Toward Perception-Driven Urban Environment Modeling for Automated Road Vehicles

Jens Rieken, Richard Matthaei and Markus Maurer[1]

***Abstract*—Automated driving is a widely discussed topic nowadays. Impressive demonstrations have shown the potentials of vehicle automation. However, many projects in the context of automated driving use a priori data in order to compensate insufficiencies in perceiving and understanding the vehicle's environment. Additionally, in terms of functional safety and redundancy, it is not yet known whether such localization- and map-based approaches are really path breaking.**

This is the reason why we focus on on-board perception also of the stationary urban environment. While object tracking is a commonly used approach, the combination of grid-based and object-based representations for environment perception is still a research topic. The sufficient perception of lanes and drivable areas is an unsolved issue in urban environment. Several perception modules have to collaborate for a suitable representation of the vehicles' surroundings.

In this paper, we present the latest contributions of the project *Stadtpilot* to a perception-driven modeling of urban environments. We propose a lane detection approach which is based on a grid-based representation of different environmental features. Our approach is able to detect multi-lane structures and it is capable to deal with complex lane structures which are typical of urban roads. The extracted features are stabilized by a tracking module. Additionally, we incorporate a free-space representation which data is not derived implicitly from detected targets, but based on an explicit ground representation. Extensions of our dynamic classification module focus on the start/stop behavior of other road users in order to enhance the completeness of track list (mobile objects) and grid (stationary environment). The presented algorithms run in real-time on a standard PC and are evaluated with real sensor data.

I. Introduction

The development of advanced driver assistance systems (ADAS) and automated road vehicles imposes high requirements on the perception of the host vehicle's environment. In particular in inner-city scenarios, various types of traffic participants and structures on and next to the roads make the perception of relevant road users, static obstacles and the road course an even more challenging task. The host vehicle has to be able to perceive all of these environmental elements and model their context up to a sufficient extent in order to perform safe and comfortable driving tasks.

The detection and tracking of dynamic elements in terms of other road vehicles is widely discussed and a common research topic. When advancing from typical highway scenarios into the urban domain, additional effort has to be spent on the detection of vulnerable road users (VRUs). Next to the dynamic elements, the stationary environment provides features about the road course and drivable regions. On highway-like roads, they are mainly restricted to lane markings and guardrails. In the urban domain, several more features of the stationary environment are available and relevant for the determination of the road course, like curb structures, house walls and parking vehicles at the side of the road. Road narrows and blocked roads due to temporarily stopping vehicles, e.g. at traffic lights, or temporary halting delivery services, are likely to appear and thus have to be perceived and modeled.

All of those elements make the perception and modeling a complex task in inner-city scenarios. No comprehensive solution, neither for lane detection nor the representation of static obstacles or their combination with dynamic elements, is yet available. Many approaches use a priori databases like enhanced digital maps to compensate insufficiencies of the environment perception and context modeling, which is a questionable approach in terms of functional safety and system redundancy. The risk of relying on outdated, missing or even erroneous a priori data is just one aspect which shows that the usage of this data might be unsustainable for public traffic purposes. From our point of view, a *perception-driven* approach is required for automated road vehicles, in which the vehicle has to perceive and model all required aspects of its environment via on-board sensors.

Contribution of this paper

In this paper, we present the latest contributions of the project *Stadtpilot* [1], [2], [3] to a comprehensive perception-driven modeling of the vehicle environment. We propose additions to our grid-based lane detection approach (see [4], [5]). Our approach does not require predefined lane geometries and is thus able to adapt to complex lane geometries, which frequently occur in the urban domain. Next to elevated obstacles and textural data of the ground surface, curb structures and an explicit modeling of the currently visible ground surface, based on the approaches in [6], provide further input for the road extraction process. The extracted features are filtered by a Kalman-based tracking algorithm to provide robust lane hypotheses. This approach is able to extract and track multiple lanes while it can also deal with changing numbers of available lanes and their geometries. Based on the extracted road courses, static obstacles are extracted along the lanes' paths. The effective combination of different algorithms and software modules is shown by the example of a dynamic classification module, which enables the vehicle to distinguish movable targets from elements of the stationary environment even in case of temporarily stopped vehicles.

[1]All authors are with the Institute of Control Engineering, Technische Universitaet Braunschweig, 38106 Braunschweig, Germany `rieken, matthaei, maurer@ifr.ing.tu-bs.de`

This paper is organized as follows. First, an overview over current research of other groups regarding perception systems for automated road vehicles is given in Section II. The main part of this contribution is structured according to the functional system architecture described in [7], whose layers concerning the environment perception are illustrated in Figure 1. Data acquisition and basic feature extraction required for the subsequent tasks are shortly introduced in Section III. The dynamic classification module is presented in Section IV,

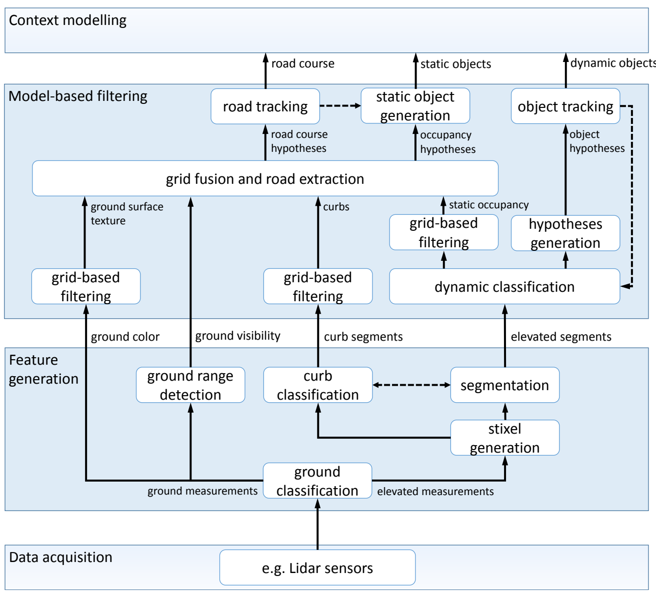


Fig. 1. Structure of the proposed environment perception system within the operational level in the sense of the functional system architecture proposed in [7].

an overview over the required dynamic object hypotheses generation and tracking module is given in Section V. The road extraction, based on the grid-based representation of the stationary environment, and the hypotheses tracking are explained in detail in Section VI. We close with a conclusion and outlook in Section VII.

