Honda Research Institute EU
Honda Research Institute Europe GmbH
https://www.honda-ri.de/

Driving Situation Analysis with Relational Local Dynamic Maps (R-LDM)

Julian Eggert, Daniela Aguirre Salazar, Tim Puphal, Benedict Flade

2017

Preprint:

This is an accepted article published in FAST-zero Symposium 2017. The final authenticated version is available online at: https://doi.org/[DOI not available]

FAST-zero '17

Driving Situation Analysis with Relational Local Dynamic Maps (R-LDM)

Graph-Based Environment Representation for ADAS and AD

Julian Eggert [1] Daniela Aguirre Salazar [2] Tim Puphal [1] Benedict Flade [1]

1) Honda Research Institute (HRI) Europe, Carl-Legien-Str. 30, 63073 Offenbach, Germany
(E-mails: [julian.eggert, tim.puphal, benedict.flade]@honda-ri.de)
2) Department of Mechanical and Mechatronics Engineering, Universidad Nacional de Colombia,
Carrera 30 No. 45-03, 111321 Bogota, Colombia

ABSTRACT: In the automotive domain, the concept of Local Dynamic Maps (LDM) is currently used to describe a central storage place where static, quasi-static and dynamic road information is integrated by means of a common coordinate reference. In this paper, we argue that future ADAS and AD applications concentrating on driving situation analysis will require, instead of a layered, a fully interconnected graph-based approach, which we propose here in form of a Relational Local Dynamic Map (R-LDM). The graph-based approach lends itself for the creation of consistent world models with richer interconnected structures on spatial, temporal and semantic levels. We explain the design and explore the capabilities of the R-LDM in one application for camera-to-map alignment for lane-detailed localization, and in another application for the preparation of predicted trajectories for risk-based behavior evaluation.

KEY WORDS: Environment Representation, Local Dynamic Maps, LDM, Graphs, Graph Databases, Neo4j, ADAS, AD, Camera-to-Map Alignment, INLANE, Risk Measures, Predictive Risk, Behavior Planning, VI-DAS.

1. Introduction

In current efforts on improving driving support from intelligent Advanced Driver Assistance Systems (ADAS) up to Autonomous Driving (AD), an increasing number of functions are introduced, each of them specializing on different assistance domains and focusing on different aspects of the world as conveyed by sensors. Future ADAS applicable to more complex situations, will increasingly have overlapping functional domains which will require a more complete and consistent view on the world.

So the paradigm shift is from limited-domain sensor-controller loops towards more general systems demanding a more detailed and deeper representation and understanding of the surrounding environment. Alongside with this, there is a need for concentrating on particularly those aspects of the world which have consequences for the own behavior (e.g. in terms of safety and utility) and which allow further behavior prediction, planning and support. For all these purposes, for ADAS and AD driving support systems it is necessary to maintain online and in real time a consistent world model which comprises all relevant static (road structure), quasistatic (changing road elements like obstacles, statistical information, weather, etc.) and dynamic entities (traffic participants), together with information about their past (i.e., measured) and future (i.e., predicted) evolution through time.

Recent approaches have addressed such world models in terms of a central storage module based on so-called Local Dynamic Maps (LDMs [(1)], with efforts for standardization undergoing by ETSI [(2)] and ISO [(3)]). The main usage of LDMs currently targets at addressing cooperative functions based on managing and communicating information about traffic entities with some of their attributes and absolute positions. This occurs using Intelligent Transport System (ITS) infrastructure facilities supporting vehicle-to-vehicle (V2V), infrastructure-to-vehicle (I2V) and vehicle-to-infrastructure communication. For ADAS and AD driving support however, the analysis of driving patterns and situations, as well as relative sensor measurements and the relationships between objects get increasingly important (e.g. in the iACC function [(4)], a constellation of several items and their distance and velocity relationships is used to predict that a neighboring car will cut into our own driving lane).

In this work, we propose to extend the concept of LDMs towards a general relational representation of the driving environment on the basis of graph technologies, which can be used for wider support of driving functions such as situation analysis, risk evaluation and motion planning.

2. Approach

2.1. Overview

Future vehicles equipped with ADAS or AD functionality will require far deeper traffic situation understanding capabilities than nowadays systems provide. Based on a comprehensive representation of the surroundings, they will need an on-board world model which is able to operate in real-time. LDMs constitute a key component of such a world model, providing functionalities to store, update, recall and ideally do higher level inference on elements and relationships stored in the world model.

Previous LDM approaches concentrated mainly on managing lists of objects with map positions and additional attributes. For this purpose, it was sufficient to rely on standard SQL-type database technologies. Nowadays, we have do deal with an increasing amount of information of different types, from different sources (like external databases, vehicle-to-X communication or on-board sensors) and of different semantic level and quality. While in the past data was often treated in separation, we now have to integrate those different sources into a

Copyright © 2017 Society of Automotive Engineers of Japan, Inc. All rights reserved

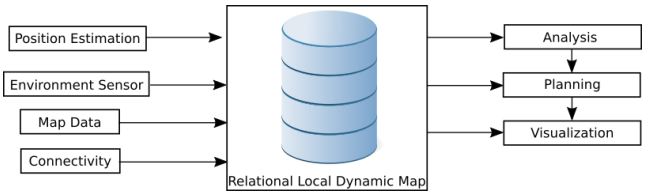

Fig. 1: Possible usage of R-LDM in ADAS applications.

single and coherent world model. In such a model, a large portion of the most valuable information resides in the embedding context, i.e., in the connectivity patterns.

With the increase of the importance of relations between objects, we propose to use native graph representations for an environment representation, with items as nodes and relative attributes as connections. This also yields a natural link to road structure data, since the road map interconnections and geometry data build a graph. The result are so-called Relational Local Dynamic Maps (R-LDMs) to emphasize the graph character. Road geometry representations in form of graphs have been proposed [5, 6], however this is to our knowledge the first specification that relies on standard graph database technology, interfaces and query languages, and which assorts several kinds of additional world knowledge within such a graph network, leading to a more general world representation.

The R-LDM serves as a hub for receiving, integrating, storing, fusing, updating and predicting ADAS-relevant data about the environment as depicted in Fig. 1. It presents an abstraction layer between sensor data (from ego vehicle or sent via connectivity from other entities) and higher level functions that evaluate environment data (such as position estimation, situation analysis, motion planning and visualization). Specifically, it stores information about spatial geometry and structure of the environment (extracted from different sources like map data and sensor readings). This is combined with entity data of objects and attributes as well as all relevant object-object, object-structure and structure-structure relations. For higher level interpretations, it is formulated in a sufficiently general way so that in future, driving situations containing patterns of interrelated objects, e.g. for situation classification, can also be handled. Further interesting candidates for storage in the R-LDM are trajectories, drivable paths and predicted trajectories, consisting of connected chains of entity positions.

Technically, we implement our R-LDM based on an off-the-shelf noSQL database specialized for graph and time series handling. References between items and map elements can occur via geospatial coordinates or by explicit relationships. An R-Tree-based spatial indexing scheme supports fast position-related queries. On top of the database we set up a series of “patterns”, which are subgraphs representing world concepts together with their properties.

2.2. R-LDM Concepts

Classical LDMs consist of a database management system for aggregating, processing and communicating multiple types of information about a vehicle’s sensed driving environment. In a specification released during the SAFESPOT project [7] and in subsequent research [8], the LDM has been realized as a four-layer

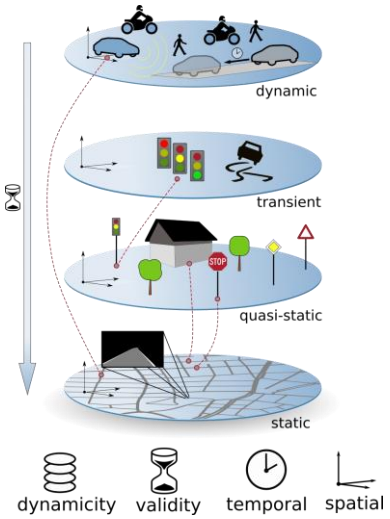

Fig. 2: Graph-based R-LDM concept.

model of data subject to change and dynamics at different timescales. On a first or bottom layer, static data (mainly map, road and infrastructure related) is managed, which is expected to change on a very slow timescale. A second layer consists of quasi-static data, which might be subject to change on a scale of several days or longer, such as roadside infrastructure, landmarks, traffic signs and construction sites. The third layer consists of relatively transient data, which changes on hours scale or faster, like traffic signals/phases, traffic density and congestion, or temporal road changes like slippery road conditions. Finally, the fourth layer comprises highly dynamic data about the state of traffic participants, including vehicles and pedestrians. While the lowest layer relies mostly on offline data stores, downloading the necessary data on demand, it is the highest layers which store and maintain the dynamic scene information recorded from post-processed sensor information.

Dividing the LDM into 4 layers makes sense to distinguish entities based on their dynamicity and length of validity. A vertical connection between the 4 layers is given by a reference to a shared spatial map coordinate system. However, the layering of the classical LDM also pays tribute to the necessary underlying SQL object type implementation. In general, the length of validity is related to the dynamicity, since entities of lower layers are likely to have a longer life expectancy than the ones on a higher level. Nevertheless, length of validity presents a separate axis and every entity of the LDM has its own validity value.

For a comprehensive representation of the environment, at least two further axes, in addition to dynamicity and length of validity, are of interest. On the one hand, regarding the purely spatial aspect different levels of spatial granularity are necessary to describe different levels of environment detail (see Section 2.4). On the other hand, it is important to separate past recorded state sequences of dynamic items from items that are valid for the current scene analysis and these from predicted future states of items for situation forecasting. In combination, this leads to a world model which is heavily interconnected and where the main information does not reside in the isolated items, but in the links and interconnection patterns that these form. If you then add a new item of data, you can easily connect it to the rest of the world.

Copyright © 2017 Society of Automotive Engineers of Japan, Inc. All rights reserved

This enriches your overall representation, since it embeds your item into a context. All four axes (dynamicity, validity, temporal and spatial axis) are illustrated in Fig. 2.

An efficient way to realize this enhanced and highly interconnected LDM concept is to use a graph for the representation. In the current implementation, we use a native graph database for this purpose, more specifically a labelled property graph model, resulting in a so-called R-LDM. The property graph contains nodes, which can hold attributes, expressed as key-value-pairs and connected by typified relationships also expandable by attributes. Additionally, the nodes can be tagged with labels to represent roles, classes or other metadata.

Relationships connect pairs of nodes to provide named semantically relevant directed information. The attributes of relationships can comprise quantitative information, like distances, costs or weights. Relating this back to our R-LDM, arbitrary relationship patterns of different types between nodes can be used to express the connections of the items along the different types of axes indicated previously. The power of such an R-LDM lies in the information contained in the connectivity between the data. As depicted in Fig. 2, entities of different layers of the R-LDM do not only share a common reference coordinate system, but are also connected by explicit relationships (e.g. direct connection of a dynamic vehicle with a road segment on which it is currently located). An additional advantage is that classical queries to an LDM, like routing and route finding at different levels of granularity, are native capabilities where graph models excel.

In our implementation, we use the latest version of Neo4j as a graph database for the R-LDM. This comes with the benefit of standard database capabilities, like security modeling with authentification and encryption, but also provides a standardized query language called Cypher for intuitive access. Cypher uses a declarative pattern to describe parts of the graph with an ASCII-type syntax. It allows you to easily describe what you want to find or do with the R-LDM information without having to care exactly on how to do it.

2.3. R-LDM Schema

The overall schema and specification of our R-LDM is given by different node and relationship types. Node types assign nodes to different classes, e.g. specific types of road elements, whereas relationship types define connectivity semantics. The (very small) basic set of relationship types is indicated in the list below:

- :HAS_PROPERTY
- :HAS_PART
- :SPECIALIZES_TO
- :NEXT_FORWARD
- :NEXT_LEFT
- :NEXT_RIGHT
- :CONTAINS

The relationship types determine the way they are used within the R-LDM. First of all, basic concepts are represented by nodes with concept labels acting as hubs and forming star-shaped graph subpatterns. A concept uses the first and second relationship types (“HAS_PROPERTY” and “HAS_PART”) to specify its meaning. As an example, a subgraph representing a car type instance is a hub centered on a node with label “:Car” and property nodes with labels “:Velocity”, “:Color”, “:Brand” connected by “:HAS_PROPERTY” relationships, as well as part nodes “:Motor”, “:Windshield”, etc. connected by “:HAS_PART” relationships. The difference between “:HAS_PROPERTY” and “:HAS_PART” relationships is that “:HAS_PART” designate hysical subcomponents which can by themselves again have subcomponents via a hierarchy of further :HAS_PART relationships. In the following, we will call the corresponding nodes linked by “HAS_PROPERTY” and “HAS_PART” the “property” and “part” nodes.

To build related but more specific concepts, we can use “:SPECIALIZES_TO” relationships connecting e.g. the car with another node of type “:PoliceCar”. Similar to “:HAS_PART”, “:SPECIALIZES_TO” relationships are transitive, so that you can build hierarchies / class taxonomies. The “:NEXT_FORWARD”, “:NEXT_LEFT” and “:NEXT_RIGHT” relationships are used to describe navigational information between elements, e.g. for connecting road parts which we will describe in the next Section 2.4. In order to describe links between entities of different LDM layers, “:CONTAINS” relationships are introduced. Thus, “:CONTAINS” presents an additional mean to connect the layers of the LDM independently of an explicitly shared geometrical reference system. This makes it possible, for instance, to connect dynamic objects such as cars with static road geometry information as e.g. the current lane or to assign traffic signs to the corresponding road. A more in-depth description of the process is given in Section 2.5.

2.4. R-LDM Static Road Infrastructure

The backbone of the road infrastructure network representation in the R-LDM is composed by road element nodes connected by relationships which indicate spatial connectedness (e.g. via “:NEXT_FORWARD”) and which can be traversed for path investigation and navigation. The road infrastructure is considered to be static at the moment of usage, therefore it is located in the first of the 4-layered LDM conceptualization.

One focus of our R-LDM is a description of road infrastructure and drivable area geometry at the lane level of accuracy. This is necessary for AD and ADAS applications, since they require a lane- or sub-lane-accurate localization of the traffic participants and a description of the drivable environment in order to foresee how the driving scenes evolve and estimate the appropriate behavioral response. In track with different efforts for capturing highly detailed lane geometry [(9)] and standardization [(10)], we store lane segments as subgraph patterns consisting of a central node (node label “:LaneSegment”) together with attached part and property nodes such as

- Polygonal lines (polylines) for left and right boundary
- Center polygonal line, corresponding to a drivable path section
- Lane markings and stop lines

Copyright © 2017 Society of Automotive Engineers of Japan, Inc. All rights reserved

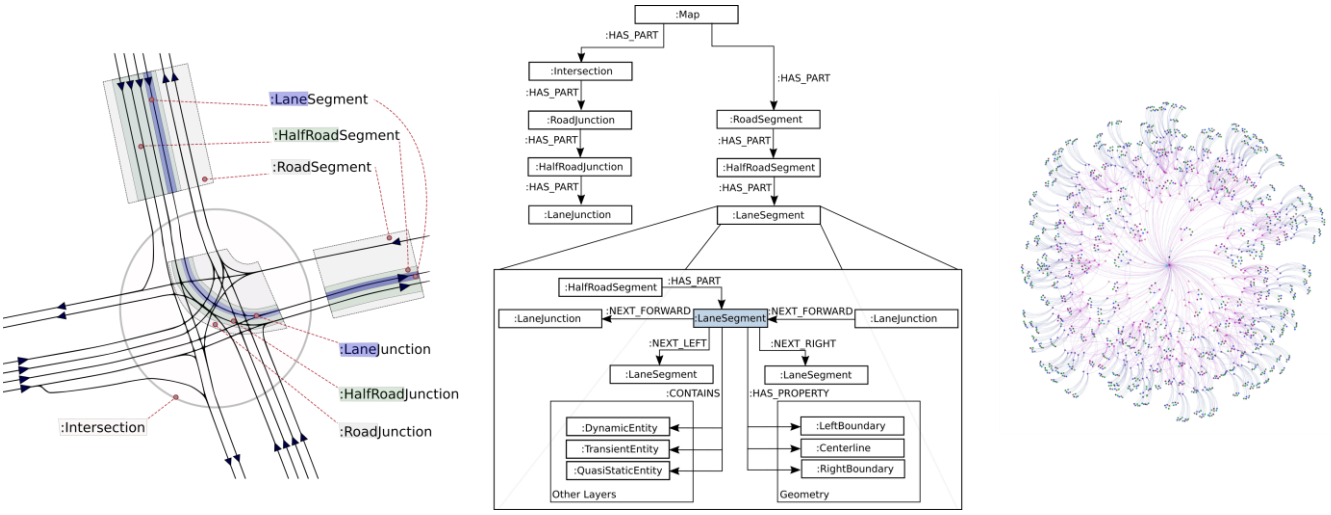

Fig. 3: Left: Three granularity levels of static elements. Middle: Map infrastructure diagram. Right: Visualization of R-LDM graph.

- Angles at intersections
- Driving direction
- Other information like height and curvature

It should be noted that the choice of properties is not limited to the listed ones and highly depends on the targeted use case. Overall, there are no limits to your creativity but also the omission of properties can be a method of choice, since the performance may raise if e.g. only intersection centers are stored instead of full geometry information.

For applications such as estimation of path-dependent collision risk, the actual paths a driver most likely takes are of high interest. Usually, the center polygonal line of a lane serves as a suitable approximation of a drivable path section. However especially when dealing with complex intersections, the actual drivable path may differ strongly from the lane center and therefore an explicit storage of a drivable path property may be beneficial. Furthermore, in such applications dynamic objects like other traffic partipicants (layer 4) are closely related to the static information, namely the drivable path and further properties. In this context, the interconnectivity due to its layer of overarching relationships pays off and allows convenient and quick information retrieval.

Each lane segment describes a section of a road with the width of a lane which connects to other road elements at its starting and ending sections via ":NEXT_FORWARD" relations and laterally via ":NEXT_LEFT" and ":NEXT_RIGHT" relations. Its counterpart is the ":LaneJunction" which is a section of a road and connects two lane segments typically at intersections. The lane segments and lane junctions together form the road network at lane level.

Depending on the AD/ADAS task, also road geometry information at coarser levels of detail might be required. In our R-LDM, we store geometrical and topological information in three different levels of detail (LOD). The coarsest LOD is the road and intersection level, here the geometry of an entire ":RoadSegment" is described along with road information like the road category (highway, rural, residential, etc.) and the road name. In case of an ":Intersection", the polygonal intersection area is stored. At a second level of detail, we describe ":HalfRoadSegments" and ":HalfRoadJunctions" which represent pieces of plain road and intersection connectors with a well defined driving direction. At the highest level of detail, we have the ":LaneSegments" and ":LaneJunctions". The 3 levels of detail form a ":HAS_PART" hierarchy with ":RoadSegments" being composed of typically two ":HalfRoadSegments" and each of them comprising all ":LaneSegments" with a matching driving direction.

In the case of crossings, highway exits or similar road structures, several roads may intersect leading to several ":RoadJunctions". Therefore we introduce ":Intersections" which serve as a hub for all related ":RoadJunctions". From a geometrical point of view, ":Intersections" store the polygonal intersection area and the coordinate of the intersection center.

Fig. 3 shows on the left side the spatial hierarchy of R-LDM road elements, the road infrastructure diagram in the middle and the whole graph of a map section on the right.

2.5. R-LDM Dynamic Objects

The static road infrastructure of the R-LDM is modified and enhanced by dynamically changing object nodes of the last three LDM layers. If new information is acquired from the ego car's sensors, through sensor inputs of other surrounding cars via V2V or over nearby infrastructure units with I2V, an object node is created and attached to the corresponding road element nodes with the relationship ":CONTAINS".

According to the type of the information, we moreover set a label for the object node (e.g. ":Car"). In this way, it is possible to query the object node not only when following structure-structure and then object-structure relations, but also through direct object-object connections with the help of the graph-based label search. Appending and using supplemental part and type nodes with ":HAS_PART" and ":SPECIALIZES_TO" relationships to the object node from Section 2.3 refines the search.

Every added object node is the central node of a subgraph with property nodes labeled ":Timestamp" and ":Values", which are linked with ":HAS PROPERTY" relationships to the central node. Once further values of the same dynamic object are received, both property node entries are extended. Even if the dynamic object leaves the road element nodes it is linked to, the ":CONTAINS"

Copyright © 2017 Society of Automotive Engineers of Japan, Inc. All rights reserved

relationship does not change. On the contrary, if the dynamic object is not relevant for the ego car anymore, the object subgraph is deleted in the R-LDM.

Lane segments can contain traffic lights and signs as well as other vehicles (cars, pedestrians and motorcycles). The stored properties thus are

- traffic light phases and sign status
- vehicle's current acceleration, velocity and yaw rate together with past measured and predicted trajectories

Road segments potentially have dynamically altering traffic congestion, road state, on road or road side obstacles and construction sites. We enclose and remove subgraphs with properties

- average velocity of convoy
- friction coefficient of road surface
- position of obstacle
- limited velocity at road works

Currently, only the longitudinal and lateral position of the static road infrastructure is saved as polylines in road element nodes. When the height of the road is additionally considered, damages of the roadway lead to changing properties of the road element nodes. Similarily at construction sites, road element nodes occasionally have to be restructured over time.

3. Applications

3.1. Camera-to-Map Alignment

For many ADAS and AD applications, dynamic objects from the environment have to be matched to a particular driving lane by their positions and orientations (see e.g. Section 3.2 for an example). This requires a high overall system accuracy which consistently represents and aligns sensor readings as well as state estimation of the ego vehicle and other traffic participants on one side with the road infrastructure data at lane level on another side.

On the far future, many efforts like highly accurate map creation and sub-lane vehicle localization will contribute to improve the absolute accuracy and consistency. Nevertheless, there are many sources of errors for the different types of information, as e.g. offsets in ego-localization, map data and satellite images as well as in vehicle sensor calibration.

For ADAS and AD, one solution is to try to improve the relative matching between sensor-related data (layers 3 and 4 in Fig. 2) and the static environmental information (layers 1 and 2). In the following application example, which is related to the H2020 project *INLANE*, we show an example in which this matching is used for map-relative local localization of the ego car.

Fig. 4 shows the general pipeline of the camera-to-map alignment approach. The two major inputs are the camera and the static map information stored in the layers 1 and 2 of the LDM. Due to its wide availability in nowadays production vehicles, cameras present a suitable sensor for achieving the alignment objective. The idea is to match visual cues detected within the camera image with corresponding features of the static geometric road environment stored in the LDM. Examples of such visual cues are lane markings, curbs or road texture. A wide range of camera types is possible, ranging from mono front cameras to 360

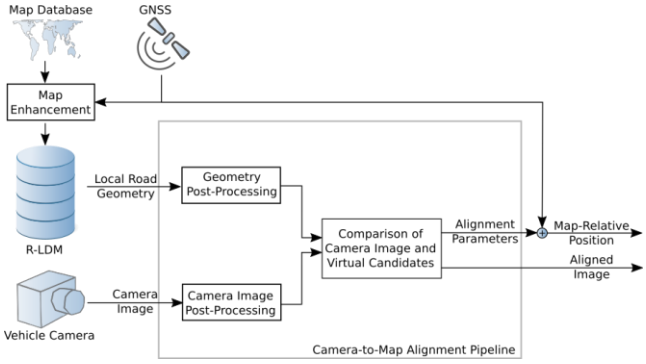

Fig. 4: General pipeline of camera-to-map alignment.

degree surround view stereo camera systems. We here provide an approach that is not limited to, but suitable for affordable mono front cameras without the restraint of requiring high end sensor equipment.

Depending on the underlying map data and the expected purpose of the alignment, a map enhancement may be necessary. In our research [11], we derive map geometry from the publicly available crowdsourced map database OpenStreetMap (OSM) [12]. The advantages are, amongst others, regular updates and the possibility of easily editing the data with open-source software. However, OSM only provides basic geometric information such as the center polyline on road and half-road level as well as tagged information such as e.g. road type and lane number. For a matching of cues such as detected lane markings or curbs, further map geometry information like left and right delimiting polylines are needed. We therefore enhance the OSM data as depicted in Fig. 5.

Building upon the road level centerlines obtained from OSM (Fig. 5 a), we infer the corresponding lane centerlines using lane number information and road-type related lane width estimation (Fig. 5 b). Based on the lane level centerlines, we additionally infer connections between the individual lane segments by heuristics (Fig. 5 c). At this point, the connection information is exclusively on a topological level, but no geometrical junction information is available yet. We therefore interpolate between all related lane segments leading to gapless lane-level geometry information (Fig. 5 d). In a last step, we add the left and right delimiter polylines (Fig. 5 e).

Our alignment algorithm is based on a comparison of post-processed road geometry information with the post-processed real-world camera image. The road geometry information is dynamically obtained from the R-LDM and includes the road delimeter markings. A possible road view, referred to as candidate, is obtained by simulating the view of the road shape as seen from the position of a virtual camera.

With the two input sources being defined, the next step is to recall the needed map information. For the alignment, only local map geometry within the range of vision is needed. The R-LDM natively allows an efficient traversal through the graph. Surrounding road elements can therefore be easily retrieved by following relations such as ":NEXT_FORWARD", ":NEXT_LEFT" or ":NEXT_RIGHT". Nevertheless, we require

Copyright © 2017 Society of Automotive Engineers of Japan, Inc. All rights reserved

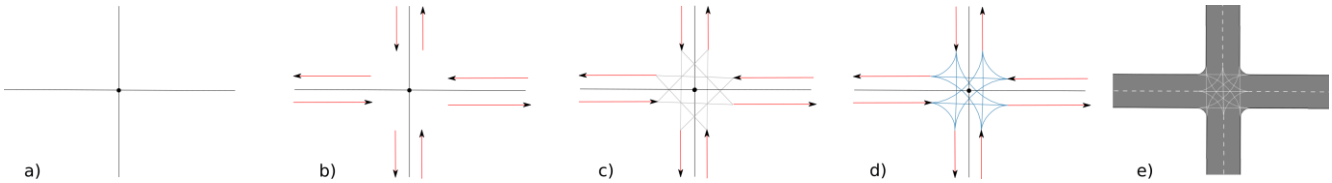

Fig. 5: Map data enhancement. a) Road centerlines. b) Calculation of lane centerlines. c) Inference of topological connections. d) Interpolation of junction geometry. e) Estimation of delimiter polylines.

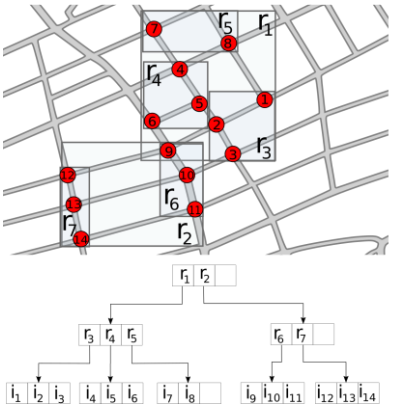

Fig. 6: Indexing of 14 intersections using an R-Tree.

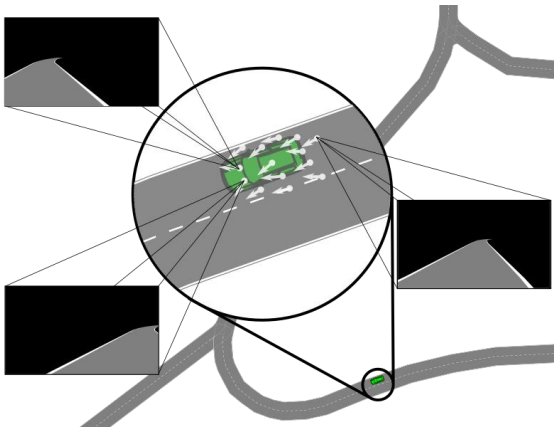

Fig. 7: Ego vehicle with three potential candidates zoomed in.



Fig. 8: Examples of successful alignment of road geometry (white lines) and camera image.

an initial node from which the graph is traversed. If known from previous queries, the road, half-road or lane on which the ego car is located can be used as an entry node. However, especially when first recalling information with no starting node given, we rely on absolute coordinates. In this context, we use Global Navigation Satellite Systems (GNSS) such as GPS, GLONASS or Galileo to obtain a coarse absolute position estimation. We use an R-Tree, which is also a graph structure, for efficient indexing from metric coordinates (e.g. latitude / longitude) to graph elements. R-Tree data structures group objects by their location and store the minimum bounding rectangle of each group. Fig. 6 shows the center of 14 intersections, plotted in a 2D coordinate system as well as the bounding boxes and the corresponding R-Tree graph. Besides quick graph entry, the R-Tree allows efficient recall of nodes within a certain spatial radius around a given position or to find the nearest node of a given type.

Fig. 7 illustrates the ego vehicle with potential candidates (white arrows) around the coarse position estimate. Three of the candidates are zoomed in, showing the virtual road view generated from the geometric LDM information. The set of road views considers different possible ego vehicle positions and orientations allowing up to six degrees of freedom. By performing a best-match search based on a feature comparison between the post-processed camera image and the virtual road views, we get the best matching candidate as well as its parameters.. Fig. 8 shows three examples of successful map to camera image alignment using videostreams from the KITTI Dataset [13]. The map geometry (white lines) matches the road boundaries of the camera image.

Regarding the target of allowing lane-level navigation, we can then use the parameters of the best matching candidate for the ego vehicle. With this new aligned position, a relation is established between static map data derived from external sources estimating a corrected map-relative position and orientation of and measurements of different kinds of sensors, denoted in vehicle coordinates, such as detected other traffic participants or road constructions.

3.2. Path-Dependent Risk Estimation

Situation understanding and concurrently risk estimation, two core features of ADAS and AD, not only require lane-accurate positioning of cars on map data from Section 3.1, but also the deduction of their future paths and intentions. For example in the so-called eHorizon [14], road properties of the ego car's future path (e.g. slope, curvature and speed limit information) are extracted with the help of GNSS and map data. A predictive cruise control function then adjusts the engine management automatically for a reduced fuel consumption. However, the underlying map is only based on the coarse road level and no interaction with other traffic participants is taken into account.

With the purpose of ADAS and AD behavior planning, we now rely on the R-LDM to extract an extended virtual horizon which is an environment model interrelating road structure on the

Copyright © 2017 Society of Automotive Engineers of Japan, Inc. All rights reserved

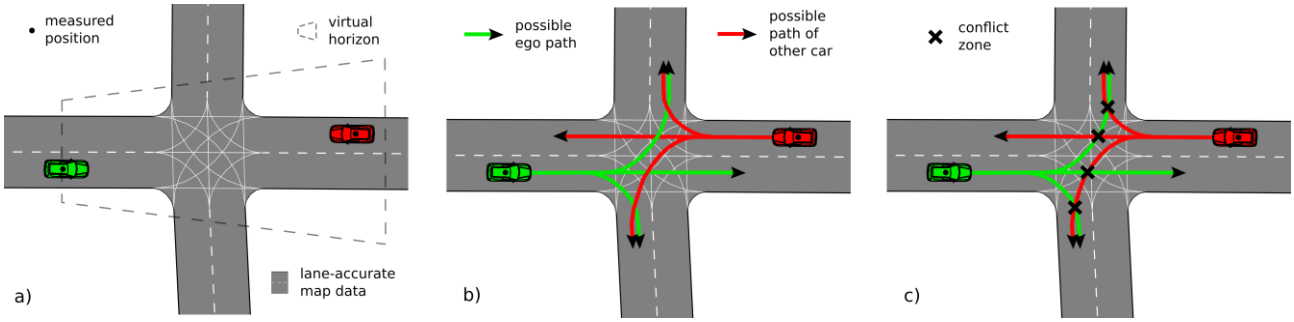

Fig. 9: Steps of the virtual horizon. a) Filtering relevant cars and their position. b) Extraction of possible future paths. c) Map-based risk estimation.

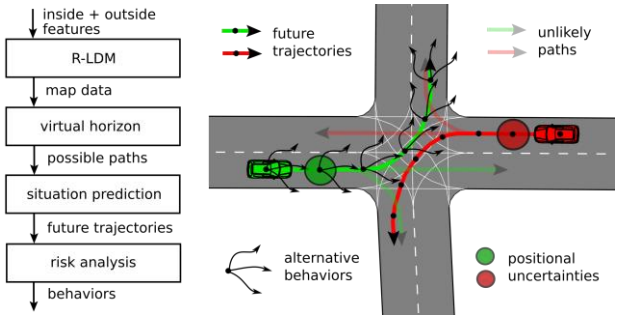

Fig. 10: Path-dependent risk estimation with supplemental situation classification, trajectory prediction and risk analysis. Left: Block diagram. Right: Visualization of methods.

lane level with driving behavior. Fig. 9 illustrates its functionality for a simple intersection scenario with two approaching cars. First, we map the position of the ego car (green) with the R-LDM over a ":CONTAIN" connection by finding the nearest lane segment. The surrounding lane segments and the lane junctions of the intersection as well as the other car (red) can then be retrieved by the virtual horizon with queries on the R-LDM up to a certain distance from the ego car using ":NEXT_FORWARD", ":NEXT_LEFT" and ":NEXT_RIGHT" relationships (Fig. 9 a). When we scan and concatenate the geometries of the subsequent lanes from the aligned position of both cars in the virtual horizon and order each branch (straight, left and right) according to a tree structure, we obtain three possible paths for the ego and other car respectively (Fig. 9 b). The R-LDM has thereby the advantage of a graph-based embodiment, allowing a fast retrieval of connected road information. Calculating the intersections of each path pair eventually leads to conflict zones, which are indicative for the collision risk of the ego car (Fig. 9 c).

The combination of R-LDM and virtual horizon is employed in the H2020 project *VI-DAS*, in which cameras record the outside and inside of the ego car to warn the driver of upcoming risks. Each measurement of static and dynamic objects as well as of the ego driver characteristics are fused into the R-LDM. Based on selected velocity profiles assumed for each car, the possible paths from the virtual horizon are analysed on spatiotemporal co-occurrence probability with features (e.g. distance to intersection) being supplied from the R-LDM. The virtual horizon serves the purpose to constrain the trajectories to the underlying detailed road geometry. Finally, if we compute the distances between the most likely trajectory of the ego and other car and apply a risk analysis which models positional uncertainty as e.g. Gaussian methods [15] or the survival analysis [16], we obtain criticality indicators for behavior recommendations or actuation. The presented complete path-dependent risk estimation approach is summarized in Fig. 10 for the intersection example.

4. Conclusion

In this paper, we proposed a graph-based Relational Local Dynamic Map (R-LDM) as an alternative resp. extension of the classical LDM concept. Its structure in terms of interconnected nodes suits to several functions in a natural way, in particular concerned with path finding, routing and tree search operations. In addition, we propose to exploit the usage of relationships between nodes in a more intensive way, since much information lies in the embedding context of an object and thus in the way that it connects to its related world entities. In this way, information of very diverse type, including semantic information, can be used in the R-LDM to build a consistent world model around the driver. Such a world model also lends itself in a much more natural manner to the extraction of behaviorally relevant situation and driving information, such as shown with the application examples for the upcoming lane-detailed road information or the trajectory prediction of appearing other cars for risk prediction.

Acknowledgments

This work has been supported by the European Union's Horizon 2020 projects *INLANE* (grant agreement number 687458) and *VI-DAS* (grant agreement number 690772).

References

(1) Z. Papp, C. Brown and C. Bartels : World Modeling for Cooperative Intelligent Vehicles, in *Intelligent Vehicles Symposium*, pp. 1050-1055 (2008).
(2) ETSI TR 102 863 (V1.1.1) : Intelligent Transport Systems (ITS); Vehicular Communications; Basic Set of Applications; Local Dynamic Map (LDM); Rationale for and Guidance on Standardization (2011).
(3) ISO 17931 : Intelligent transport systems - Extensions of map database specifications for LDM for applications of Cooperative ITS (2013).
(4) J. Schmuedderich, S. Rebhan, T. Weisswange, et al. : A novel approach to driver behavior prediction using scene context

Copyright © 2017 Society of Automotive Engineers of Japan, Inc. All rights reserved

and physical evidence for intelligent Adaptive Cruise Control (i-ACC), in *FAST-zero Symposium*, pp. 85-92 (2015).
(5) S. Ulbrich, T. Nothdurft, M. Maurer, et al. : Graph-Based Context Representation, Environment Modeling and Information Aggregation for Automated Driving, in *Intelligent Vehicles Symposium*, pp. 541-547 (2014).
(6) J. Knaup and K. Homeier : RoadGraph – Graph based environmental modelling and function independent situation analysis for driver assistance systems, in *Intelligent Transportation Systems Conference*, pp. 428-432 (2010).
(7) C. Zott, S. Ying Yuen, C. L. Brown, et al. : SAFESPOT Local Dynamic Map – Context-dependent view generation of a platform's state & environment, in *World Congress on Intelligent Transportation Systems* (2008).
(8) H. Shimada, A. Yamaguchi, H. Takada, et al. : Implementation and Evaluation of Local Dynamic Map in Safety Driving Systems, in *Journal of Transportation Technologies*, pp. 102-112 (2015).
(9) P. Bender, J. Ziegler, C. Stiller : Lanelets: Efficient Map Representation for Autonomous Driving, in *Intelligent Vehicles Symposium*, pp. 420-425 (2014).
(10) R. Behrens, T. Kleine-Besten, W. Pöchmüller, et al. : Digitale Karten im Navigation Data Standard, in *Handbuch Fahrerassistenzsysteme*, pp. 514-523 (2015).
(11) G. Cao, F. Damerow, B. Flade, et al. : Camera to Map Alignment for Accurate Low-Cost Lane-Level Scene Interpretation, in *Intelligent Transportation Systems Conference*, pp. 498-504 (2016).
(12) OpenStreetMap Contributors, accessed 2017-22-06, available: www.openstreetmap.org.
(13) A. Geiger, P. