeling," *Proc. BMVC*, 2012.

[18] M. Enzweiler and D. M. Gavrila, "A multi-level Mixture-of-Experts framework for pedestrian classification," *IEEE Trans. on IP.*, vol. 20, no. 10, pp. 2967–2979, 2011.

[19] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," *Proc. CVPR*, pp. 886–893, 2005.

[20] C. Wöhler and J. K. Anlauf, "An adaptable time-delay neural-network algorithm for image sequence analysis," *IEEE Transactions on Neural Networks*, vol. 10, no. 6, pp. 1531–1536, 1999.

[21] C. Wojek, S. Walk, and B. Schiele, "Multi-cue onboard pedestrian detection," *Proc. CVPR*, 2009.

[22] M. Enzweiler et al., "Multi-Cue pedestrian classification with partial occlusion handling," *Proc. CVPR*, 2010.

[23] M. Enzweiler, M. Hummel, D. Pfeiffer, and U. Franke, "Efficient stixel-based object recognition," *IEEE IV Symp.*, 2012.

[24] F. Lindner, U. Kressel, and S. Kaelberer, "Robust recognition of traffic signals," *IEEE IV Symp.*, 2004.

[25] F. Ramm, J. Topf, and S. Chilton, *OpenStreetMap*. UIT Cambridge, 2007. [Online]. Available: http://openstreetmap.info/content/OpenStreetMap-Kapitel5-Mapping-Praxis.pdf

[26] H. Lategahn and C. Stiller, "City gps using stereo vision," in *IEEE International Conference on Vehicular Electronics and Safety*, Istanbul, Turkey, Juli 2012.

[27] H. Lategahn, M. Schreiber, J. Ziegler, and C. Stiller, "Urban localization with camera and inertial measurement unit," in *IEEE Intelligent Vehicles Symposium*, Gold Coast, Australien, 2013.

[28] H. Lategahn and C. Stiller, "Vision only localization (submitted)," *IEEE Transactions on Intelligent Transportation Systems*, vol. 00, no. 00, 2014.

[29] M. Enzweiler, P. Greiner, C. Knoppel, and U. Franke, "Towards multi-cue urban curb recognition," in *Intelligent Vehicles Symposium (IV), 2013 IEEE*. IEEE, 2013, pp. 902–907.

[30] M. Schreiber, C. Knoppel, and U. Franke, "Laneloc: Lane marking based localization using highly accurate maps," in *Intelligent Vehicles Symposium (IV), 2013 IEEE*. IEEE, 2013, pp. 449–454.

[31] D. Harel, "Statecharts: a visual formalism for complex systems," *Science of Computer Programming*, vol. 8, no. 3, pp. 231–274, Jun. 1987. [Online]. Available: http://www.sciencedirect.com/science/article/pii/0167642387900359

[32] Y. Boykov and V. Kolmogorov, "An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, pp. 1124–1137, September 2004.

[33] K. Kant and S. W. Zucker, "Toward efficient trajectory planning: The path-velocity decomposition," *The International Journal of Robotics Research*, vol. 5, no. 3, pp. 72–89, 1986.

[34] H. Pacejka and I. Besselink, *Tire and Vehicle Dynamics*, ser. Butterworth Heinemann. Butterworth-Heinemann, 2012. [Online]. Available: http://books.google.de/books?id=ETnzam6qS2oC

[35] A. K. Kaw and E. E. Kalu, *Numerical Methods with Applications*, 2nd ed. http://www.autarkaw.com, 2010. [Online]. Available: http://numericalmethods.eng.usf.edu/topics/textbook_index.html

[36] J. Nocedal and S. J. Wright, *Numerical Optimization*, 2nd ed. New York: Springer, 2006.

[37] J. Ziegler, M. Werling, and J. Schröder, "Navigating car-like robots in unstructured environments using an obstacle sensitive cost function," in *Intelligent Vehicles Symposium*, 2008.

[38] H. Fritz, A. Gern, H. Schiemenz, and C. Bonnet, "CHAUFFEUR assistant: A driver assistance system for commercial vehicles based on fusion of advanced ACC and lane keeping," in *Intelligent Vehicles Symposium*, 2004.