II. Related work

Many research groups are working on aspects of automated driving for road vehicles. Based on the definition of automated vehicles given in [7] and [8], only few groups address automated driving to its full extent. A brief overview of existing projects, their perception concepts and addressed domains is given in the next paragraphs.

While the detection and tracking of dynamic elements is required to be done on-line in order to interact with other road users, especially the perception of the stationary environment is often supported by a priori information. Based on whether they require a priori data not only for navigational purposes, current projects can be categorized into two major groups, *localization-driven* and *perception-driven* approaches.

a) Localization-driven approaches: This concept follows the idea of controlling the vehicle's pose in relation to a global coordinate system. Usually, map data is used to provide information about the stationary environment, especially about the course of the lanes. Often, even more detailed maps are used to improve the host vehicle's global map-relative position due to insufficient availability of a precise GNSS-based localization. Thus, their driving capabilities do not solely rely on on-board environment perception, but require a localization relative to the a priori data, e.g. by GNSS platforms or landmark matching algorithms.

Team AnnieWay focuses on highway scenarios and uses radar sensors to perceive other road users during their participation in the Grand Cooperative Driving Challenge [9]. Information about the road course is provided via detailed maps. Map data are used for the assignment of on-line perceived objects to provided road data.

The demonstration at the Bertha-Benz historic route in 2014 relied on on-board video and radar systems for the perception of dynamic objects [10]. Positions of lane markings, traffic lights and right-of-way rules were provided via high-accurate maps, which were recorded and manually labeled in advance of the demonstration. Among others, the taken route contains highway and suburban roads, traffic lights and intersection areas.

The project *Stadtpilot* addresses the urban domain. During the first project phase, only the perception of other road users was performed via on-board sensors. Information about the stationary environment was provided solely by detailed a priori maps. Behavior and trajectory planning was done in a global coordinate system based on the localization provided by a DGPS-platform [1], [2], [11].

b) Perception-driven approaches: The second group of approaches is designed to perceive the complete environment with on-board sensors. First attempts of perception-driven systems were presented during the DARPA Urban Challenge, e.g. [12].

The project BRAiVE uses advanced computer vision systems for environment perception. The experimental vehicle is able to detect and classify traffic signs as well as lane markings via mono and stereo camera systems [13]. Other vehicles are perceived based on the detection of symmetric features as well as tail- and headlights. The algorithms are designed to work on the highway, on rural roads as well as in the urban domain. The system is able to derive the current domain from the detected features. Camera data is fused with Lidar sensor data to determine accurate distances of relevant objects. Additional map data based on the OpenStreetMap format is used to provide information about lane width and the position of traffic lights.

Lidar and camera systems are also used by the team MuCAR-3 [14], [15]. They are able to extract the course of the road without the aid of map data and they can detect splitting roads and intersections based on the perceived data. Other vehicles are detected via computer vision algorithms [15] and Lidar systems [16]. The project focuses on the application in offroad terrain.

Due to the experience gained from former activities, further research in the project Stadtpilot focuses on the perception-driven approach. This is valid for the stationary environment as well as for dynamic elements (e.g. [5], [17], [18]). Current contributions are addressed in this paper.

III. Preprocessing and Feature Generation

In our approach, an object-based representation of dynamic elements is used in combination with a grid-based representation of the stationary environment. Before model-based filtering algorithms can be applied to sensor data, several preprocessing steps are required.

A Velodyne HDL-64 S2 is used as a prototype Lidar sensor. Measurements are captured from elevated targets as well as from the ground surface around the host vehicle. The ground surface and the elevated targets contain different features and are processed by different algorithms, as described in the following paragraphs. Thus, the first task is to classify each measurement whether it belongs to the ground surface or to an elevated target.

A. Ground data processing

The ground classification combines a slope-based channel-wise point classifier with a grid-based ground surface estimation, as described in [6]. As a result, measurements are separated into elevated targets and ground surface data. Based on this classification, the reflectivity values of measurements of the ground surface are used to generate a gray-scale image of the road surface [5], similar to the approach in [19]. The visibility of the ground surface is derived and used as an additional feature for environment modeling (c.f. Section VI). Additionally, the ground surface information is used for calculating the host vehicle's ground-relative pitch angle and to correct the measured height of detected objects.

B. Processing of elevated measurements

The elevated targets are transformed into a 2.5D stixel representation (c.f. [20]), using a polar grid, as described in [17] and [5]. The algorithm was extended to handle low-density point clusters at larger distances by taking data from neighboring cells into account. The stixels are then segmented based on the algorithm first proposed in [21]. Modifications were applied in order to be able to deal with measurements from curb-like structures (see [6] for further details) and multiple targets per channel.

IV. Dynamic Classification

In order to combine the grid-based and the object-based representations efficiently, a separation of static and dynamic elements is required. In this contribution, the term *static* applies to non-movable stationary elements, such as buildings, and those elements that have not been perceived as moving yet (e.g. parking vehicles). The term *dynamic* is used for elements which are currently moving or were perceived moving in the past (e.g. a car which has been tracked approaching an intersection and is currently stopping at a traffic light).

Typically, road users are present in certain areas of the environment, e.g. on roads and on sidewalks. Other parts are occupied by stationary elements (e.g. vegetation, traffic signs, buildings). Thus, we perform a dynamic classification based on the area in which the targets occur.

Due to this area-based characteristic, a grid-based approach is suitable. Dynamic classification can be accomplished by the detection of inconsistent cell states. Earlier research of our group ([4], [22], [23]) proposed the usage of a specialized cell type to explicitly model these inconsistencies.

This *consistency layer* models each cell with a state machine (see Figure 2). The state machine is triggered by the sequence of hit (seen) and miss (not seen) updates from the sensor.

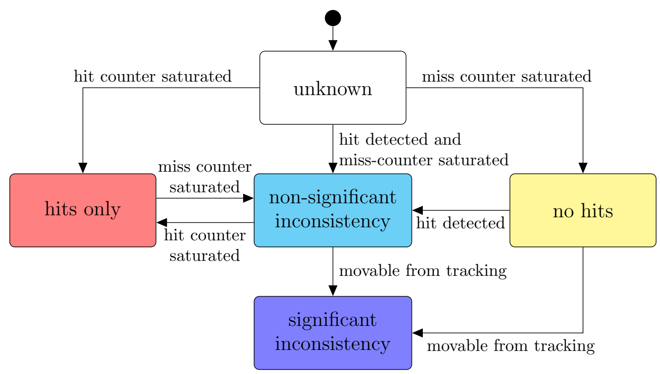


Fig. 2. Adapted state machine of the consistency layer cells. Hit and miss updates from distance-measuring sensors are used to generate state transitions, but are not feasible to trigger a transition to the final *significant inconsistency* state. Dynamic information, e.g. from an object tracking module, can be used for this purpose, as proposed in [22], [23], [4].

Information about the dynamic properties at designated positions can be derived from these states, i.e. cells with inconsistency states are likely to contain dynamic elements, while cells with hits only indicate static areas. Cells with less updates or only miss updates are considered as areas without an inferable tendency to dynamic or static state.

The consistency layer yields good results in inner-city scenarios, especially if the state transitions are adapted to the velocity of the host vehicle [4].

A. Adaption of the consistency layer

As long as the cells' consistency states are assumed to be constant over time, the possibility of state transitions is a suitable way to handle occasional false measurements. But when regarding stop-and-go traffic, e.g. at intersections, the assumption of time-constant cell states does not hold. This type of traffic generates a large number of hit updates in the same cells while stopping. Thus those cells will be classified as static even if they were known to contain inconsistencies before. A final state *significant inconsistency* is required to deal with these situations. To avoid self-affirming false classifications, transitions to this state must only be possible under unambiguous conditions. This requires multiple detections of inconsistencies, which usually cannot be guaranteed even within a short viewing range around the sensor, as shown in [4].

Additional dynamic information can be used to reduce the number of required sensor updates for a transition to the *significant inconsistency* state. In [23], direct input of velocity-measuring sensors, e.g. a radar system, was proposed. Alternatively, a feed-back of stable object hypotheses can also be used for this purpose, as considered first in [4]. In addition to the usage of direct sensor input, tracked object hypotheses are also capable to classify measurements as originating from a moving object, even if the object has stopped temporarily. This aspect is further evaluated in this contribution. Additionally, provide an association algorithm which is able to process the sensors' point cloud data under real-time conditions.

The adapted data processing structure is shown in Figure 3. Newly added modules are introduced in the next paragraphs.

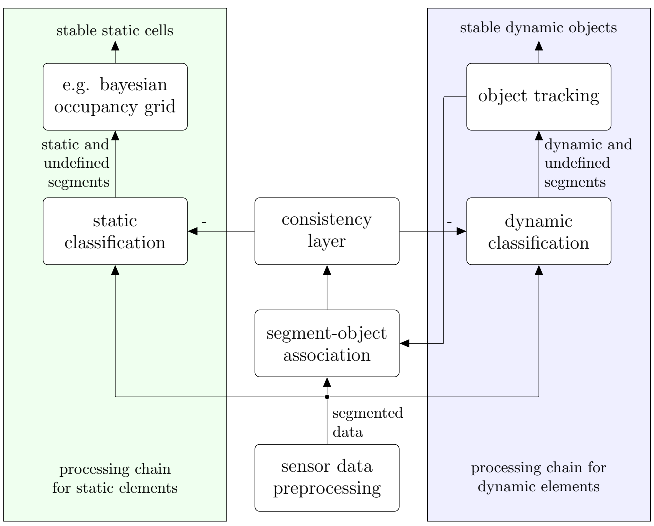


Fig. 3. Structure of the dynamic classification module. Segmented data is associated with stable object tracks and processed by the consistency layer. The stored consistency information is then used to remove unsuitable data from further processing by the respective processing chains.

B. Segment-to-object association

As stated above, the first step of the dynamic classification algorithm is the association of incoming sensor data with existing, stable dynamic object hypotheses (tracks). The association can be performed on different levels of data processing, e.g. at point cloud level or based on contour models such as the center of gravity of a segment. In inner-city scenarios, a large number of segments is typically extracted. This leads to ambiguous associations when further simplifying the segment contour. Thus, we use the sensor raw measurements for association.

In order to keep this approach computationally feasible, we project the bounding boxes of the predicted tracks into an image-like structure with a sufficient small cell size (e.g. 10 cm per dimension). This allows the association step of each measurement to be reduced to a simple point test inside the image. Uncertainties of the tracks' contour description are considered by an artificial expansion of the drawn track footprint. The data originating from a dynamic object is flagged and is used to update the consistency layer cells.

C. Data classification step

The states of the consistency layer cells are used to classify and filter incoming sensor data. The classification is performed for each measurement and a voting algorithm is applied to ensure one consistent dynamic class per segment.

Segments with stable classification are removed from the input data stream of the contradicting representation, e.g. segments with static classification are removed from the object hypotheses generation, and segments with dynamic classification are removed from the grid-based processing stages. Segments with unknown classification are forwarded to both representations to ensure the processing of all sensor data. They will stabilize over several timesteps only in the suitable representation. Once stabilized, the corresponding hypothesis in the non-suitable representation will not receive any further updates and will be removed. The results of the different processing steps are shown in Figure 4.

V. Modeling of the dynamic environment

A. Hypotheses generation

Object hypotheses are generated from all segments which are not classified as belonging to the stationary environment. For the addressed road vehicles, an oriented bounding box model is used to describe their outer contour. The contour estimation and hypotheses generation steps are presented below.

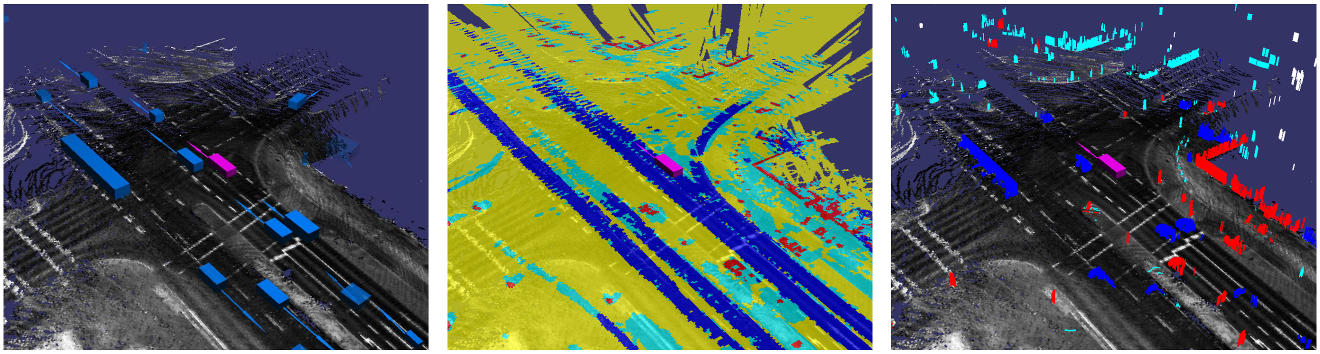
a) Contour estimation and form classification: Typical Lidar-based sensor systems generate a large number of measurements belonging to a single object. To extract the outer contour, a channel-wise skeletonization is first applied to remove irrelevant measurements. The segments' skeleton is further reduced to a simplified polygon chain using an adapted form of the Ramer-Douglas-Peucker algorithm [24].

Although the calculation of the minimum area bounding box provides useful orientation information in many cases, the evaluation of real-world sensor data has shown that this approach often fails, especially if rounded corners are dominant elements of the contour, e.g. the rear end of a vehicle. To determine the most suitable orientation, the outer contour is then classified as L-, U- or I-like using a simple set of geometric classifiers. Based on the segment type classification, the relevant segment of the contour description is selected. A fitting algorithm [25] is then used to determine the hypotheses' orientation. The skeletonization, contour estimation and the classification steps are illustrated in Figure 5a and 5b.

b) Bounding box generation: A bounding box is calculated for each segment, based on the skeleton points and the estimated contour orientation. The resulting object hypotheses are shown in Figure 5c.

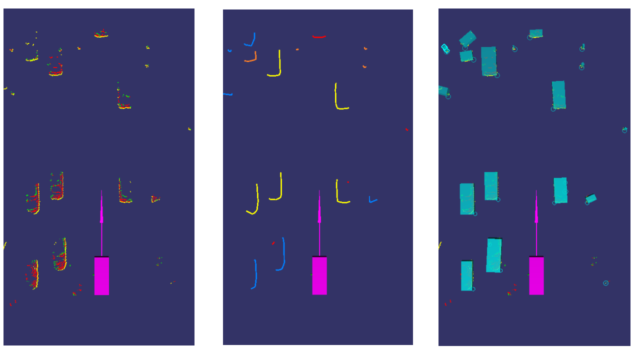
Information about the used reference point is added to each hypothesis. This point describes the best seen edge or side, whose consideration allows more stable tracking results under shape changes and occlusion conditions. We use a reference point set of nine reference points at the outer contour and the center of the box model, as introduced in [26].

The used reference point model explicitly distinguishes between the front, rear and the sides of a target. Information



(a) Tracked objects above reflectance layer (b) Consistency layer overlay (c) Classified stixel data

Fig. 4. Results of the dynamic classification. Fig. 4a shows the stable object hypotheses above the reflectance layer data. Fig. 4b shows the contents of the consistency layer: Red cells indicate areas with consistent hit information. Yellow cells mark areas with miss updates only. Cyan areas hold inconsistent information (mixture of hit and miss updates) and indicate areas of dynamic elements. Blue cells were associated with valid tracks and are thus known as areas of moving objects. Fig. 4c displays the resulting stixel classification. Colors are defined as in Fig. 4b, white stixels contain unknown dynamic classification.



(a) Skeletonization (b) Contour classification (c) Final hypotheses

Fig. 5. Processing steps of the object hypotheses generation. Fig. 5a shows the result of the segment skeletonization and classification process. Points of a segment are classified as facing skeleton points (yellow), rear skeleton points (green) and inlier points (red). Fig. 5b illustrates the contour estimation and its classification results. Contours are classified as L-Shape (yellow/orange), U-Shape (blue) and I-Shape (red). The final object hypotheses are shown in Fig. 5c.

about the type of the seen corner of a hypothesis is not extracted during the hypotheses generation yet. Thus, the used reference point has to be considered as ambiguous regarding the observed corner. Information about this ambiguity is forwarded to the tracking module for a correct interpretation of the object hypotheses.

In order to evaluate the quality of the measured dimensions, each dimensional state is flagged with a quality class. This class contains information about the detection quality of the dimension value, i.e. whether its value is based on clearly seen contour elements or whether it is a result from possibly missing data, for example due to occlusion by other objects or total reflections caused by a large angle of incidence.

B. Object tracking module

The generated object hypotheses are filtered and validated by an object tracking module. First approaches were published in [17] and further extended to improve the tracking performance and effective field of view. The tracking module is based on an Extended Kalman Filter (EKF) structure which is summarized below.

a) Prediction and association step: Each hypothesis is represented by a box contour enhanced by an additional classifier improving the contour stability. The objects' dynamics are based on a constant acceleration model in a Cartesian reference frame. The used state vector is shown in Equation 1.

$$\mathbf{x} = \left(d_x, d_y, v_x, v_y, a_x, a_y, \varphi, \text{length}, \text{width}, \text{height}\right)^T \tag{1}$$

Objects are associated with the existing hypotheses using a priority-based Local-Nearest-Neighbor algorithm. The object similarity is determined by a Mahalanobis distance over positional and dimensional states. The association algorithm exploits the marked reference point of the incoming measurements and uses the given reference point ambiguity information to resolve multiple association hypotheses. If multiple hypotheses are located within the association gating range of a measurement, associations to stable tracks are preferred.

b) Correction step and object adaption: The associated pairs of existing hypotheses and measurements are transformed to their best-matched reference point determined by the association step and are then updated by the EKF algorithm. Possible reference point ambiguities are resolved using the estimated moving direction of the hypotheses.

Created hypotheses have to be able to adapt to contour changes, e.g. when passing a tracked vehicle. If occlusion by other objects is regarded, such contour models are affected by unintended dimensional changes. Once measured dimensions might change although they were stable over a large period of time.

To avoid this while still being able to adapt the contour, if required, a contour classifier is used. Based on the evidence information provided from the hypotheses generation, the updates of the objects' dimension information during the last update cycles are evaluated. Dimensions with full visibility over a defined time period which feature a small standard deviation are considered as *stable*. The dynamic model is then configured with less process noise, resulting in a more stable contour description. In terms of the Kalman filtering theory, this adaption can be categorized as dynamic model switching.

In addition to the aforementioned steps, a Dempster-Shafer based object classification is performed using dimension and velocity information.

c) Database maintenance and object deletion: After the processing of associated measurements, the unassociated measurements are used to instantiate new object hypotheses. Existing hypotheses are validated against publishing rules which define the conditions a hypothesis has to fulfill to be considered as a valid one. The minimum number of required updates, as well as the hypothesis' moved distance is used as a criterion. Hypotheses are removed from the database if they exceed their *time-to-live*, which represents the prediction/update ratio during a limited number of past update cycles.

VI. Modeling of the stationary environment

A grid-based structure is used to hold information about the stationary environment, which includes elevated elements as well as ground surface data.

A. Multi-layer concept

Various types of information can be deducted from the environmental sensors. Each type differs in its characteristics regarding semantic information, processing algorithms and storage types. A single grid layer is not suitable for storing data from different sensor technologies [27], the same applies for different information types. In order to handle different layer types, we proposed a multi-layer concept in [5]. After a type-specific update algorithm is performed, the different layers are abstracted to a simple tristate value (free/unknown/occupied) and fused into a single fusion layer, which stores the combined information. An overview over the multi-layer concept with its current layer types is shown in Figure 6. Typical data resulting from the currently used layers and the result of its fusion are shown in Figure 7.

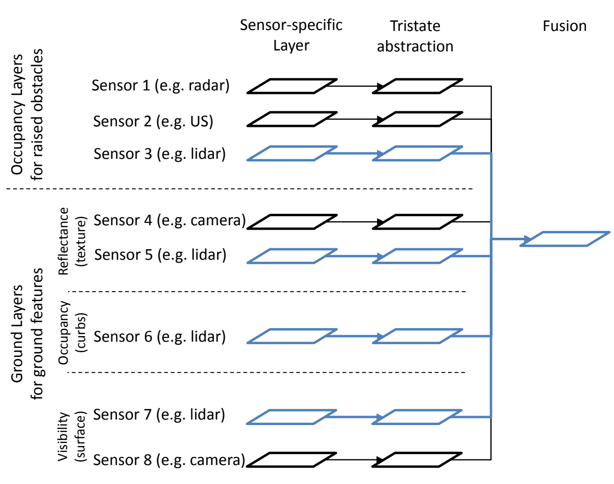
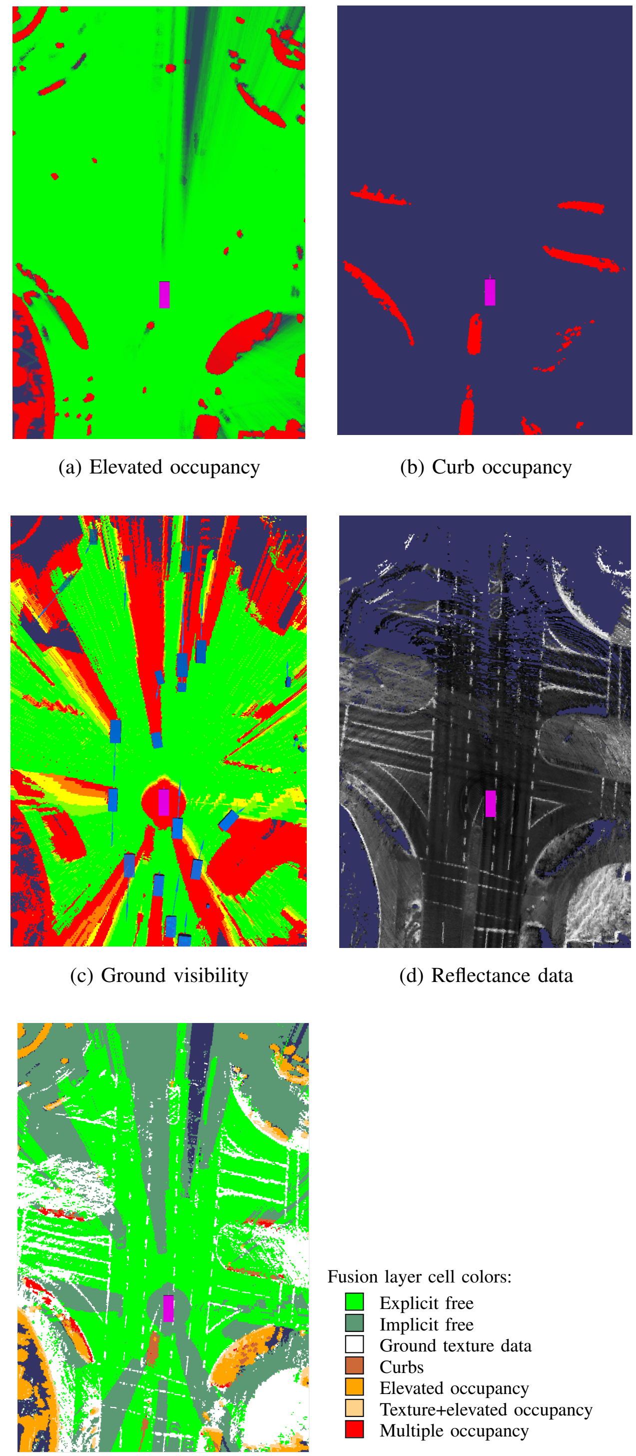


Fig. 6. Design of the multi-layer concept: Several sensor- and data-specific layers are combined by using a tristate abstraction and are fused to a resulting layer. Currently used layer types are marked in blue.



(a) Elevated occupancy

(b) Curb occupancy

(c) Ground visibility

(d) Reflectance data

(e) Fusion result

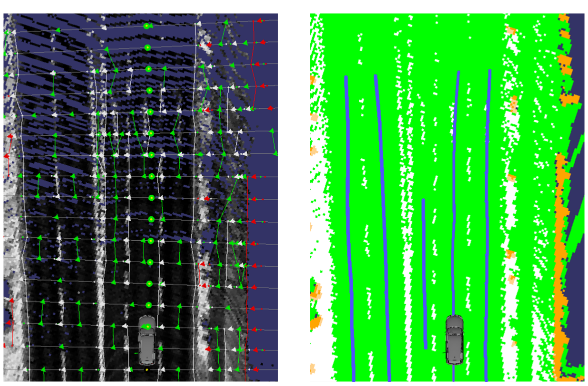
Fig. 7. Results of the stationary environment modeling at an inner-city intersection. Fig. 7a and 7b display the current data of the Bayesian occupancy grids for elevated targets and curb features. Fig. 7c illustrates the extracted visibility of the ground surface (colored by the age of the last update from green (last scan) to red (five scans before)). Hypotheses of tracked dynamic objects are shown for comparison. Fig. 7d shows the accumulated reflectance data of the ground area, mainly containing the road surface around the host vehicle. The result of the tristate fusion algorithm is drawn in Fig. 7e.

Currently, we use a combination of four layers to generate a comprehensive model of the stationary environment. Occupancy information of elevated targets is represented in a Bayesian grid. Curb-like occupancy features are stored in a second Bayesian grid, due to different effective detection ranges of the features [6]. Reflectance data of the ground surface is accumulated in another grid layer and provide information about the textural properties of the ground and road surface (c.f. [5]).

A fourth layer stores information about the visibility of the ground surface. Unlike other layers, this ground visibility layer allows us to represent free information explicitly and thus provides a conservative estimation of currently drivable areas. The ground visibility is derived from the detected ground points in the current sensor scan and stored inside the cell structure by the last time the ground surface was seen. This data can also be gathered by other sensors, e.g. a camera system. A cell is considered to be free ground if only ground measurements are present at the cells' coordinates. Ground surface measurements below objects, e.g. driving vehicles, which appear due to the sensor mounting position at the roof of the host vehicle, are excluded from further processing by detecting the presence of overhanging targets. For current evaluations, a total grid size of $280\,\mathrm{m}$ x $280\,\mathrm{m}$ with a cell size of $0.2\,\mathrm{m}$ is used.

B. Road feature extraction

In [4] and [28] we proposed an adaptive approach for road and lane extraction in urban environments which is able to deal with non-clothoid courses of the lanes, multiple lanes as well as changing number of lanes. The approach is based on an interpretation of connected free-spaces, which do not contain road markings, elevated targets and curbs. All information describing the course of the road and the lanes is considered jointly while searching for the course of the lanes, based on the grid fusion result of our multi-layer concept. An example of the extracted features is shown in Figure 8.



(a) Road elements

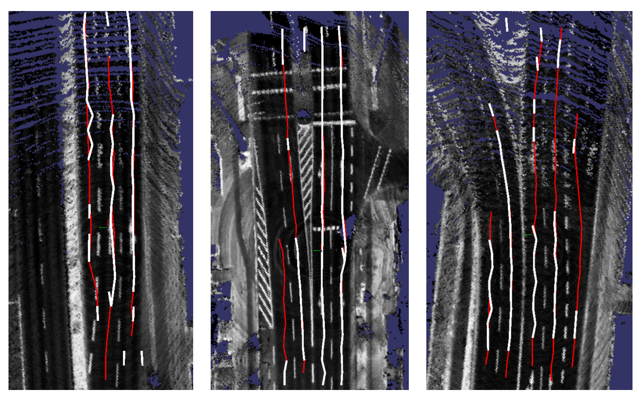
(b) Lane center lines

Fig. 8. Results of the extraction of lane center lines and road courses [29]. Fig. 8a shows the extraction of the road course based on road elements. Lines and triangles mark the free space (green), occupancy (red) and lane markings (white) features. The reflectance layer is shown in the background. Based on this data, the extracted lane center lines (blue) are shown in Fig. 8b in front of the tristate fusion layer.

C. Road hypotheses tracking and validation

The extraction of the lane center lines from the grid-based environment model provides a sufficient availability of lane center-lines for map-relative localization in lane-level maps. However, the availability and detection range in front of the vehicle has to be significantly improved in order to be used for automated lateral control and scene understanding (context modeling). Since common model-based approaches, like moving clothoids or arches, fail in urban environments due to unpredictable and sudden changes of the lane's curvature, we propose a different approach with less model assumptions. Each lane hypothesis is represented by a list of support points (polyline). These support points are distributed in equidistant steps along the lane and assumed to be fixed in place. As the extraction of the lane center lines from the grid also works based on a reference point fixed in place, the tracked positions directly correspond to incoming support points.

This allows a simple association of new hypotheses with existing ones and thus to track each of them in a Kalman-filter similar to the approach in [30]. Newly incoming polylines are associated to existing ones if possible. The support points are then updated or added to the existing hypotheses. If a polyline cannot be associated to an existing one, a new hypothesis is created. Support points which are further away from the host vehicle than a certain distance are deleted in order to keep the number of tracked points limited. First results are shown in Figure 9.



(a) Tracking of opening lanes at inner-city intersections.

(b) Tracking of non-parallel lanes.

(c) Tracking of non-parallel lanes and opening lanes at inner-city intersections.

Fig. 9. Benefit of the subsequent tracking: The results of the lane tracking (red) improve the availability of the lane center lines (detections in white) significantly. The approach copes with a changing number of lanes as well as non-parallel lanes up to a certain extent.

VII. Conclusion and Outlook

In this paper, we have presented the latest contributions of the project *Stadtpilot* toward a comprehensive perception-driven approach of urban environment modeling. We proposed additions to dynamic classification algorithms which is able to deal with with stop-and-go traffic. An approach of using explicit free-space information was shown and first results of a grid-based road tracking algorithm were presented. The introduced dynamic classification and object tracking module are already used in public traffic, the road extraction and tracking algorithms are currently evaluated at the test track. The presented algorithms run in real-time on a PC platform (Intel i7-4770), in relation to the sensors' update rate of 10 Hz. After the feature generation, the subsequent filtering algorithms can be applied in parallel in order to reduce the systems' latency. Typical execution times do not exceed 60 ms.

Based on the presented algorithms, we have been able to further extend our capabilities to model the vehicle's environment via on-board sensors only. Nevertheless, still a lot of work has to be done for a complete perception and modeling of relevant aspects. Future research will address the extraction of features to determine topological properties of lanes and to model the context between the detected elements. This will further enhance our perception-driven approach of an environment modeling in the urban domain.

Acknowledgment

The authors wish to thank the Stadtpilot team, especially Christopher Karberg, for their support.

References

[1] J. M. Wille, F. Saust, and M. Maurer, "Stadtpilot: Driving autonomously on Braunschweig's inner ring road," in *2010 IEEE Intelligent Vehicles Symposium (IV)*, San Diego, 2010, pp. 506–511.

[2] T. Nothdurft, P. Hecker, S. Ohl, F. Saust, M. Maurer, A. Reschka, and J. R. Böhmer, "Stadtpilot: First fully autonomous test drives in urban traffic," in *2011 IEEE International Conference on Intelligent Transportation Systems (ITSC)*, Washington DC, 2011, pp. 919–924.

[3] R. Matthaei, A. Reschka, G. Bagschik, M. Escher, T. Menzel, J. Rieken, T. Scheide, F. Schuldt, S. Ulbrich, J. T. Wendler, P. Hecker, and M. Maurer, "Das Projekt Stadtpilot - Automatisiertes Fahren an der TU Braunschweig (The Project Stadtpilot - Automated driving at the TU Braunschweig)," *Zeitschrift für die gesamte Wertschöpfungskette Automobilwirtschaft*, vol. 01/2015, no. 1, pp. 12–23, 2015.

[4] R. Matthaei, B. Lichte, and M. Maurer, "Robust grid-based road detection for ADAS and autonomous vehicles in urban environments," in *2013 16th International Conference on Information Fusion (FUSION)*, Istanbul, 2013, pp. 938–944.

[5] R. Matthaei, G. Bagschik, J. Rieken, and M. Maurer, "Stationary Urban Environment Modeling using Multi-Layer-Grids," in *17th International Conference on Information Fusion (FUSION)*, Salamanca, 2014, pp. 1–7.

[6] J. Rieken, R. Matthaei, and M. Maurer, "Benefits of Using Explicit Ground-Plane Information for Grid-based Urban Environment Modeling," in *18th International Conference on Information Fusion (FUSION)*, Washington, USA, 2015.

[7] R. Matthaei and M. Maurer, "Autonomous driving – a top-down-approach," *at - Automatisierungstechnik*, vol. 63, no. 3, pp. 155–167, 2015.

[8] R. Matthaei, A. Reschka, J. Rieken, F. Dierkes, S. Ulbrich, T. Winkle, and M. Maurer, "Autonomes Fahren (Autonomous driving)," in *Handbuch Fahrerassistenzsysteme (Handbook of Driver Assistance Systems)*, 3rd ed., H. Winner, S. Hakuli, F. Lotz, and C. Singer, Eds. Wiesbaden: Springer-Vieweg, 2015, english version accepted to appear.

[9] A. Geiger, M. Lauer, F. Moosmann, B. Ranft, 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.

[10] J. Ziegler, P. Bender, M. Schreiber, H. Lategahn, T. Strauss, C. Stiller, T. Dang, U. Franke, N. Appenrodt, C. Keller, E. Kaus, R. Herrtwich, C. Rabe, D. Pfeiffer, F. Lindner, F. Stein, F. Erbs, M. Enzweiler, C. Knöppel, J. Hipp, M. Haueis, M. Trepte, C. Brenk, A. Tamke, M. Ghanaat, M. Braun, A. Joos, H. Fritz, H. Mock, M. Hein, and E. Zeeb, "Making Bertha Drive - An Autonomous Journey on a Historic Route," *IEEE Intelligent Transportation Systems Magazine*, vol. 6, no. 2, pp. 8–20, 2014.

[11] J.-M. Wille, "Manöverübergreifende autonome Fahrzeugführung in innerstädtischen Szenarien am Beispiel des Stadtpilotprojekts," Ph.D. dissertation, Technische Universität Braunschweig, 2012.

[12] J. Leonard, J. How, S. Teller, M. Berger, M. Campbell, G. Fiore, L. Fletcher, E. Frazzoli, A. Huang, S. Karaman, O. Koch, Y. Kuwata, D. Moore, E. Olson, S. Peters, J. Teo, R. Truax, M. Walter, D. Barrett, A. Epstein, K. Maheloni, K. Moyer, T. Jones, R. Buckley, M. Antone, R. Galejs, S. Krishnamurthy, and J. Williams, "A Perception-Driven Autonomous Urban Vehicle," *Journal of Field Robotics*, vol. 1, no. 48, 2008.

[13] A. Broggi, M. Buzzoni, S. Debattisti, P. Grisleri, M. Laghi, P. Medici, and P. Versari, "Extensive Tests of Autonomous Driving Technologies," *IEEE Transactions on Intelligent Transportation Systems*, vol. 14, no. 3, pp. 1403–1415, 2013.

[14] A. Mueller, M. Himmelsbach, T. Luettel, F. v Hundelshausen, and H.-J. Wuensche, "GIS-based topological robot localization through LIDAR crossroad detection," in *2011 14th International IEEE Conference on Intelligent Transportation Systems (ITSC)*, Washington, 2011, pp. 2001–2008.

[15] M. Manz, "Modellbasierte visuelle Wahrnehmung zur autonomen Fahrzeugführung," Ph.D. dissertation, Technische Universität München, 2013.

[16] M. Himmelsbach and A. Müller, "LIDAR-based 3D object perception," in *Proceedings of 1st International Workshop on Cognition for Technical Systems*, Munich, 2008.

[17] J. Choi, S. Ulbrich, B. Lichte, and M. Maurer, "Multi-Target Tracking using a 3D-Lidar sensor for autonomous vehicles," in *2013 16th International IEEE Conference on Intelligent Transportation Systems (ITSC)*, The Hague, 2013, pp. 881–886.

[18] J. Choi and M. Maurer, "Hybrid map-based SLAM with Rao-Blackwellized particle filters," in *2014 17th International Conference on Information Fusion (FUSION 2014)*, Salamanca, 2014, pp. 1–6.

[19] J. S. Levinson, "Automatic laser calibration, mapping and localization for autonomous vehicles," Ph.D. dissertation, Stanford University, 2011.

[20] H. Badino, U. Franke, and D. Pfeiffer, "The stixel world-a compact medium level representation of the 3d-world," *Pattern Recognition*, pp. 1–10, 2009.

[21] K. Dietmayer, J. Sparbert, and D. Streller, "Model based object classification and object tracking in traffic scenes from range images," in *2001 IEEE Intelligent Vehicles Symposium (IV)*, Tokyo, 2001, pp. 1–6.

[22] R. Matthaei, H. Dyckmanns, M. Maurer, and B. Lichte, "Consistency-based motion classification for laser sensors dealing with cross traffic in urban environments," in *2011 IEEE Intelligent Vehicles Symposium (IV)*, Baden-Baden, 2011, pp. 1136–1141.

[23] R. Matthaei, H. Dyckmanns, B. Lichte, and M. Maurer, "Motion classification for cross traffic in urban environments using laser and radar," in *2011 Proceedings of the 14th International Conference on Information Fusion (FUSION)*, Chicago, 2011, pp. 1–8.

[24] U. Ramer, "An iterative procedure for the polygonal approximation of plane curves," *Computer Graphics and Image Processing*, vol. 1, no. 3, pp. 244 – 256, 1972.

[25] R. J. Adcock, "A problem in least squares," *The Analyst*, vol. 5, no. 2, pp. 53–54, 1878.

[26] K. Schueler, T. Weiherer, E. Bouzouraa, and U. Hofmann, "360 Degree multi sensor fusion for static and dynamic obstacles," in *2012 IEEE Intelligent Vehicles Symposium (IV)*, Alcala de Henares, 2012, pp. 692–697.

[27] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics (Intelligent Robotics and Autonomous Agents)*, 3rd ed. The MIT press, 2005.

[28] R. Matthaei, G. Bagschik, and M. Maurer, "Map-relative localization in lane-level maps for ADAS and autonomous driving," in *2014 IEEE Intelligent Vehicles Symposium (IV)*, Dearborn, 2014, pp. 49–55.

[29] R. Matthaei, "Wahrnehmungsgestützte Lokalisierung in fahrstreifengenauen Karten für Assistenzsysteme und automatisches Fahren in urbaner Umgebung," Ph.D. dissertation, Technische Universität Braunschweig, 2015, (accepted to appear).

[30] M. Schreier and V. Willert, "Robust free space detection in occupancy grid maps by methods of image analysis and dynamic B-Spline contour tracking," in *15th International IEEE Conference on Intelligent Transportation Systems (ITSC)*, Washington DC, 2012, pp. 514–521.