Graph-Based Context Representation, Environment Modeling and Information Aggregation for Automated Driving

Simon Ulbrich, Tobias Nothdurft, Markus Maurer, Peter Hecker

Abstract— **The *Stadtpilot* project aims at fully automated driving on Braunschweig's inner city ring road. The TU Braunschweig's research vehicle *Leonie* is one of the first vehicles having the ability of fully automated driving in real urban traffic scenarios. In this paper, we present our approaches for context representation and environment modeling for automated driving. The demonstrated approach allows to provide a simple and yet universal information storage layer for the development of complex driving applications. Moreover, we present our approach for aggregating and fusing information between dynamic traffic objects detected by the sensor systems and a-priori map information.**

I. INTRODUCTION

During the last 20 years the driving abilities of automated vehicles have progressed rapidly. Particularly the DARPA Grand Challenges put automated driving into the focus of many research teams around the world. After a successful participation in the DARPA Urban Challenge in 2007 [1], the TU Braunschweig continued its effort in the automated driving field with the *Stadtpilot* project. The goal of the *Stadtpilot* project is to drive fully automated on Braunschweig's inner city circular ring road. First accomplishments of fully automated driving in heavy inner city traffic have successfully been demonstrated to the public already [2]. In this paper, we present our approaches for context representation and environment modeling. Our goal is to combine all kinds of information from environment perception modules and an a-priori map into an aggregated graph representation to serve following driving functions as an information base for behavior decision making. So far our team used a very simple context model representing the ego vehicle's environment by a two-parallel-lanes model only consisting of the ego lane and a neighbor lane, either on the left or on the right. Our former context model was not able to provide detailed information about intersections, sidewalks, bike lanes and similar complex situations. Although being very simple it was sufficient for driving the show case maneuver in [3] with a -to some extend- limited situation awareness. Nevertheless, for more foresight in fully automated driving, it is necessary to facilitate a more sophisticated context model. E.g., if a crosswalk is represented in the context model, tactical driving behavior planning algorithms can consider a pedestrian, which is about to cross the road at the crosswalk, even if he is still on the sidewalk and has not yet entered the road.

S. Ulbrich and M. Maurer are with the Institute of Control Engineering, Technische Universität Braunschweig, Hans-Sommer-Str. 66, 38106 Braunschweig, Germany {ulbrich, maurer}@ifr.ing.tu-bs.de

T. Nothdurft and P. Hecker are with the Institute of Flight Guidance, Technische Universität Braunschweig, Hermann-Blenk-Str. 27, 38108 Braunschweig, Germany {t.nothdurft, p.hecker}@tu-bs.de

The main contribution of this paper is to present a more sophisticated context model for complex, urban driving situations as in figure 1 or 9 compared to those typically being used in the assisted driving field for less complex traffic situations. Moreover, we present an efficient algorithm for dynamic traffic object to a-priori map matching. It allows matching objects at complex intersections accurately and works well for narrow/off-centered objects like bicycles.

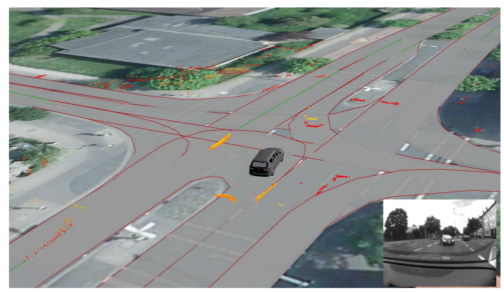

Fig. 1. Abstract graph-based context representation with approximated lane boundaries [red/green], Delaunay triangles [grey] and sensor object points

This paper is structured as follows: First of all, we introduce the reader to the problem of context modeling. We present different general approaches for context modeling and point out different relationships between context entities and layers of information. In section III, we discuss a graph based context representation approach to accommodate any of those information and relationships. Section IV shows our approach for information fusion and aggregation. Information of dynamic objects and lane segment locations are aggregated into abstract information like the distance to the next vehicle on the ego lane. Section V provides an evaluation of the algorithms being presented in this paper. The evaluation is particularly focusing on the information aggregation step since the other steps are hard to evaluate quantitatively, except from proving their general feasibility for complex environments.

II. BACKGROUND

According to Brown [4], context is "a combination of elements of the user's environment which the computer knows about." In the field of automated driving the word "user" needs to be replaced with the automated vehicle itself. Different approaches have been proposed for context modeling, see [5], [6] and [7] for a detailed discussion.

Several concepts have been used for object based context representation. Most of them are not clearly distinguishable

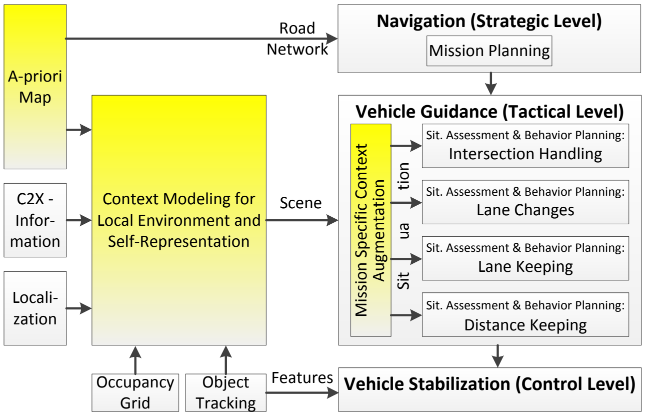

Fig. 2. Context model as a part of a simplified system architecture, based on Matthaei [8]. Modules incorporating the described context model in yellow

and share certain aspects and features with other methods. Among them are logic based and world modeling approaches.

A general approach is to use (first order) logic as a language to describe the information about the environment presented, e.g., by Russell et al. [9]. Here the information is stored in a set of first order logic clauses. This allows reasoning in the domain, but typically tends to limit the possible domain complexity.

A second approach is sometimes referred as a "world model" approach. It is similar to a graph representation of the automated vehicle's environment. Concepts (abstract or physical objects) are linked by relationships. Among those relationships there could be geo-spatial relationships like proximity or ontological relationships like "A is a subtype of B." Gheta et al. [10] provide more details. A world model can be seen as an instantiated (or: *grounded*) semantic network. All in all, it aligns very well with the object oriented programming paradigm in C++.

Maurer [11] and Rieder [12] (both from University of the German Federal Armed Forces, Munich) use a *scene tree* as a representation of context for those aspects discussed in this paper. Every node in a *scene tree* is a physical entity in the ego vehicle's environment or part of the ego vehicle, e.g., a sensor. Every branch in a *scene tree* represents a geo-spacial relationship and can be modeled by a homogeneous transformation. If another vehicle is detected by several redundant sensors, e.g., by a vehicle's sensor and a camera mounted on a traffic light, the *scene tree* no longer stays a tree but rather turns into a *scene graph* because every node might have several parent nodes. If a *scene graph* also contains semantic relationships between entities and the meaning of those entities for the ego vehicle like mentioned by Maurer [13, p. 63] the line between a pure geo-spatial *scene tree* and other world modeling approaches is blurred.

A formal representation of the crucial domain knowledge is also called an *ontology*. Ontologies are used to specify concepts and interrelations. "An ontology is a specification of a conceptualization. [. . .] A conceptualization is an abstract, simplified view of the world that we wish to represent for some purpose" [14]. Feld and Müller [15] developed an ontology for advanced driver assistance systems. They focused particularly on human-machine-interfacing and inter-vehicle information exchange. Vacek [16] developed an ontology for an automated vehicle. He based his ontology on the Ontology Web Language (OWL) standard.

Knaup and Homeier [17] developed a graph based environment model for driver assistance systems for intersections. They based their infrastructure model on standard maps for navigation systems. Papp et al. [18] presented a "local dynamic map" as an object oriented context model serving as an abstraction layer between data acquisition and high-level behavioral functions for cooperation between different vehicles. Schlenoff et al. [19] proposed a central "knowledge base" and presented a multi-resolution approach for infrastructure information representation and evaluated an object prediction algorithm being based on that with some simulated data.

Galindo et al. [20] and Oberlander et al. [21] grouped context modeling approaches, in particular mapping approaches into metrical, topological and semantic representations. A metric representation subsumes aspects of measurable distances, lengths and sizes and is closely related to the physical measurements being made by the sensor systems. A plain scene tree is a commonly used method for representing such metrical relationships. However, more abstract context representations exist. Among them are topological and semantic levels. The topological level describes the connections between entities, e.g., like a road network of a city in a traditional road map or even how lane segments are linked together in an intersection. The semantic describes the relationships between entities. Being connected is one of them. However, it subsumes more abstract relationships like *"Is On"* or even *"Will be on"*. Regele [22] developed a hierarchical world model for complex traffic and infrastructure scenarios, in which lane segments are linked by conflict-, neighboring-, opposing- or successor-relationships.

A central aspect for infrastructure context modeling is the coordinate system being used. Lane detection and tracking algorithms for driving on highways are typically performed in an ego-relative world and use some kind of iterative clothoid model or parabolic function fitting, e.g., with Kalman filters, see McCall et al. [23] for a detailed survey. These context modeling approaches assume small heading angles between the automated vehicle and the lane being driven on, low curvature rates of the road network and a relatively short model fitting span. While this holds true for most driving situations, it is a severe limitation for dense urban areas with complex road geometries and intersections. To address these limitations, Gregor et al. [24], [25, p. 51] introduced a second, not ego-relative but geo-stationary context model for representing an a-priori infrastructure map as a part of the *scene tree*. Manz et al. [26] developed a lane tracking and infrastructure context modeling framework in a geo-stationary world for rural terrain. In both publications, each arm of an intersection is modeled by a geo-stationary clothoid segment. In Manz et al. there was no need to model each intersection arm by a series of several clothoid segments (spline) because the lane segment length that was to be modeled was rather short as it was limited by the rather short field of view of a lane tracking camera system.

Reyher [27, pp. 93] used an arc to model a lane segment and used the distance to the lane segment center line and an estimated, but over its length constant, lane width from an online lane tracking for an object-to-lane-matching. Rieder [12, p. 91] used the distance to the clothoid modeled lane segment center line and a linearly increasing/decreasing lane width for an object-to-lane-matching. In both publications, we did not find any description of an object-to-a-priori-map-matching algorithm. Homeier and Wolf [28] presented an approach for object-to-a-priori-map-matching. They assumed the width of a lane segment to be constant over its length. Hence, they based their object to lane association method on the distance to the center of a lane segment.

III. GRAPH BASED INFORMATION REPRESENTATION

In order to make coherent decisions, information needs to be collected and aggregated as in figure 2. This section presents our approach to represent static infrastructure information, dynamic environment information and the automated vehicle's ego information. Combining all those information results in a scene representation of the current local environment and a self representation for the state of the automated vehicle itself. However, for automated driving such a mission-independent scene needs to be augmented by mission specific context information (cp. figure 2) like the planned route or planned, future actions, e.g., lane changes or parking maneuvers.

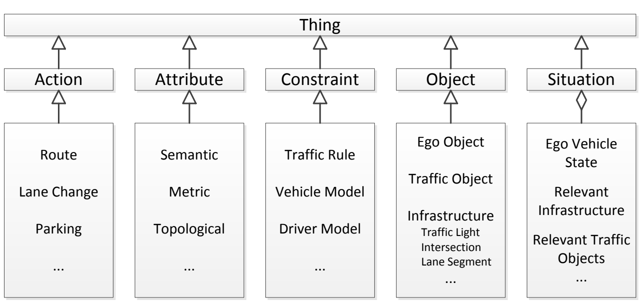

Fig. 3. Proposed ontology for an automated vehicle's context model

Figure 3 shows the developed ontology. Similar to the OWL standard, we use a central root element "thing" with several child objects for physical objects like the ego vehicle itself, other traffic objects and all kinds of infrastructure objects. Moreover, we use constraints to represent, e.g., traffic rules or assumed dynamic models and general (logical) attributes on different hierarchical levels. Last of all, it is possible to incorporate situation specific information in the context model, to represent information about the planned route and planned actions. As introduced in the last section, we propose a hybrid approach of different, hierarchical information layers as shown in figure 4. On a high level, only topological attributes and relationships are used, on lower levels semantic and metric information will be used for specific driving tasks.

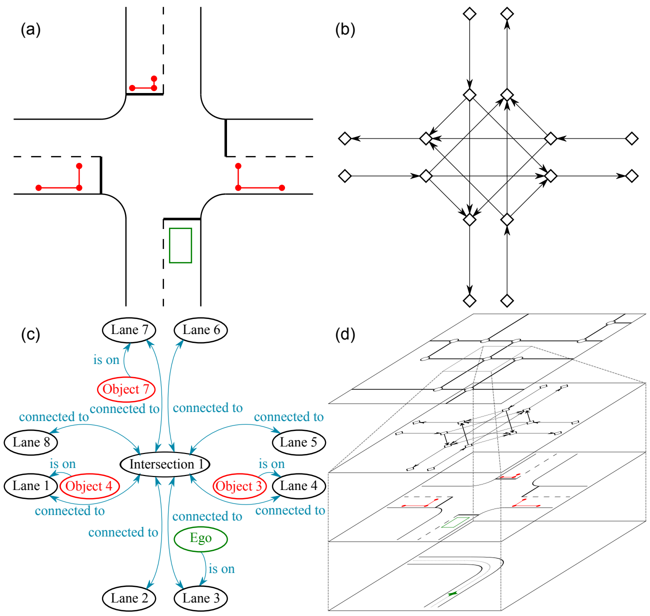

Fig. 4. Metrical, topological and semantic representation of information in the proposed context model. (a) indicates geo-spatial distances of reflectance points and lane markings. (b) shows topological relationships between lane segments at an intersection. (c) illustrates a semantic layer of information. (d) illustrates the overall hybrid context model containing all the individual information layers. The semantic layer is hidden in (d) for readability

Figure 5 illustrates our implementation of our world model context representation approach. It shows several different kinds of information nodes. Among the nodes for infrastructure representation are lane segments itself, lane boundaries linking two neighbor lane segments in the traversal direction toward each other and waypoints linking lane segments longitudinally together. Each object may have additional attributes to provide a more accurate description of the object. These attributes could contain information, e.g., if a lane boundary is allowed to be crossed, the speed limit of a lane segment, etc. Likewise, occupancy grid information is extracted and linked to lane segments with an *"Is Drivable"*-attribute. The arrows indicate relationships among the objects/nodes; they allow to traverse to any node from every other node. We follow the idea of Manz et al. [26] and use a similar geo-stationary approach for a-priori infrastructure context modeling. However, since our context model is not necessarily limited to the short field of view of a lane tracking camera system, we use a series of multiple short lane segments (spline) instead of a single clothoid representation. Instead of doing the rather complex approximation of a clothoid's integral, we use a simple polynomial spline function for lane modeling.

IV. INFORMATION AGGREGATION

In this section, we describe our approach for information aggregation. Information aggregation subsumes all aspects to compile sensor data, infrastructure information or any other

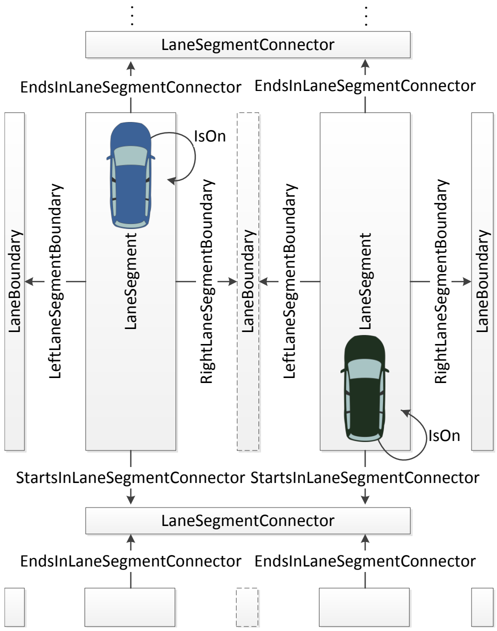

Fig. 5. Example of the implementation for a context model of a simple traffic situation on a straight road of two parallel lane segments

kind of information into the "context model." More specifically, this includes integrating static infrastructure information like traffic lights, or conflicts between lane segments and combining dynamic environment information from the sensor systems with the a-priori infrastructure information.

In this paper, the authors focus on combining dynamic environment information and static infrastructure information. Dynamic environment information is essentially a vector of points representing a dynamic traffic object (other cars, cyclists, pedestrians, etc.). These need to be located on the lane segments. For simple traffic situations, e.g., on a highway, this essentially boils down to locating those object related points on the ego or neighbor lane segments' polygons. However, at complex intersections this task is far more complicated, because thousands of those point-in-lane-segment operations are needed per second. Parked objects at the side need to be accurately distinguished from objects on the lane, while narrow objects outside the center of the lane like bicycles must not be missed.

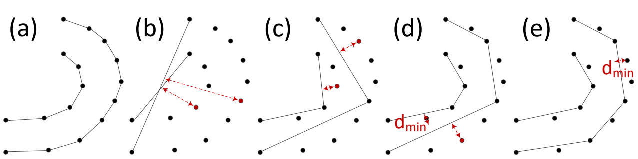

Fig. 6. Visualization of the spline approximation for lane boundaries

To achieve this data processing rate, some computational ploys need to be exploited: First of all the lane segment polygons are approximated by a Ramer-Douglas-Peucker (DP) approximation [29], [30] as illustrated in figure 6. The original piecewise linear polygon of the lane segment's boundaries is separated into the polygon of the left and right lane segment boundary (a). Starting from the two maximally separated vertices (b), the additional vertices are iteratively added to the approximated polygon (c)-(e) until the approximation is closer to the initial polygon than a threshold d_{min}. $d_{min} = 0.1$m was used for our implementation. It is a viable compromise between geometric accuracy and computation speed.

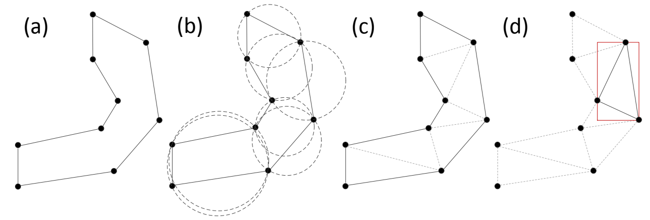

Fig. 7. (a) Approximated lane segment, (b) circumcircles of the triangles that fulfill the properties for a Delaunay triangulation, (c) Delaunay triangulation, (d) bounding box [red] and minimum area bounding rectangle [green] of the black triangle

As a next step, bounding boxes and minimal area bounding rectangles are calculated for every lane segment as a whole. It helps to avoid wasting time by trying to locate a point on a lane segment physically far away from the point's location. As a last step, the Delaunay triangulation [31] of every lane segment is calculated to decomposition the point-in-polygon problem into faster to solve point-in-triangle problems. The Delaunay triangulation is a commonly used method in computational geometry to decomposition an abstract shape into atomic, simple shapes, called simplices. For a set of points $P \in \mathbb{R}^2$ these simplices are triangles. The Delaunay triangulation $DT(P)$ is a triangulation that fulfills the property that the circumcircle of each triangle contains no other points of the input set of points P. $DT(P)$ maximizes the minimum angle, of all the triangles' angles, in the triangulation. This tends to avoid "skinny" (=long and thin) triangles. Figure 7 illustrates the Delaunay triangulation for the lane segment from figure 6. Figure 7 shows the Delaunay triangulation, bounding boxes and minimum area bounding rectangles for a part of the road infrastructure in Braunschweig. Figure 8 illustrates the four steps of the point-in-lane-segment algorithm. First of all, it is checked if a point is within the light blue bounding box of the lane segment. E.g., in case of P_1 it is not necessary to proceed with further analysis. Nevertheless, further analysis is necessary for objects at the positions P_2 till P_4. As a next step, it is checked if the points to be analyzed are located inside of the green minimum area two dimensional bounding rectangle (P_3, P_4). Many lane segments are almost rectangular, thus this is a very good approximation, although it is computationally far more expensive than a simple comparison of the x and y coordinates for the point-in-bounding-box comparison. As a third step, it is checked if a particular point is in any of the Delaunay triangles' bounding boxes of the lane segment. This holds true for point P_4 but not for point P_3 for the example in figure 8. For those points that are in one of the triangles' bounding boxes, it is finally checked if the point is also inside of the particular Delaunay triangle. If this holds true, the point is proven to be on the lane segment.

Further speedup could be achieved by utilizing the fact that an object will only marginally move from the lane segments

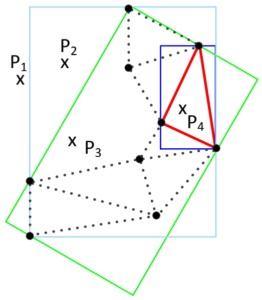

Fig. 8. Illustration of the steps for the point-on-lane-segment-algorithm. Lane segment and its Delaunay triangulation [dotted lines], highlighted Delaunay triangle [red], bounding box of that triangle [blue], two dimensional minimal area lane segment bounding rectangle [green] and lane segment bounding box [light blue]

it was located on in the last iteration. However, since it is already fast enough no additional efforts were made to implement this idea.

V. EVALUATION

In this section, we evaluate the approaches being presented so far. A direct quantitative evaluation of the context model itself is hard to achieve. As a benchmark, figure 9 shows our context model representing one of the most complex intersections in Braunschweig. Table I presents a qualitative evaluation. All kinds of information sources, e.g. traffic lights with C2X-communication, or onboard sensor systems can be integrated. The context model holds a detailed infrastructure model and the algorithm in section IV allows to combine dynamic traffic information and infrastructure information. The skill and state representation for the ego-vehicle provides basic vehicle information but is not yet able to provide an abstract skill and ability model. The graph structure allows to represent several, even contradicting semantical, topological or metrical relationships between information entities. Information quality and integrity can be represented by attributes. Yet, a consistency checking and monitoring still remains to be integrated. By avoiding the need for a sophisticated database engine, logic solver or semantic reasoner, information access is only limited by the RAM access delays and thus sufficiently fast for real time information access.

TABLE I

QUALITATIVE EVALUATION FOR CONTEXT MODELING APPROACH

Requirements	Assessment
Handling of heterogeneous information sources	++
Representation of behavior relevant information	
Infrastructure-related	++
Dynamic traffic environment-related	+
Ego-Vehicle-related	+
Representation of context relationships between information nodes	++
Context information quality and integrity representation	+
Context consistency representation and monitoring	o
Real time information access	++

We did a more quantitative evaluation for our algorithm to locate objects on lane segments for automated driving on a stretch of Braunschweig's inner city ring road in figure

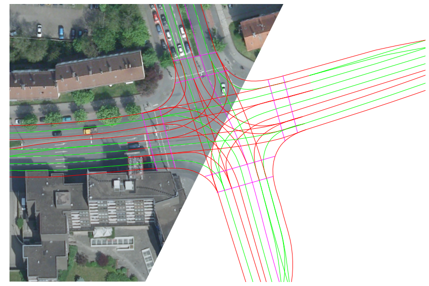

Fig. 9. Lane network for a complex intersection

10. The chosen approach is capable to be used for real-time driving applications. The road/lane segment network graph for Braunschweig in its current complexity consists of 1079 lane segments and fits easily into a standard computer's random access memory. All in all, the computational ploys in section IV help to reduce the processing time by a lot. We measured the necessary time for an average point-in-lane-segment operation based on 230 million point-in-lane-segment operations on *Leonie's* Core2Quad-system from 2008. On average, such a point-in-lane-segment operation takes $0.126\,\mu s$. This translates into on average to about $40\,ms$ processing time for one update cycle. Hence, sufficiently fast for a cycle time of $100\,ms$, being decided as a tradeoff between the required update rate for the context representation and the speed the automated vehicle's environment typically changes. For a typical urban environment the sensor data fusion in its state at the time of writing will generate at most up to about 50 objects around the automated vehicle. On average, they consist of about 6 points and they are sent with an update rate of about $10\,Hz$. Hence, this proved to be enough even for extremely complex road intersection situations.

Fig. 10. Stretch of Braunschweig's inner city ring road used for the evaluation

A central aspect for locating objects on lane segments is the fact that sensor systems produce several object hypotheses for objects being part of the static infrastructure like buildings, trees, street lamps, or fences. Figure 11 shows that only a small fraction ($< 7\%$) of all tracked object hypotheses are actually on the ego lane segment/route and thus viable to be handled with a state-of-the-art lane tracking and object-to-lane-matching algorithm (at least if not being narrow/off-centered like bicycles). More specifically it shows the lateral distance of the center of the ego lane/planned route to object's point being closest to that center line. With an update cycle of $100\,ms$ and a setup of two Ibeo Alasca

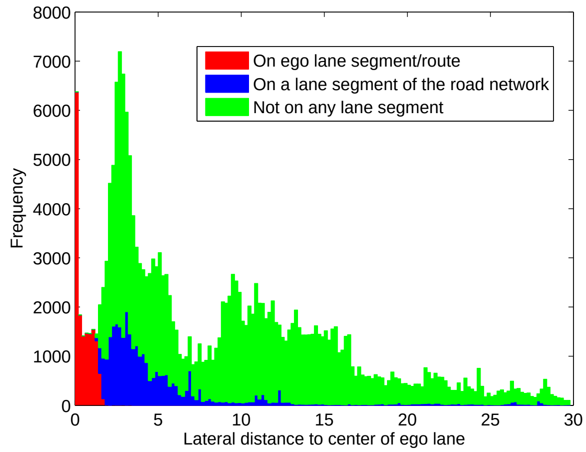

Fig. 11. Objects being on the ego lane segments or other lane segments as a function of the minimal lateral distance to the route constituting lane segments

XT LIDAR sensors to the front and one Ibeo Lux LIDAR sensor to the rear [2], we had $2.37 \cdot 10^5$ object hypotheses to be located on lane segments for the track in figure 10. 78% of the object hypotheses are found in locations not being part of the road network. Particularly many objects are found in distances of about 3 meters. Many of those are parked cars, fences and trees. Most objects with a lateral distance offset of less than 1.5 meters are located on the ego lane. They are particularly relevant for lateral distance keeping and adaptive cruise control driving applications. However, for lane changes and other more complex driving maneuvers, it is also important to handle the blue marked objects that are also on the road network but not on the ego lane.

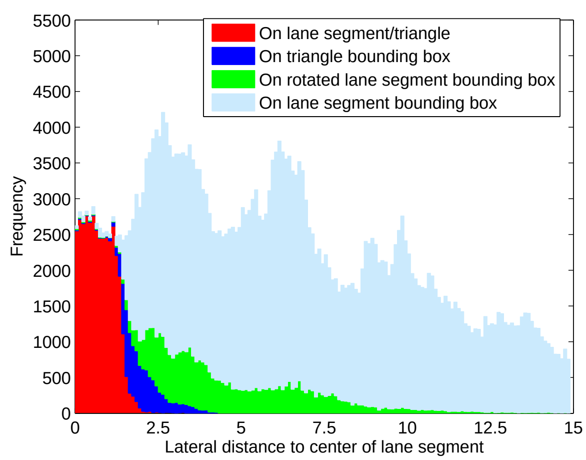

Fig. 12. Number of points for the different steps of the point-on-lane-segment algorithm as a function of the lateral distance to the center line of the specific lane segment

Figure 12 illustrates the lateral distance of sensor object contour points not as a function of the distance to the center of the ego lane but with the distance to the center of any lane segment. It shows the fraction of sensor object points being assumed to be located on the lane segment in the different steps of the algorithm. As shown in figure 8, the algorithm consists of four steps. The cumulated frequency of all four curves in figure 12 represents the fraction of sensor object contour points being inside of the bounding box of any of the road network's lane segments. The light blue curve shows the fraction of objects being inside of the lane segments' bounding boxes but not being inside of any of the lane segments rotated two-dimensional minimal area bounding rectangles. Accordingly, the green curve indicates the fraction of points being part of a lane segment's rotated bounding box but not being part of any of the lane segment's Delaunay triangles' bounding boxes. The red curve finally indicates the fraction of objects being found to be inside of at least one lane segments' Delaunay triangles and thus being on the lane segment. All in all, we based our analysis on a representative part of the track in figure 10 with $2.2 \cdot 10^8$ point-in-(any)-lane-segment operations. In 99.81% of the situations the point was not even in the lane segment bounding box. Luckily, the point in bounding box test is a computationally simple comparison of two coordinates and thus extremely cheap. However, even for the remaining $4.2 \cdot 10^5$ points, which are in fact in a lane segments' bounding box, the fraction of the points being part of the red curve in figure 12 is very small. Figure 13 shows the fraction of the points being only under the red curve (6% being inside of a lane segment), under the red curve and the blue curve (8% being at least in the bounding box of one of the lane segments' triangles and maybe even inside of the lane segment), etc.

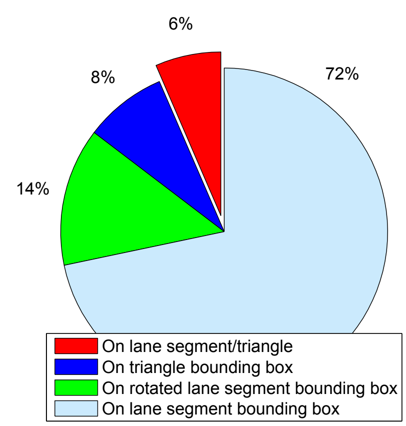

Fig. 13. Fraction of points that still have to be analyzed for the different steps of the algorithm

VI. CONCLUSIONS

In this paper we presented our approach for context representation, environment modeling and information aggregation for automated driving in urban traffic scenarios. The presented approach allows fast information access and is yet able to cope with the complexity of environment information being faced in automated driving. Our algorithms for information aggregation of dynamic traffic objects and a-priori map information are able to differentiate relevant traffic objects on the road precisely from other objects at the shoulders of the road like trees, parked cars, etc. By using the exact lane contour given by a lane's left and right lane boundary, our algorithm is also able to handle narrow objects like motorbikes, bicycles or pedestrians being rather close to the boundary of a lane than to its center line. Moreover, it works well with oddly shaped lane segments to be found in dense urban scenarios. It could even be used for matching objects to oddly shaped lane segments directly obtained from sensor based reflectance grid maps. The context model itself is already relatively sophisticated for the application of automated driving. However, room for improvement is in the field of consistency checking and integrity monitoring. So far our algorithms will blindly integrate all kinds of information, no matter whether they align well with existing information or being entirely contradictory. For providing automatic driving functions it is essential to have some kind of reliability measures to allow graceful degradation to a lower degree of automation or even taking a human driver back into the loop. First steps have already been undertaken in that direction [32], but further research needs to be conducted on finding appropriate measures and degradation strategies.

ACKNOWLEDGMENT

The authors thank the *Stadtpilot* team, namely Bernd Lichte, Falko Saust, Andreas Reschka, Sebastian Ohl, Martin Escher, Jaebum Choi, Jens Rieken, Fabian Schuldt, Richard Matthaei and Sven Böhme for their support in improving hard- and software of the research vehicle *Leonie* and spending days and weeks on the test track and public roads to tune the algorithms until they reached the status quo.

REFERENCES

[1] F. Rauskolb, K. Berger, C. Lipski, M. Magnor, K. Cornelsen, J. Effertz, T. Form, F. Graefe, S. Ohl, W. Schumacher, J. M. Wille, P. Hecker, T. Nothdurft, M. Doering, K. Homeier, J. Morgenroth, L. Wolf, C. Basarke, C. Berger, T. Gülke, F. Klose, and B. Rumpe, "Caroline: An autonomously driving vehicle for urban environments," *Journal of Field Robotics*, vol. 25, no. 9, pp. 674–724, 2008.

[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 *14th International IEEE Conference on Intelligent Transportation Systems (ITSC)*, Washington, USA, 2011, pp. 919–924.

[3] F. Saust, O. Bley, R. Kutzner, J. M. Wille, B. Friedrich, and M. Maurer, "Exploitability of vehicle related sensor data in cooperative systems," in *13th International IEEE Conference on Intelligent Transportation Systems, 2010 IEEE*, Funchal, Portugal, 2010, pp. 1724–1729.

[4] P. J. Brown, "The Stick-e Document: a Framework for Creating Context-aware Applications," in *Proceedings of Electronic Publishing'96*, Palo Alto, USA, June 1996, pp. 259–272.

[5] R. Chatila and J. Laumond, "Position referencing and consistent world modeling for mobile robots," in *Proceedings of IEEE International Conference on Robotics and Automation*, vol. 2, St. Louis, USA, March 1985, pp. 138–145.

[6] C. Becker and F. Dürr, "On location models for ubiquitous computing," *Personal and Ubiquitous Computing*, vol. 9, pp. 20–31, 2005.

[7] T. Strang and C. Linnhoff-Popien, "A context modeling survey," in *Workshop on Advanced Context Modelling, Reasoning and Management, 6th International Conference on Ubiquitous Computing*, Nottingham, England, 2004, pp. 1–8.

[8] R. Matthaei and M. Maurer, "System Architecture for Smart Vehicles in the Context of Automated Map Updates," in *Fahrzeugsystemtechnik-Workshop*, 2014, accepted to appear.

[9] S. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach*, 3rd ed. Prentice Hall, 2009.

[10] I. Gheţa, M. Heizmann, A. Belkin, and J. Beyerer, "World modeling for autonomous systems," in *Proceedings of the 33rd Annual German Conference on AI*, Karlsruhe, Germany, 2010, pp. 176–183.

[11] M. Maurer, "EMS-vision: knowledge representation for flexible automation of land vehicles," in *Proceedings of the IEEE Intelligent Vehicles Symposium*, Dearborn, USA, 2000, pp. 575–580.

[12] A. Rieder, "Fahrzeuge Sehen - Multisensorielle Fahrzeugerkennung in einem verteilten Rechnersystem für autonome Fahrzeuge," Dissertation, Univ. der Bundeswehr München, 2000.

[13] M. Maurer, "Flexible Automatisierung von Straßenfahrzeugen mit Rechnersehen," Dissertation, Univ. der Bundeswehr München, 2000.

[14] T. R. Gruber, "Toward principles for the design of ontologies used for knowledge sharing," *International Journal of Human-Computer Studies*, vol. 43, no. 5, pp. 907–928, 1995.

[15] M. Feld and C. Müller, "The automotive ontology: Managing knowledge inside the vehicle and sharing it between cars," in *Proceedings of the 3rd International Conference on Automotive User Interfaces and Interactive Vehicular Applications*, Salzburg, Austria, 2011.

[16] S. Vacek, "Videogestützte Umfelderfassung zur Interpretation von Verkehrssituationen für kognitive Automobile," Dissertation, Universität Karlsruhe (TH), 2008.

[17] J. Knaup and K. Homeier, "RoadGraph - Graph based environmental modelling and function independent situation analysis for driver assistance systems," in *13th International IEEE Conference on Intelligent Transportation Systems*, Madeira, Portugal, Sept. 2010, pp. 428–432.

[18] Z. Papp, C. L. Brown, and C. Bartels, "World modeling for cooperative intelligent vehicles," in *IEEE Intelligent Vehicles Symposium*, Eindhoven, Netherlands, Juni 2008, pp. 1050–1055.

[19] C. Schlenoff, R. Madhavan, and T. Barbera, "A hierarchical, multi-resolutional moving object prediction approach for autonomous on-road driving," in *IEEE International Conference on Robotics and Automation (ICRA)*, vol. 2, New Orleans, USA, 2004, pp. 1956–1961.

[20] C. Galindo, A. Saffiotti, S. Coradeschi, P. Buschka, J. Fernandez-Madrigal, and J. Gonzalez, "Multi-hierarchical semantic maps for mobile robotics," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Edmonton, Canada, 2005, pp. 2278–2283.

[21] J. Oberlander, K. Uhl, J. Zollner, and R. Dillmann, "A region-based slam algorithm capturing metric, topological, and semantic properties," in *IEEE International Conference on Robotics and Automation (ICRA)*, Pasadena, USA, 2008, pp. 1886–1891.

[22] R. Regele, "Using ontology-based traffic models for more efficient decision making of autonomous vehicles," in *Autonomic and Autonomous Systems, 2008. ICAS 2008. Fourth International Conference on*, March 2008, pp. 94–99.

[23] J. McCall and M. Trivedi, "Video-Based Lane Estimation and Tracking for Driver Assistance: Survey, System, and Evaluation," *IEEE Transactions on Intelligent Transportation Systems*, vol. 7, no. 1, pp. 20–37, Mar. 2006.

[24] R. Gregor, M. Lutzeler, M. Pellkofer, K.-H. Siedersberger, and E. Dickmanns, "Ems-vision: a perceptual system for autonomous vehicles," *IEEE Transactions on Intelligent Transportation Systems*, vol. 3, no. 1, pp. 48–59, 2002.

[25] R. Gregor, "Fähigkeiten zur Missionsdurchführung und Landmarkennavigation," Dissertation, Univ. der Bundeswehr München, 2002.

[26] M. Manz, M. Himmelsbach, T. Luettel, and H. Wuensche, "Detection and tracking of road networks in rural terrain by fusing vision and LIDAR," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, San Francisco, USA, Sept. 2011, pp. 4562–4568.

[27] A. Reyher, "Lidarbasierte Fahrstreifenzuordnung von Objekten für eine Abstandsregelung im Stop&Go-Verkehr," Dissertation, TU Darmstadt, 2007.

[28] K. Homeier and L. Wolf, "RoadGraph: High level sensor data fusion between objects and street network," in *14th International IEEE Conference on Intelligent Transportation Systems (ITSC)*, Washington, USA, 2011, pp. 1380–1385.

[29] 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.

[30] D. H. Douglas and T. K. Peucker, "Algorithms for the reduction of the number of points required to represent a digitized line or its caricature," *Cartographica: The International Journal for Geographic Information and Geovisualization*, vol. 10, no. 2, pp. 112–122, 1973.

[31] B. N. Delaunay, "Sur la sphère vide," *Bulletin of Academy of Sciences of the USSR 7*, vol. 6, pp. 793–800, 1934.

[32] A. Reschka, J. R. Böhmer, T. Nothdurft, P. Hecker, B. Lichte, and M. Maurer, "A surveillance and safety system based on performance criteria and functional degradation for an autonomous vehicle," in *15th International IEEE Conference on Intelligent Transportation Systems (ITSC)*, Anchorage, USA, 2012, pp. 237–242.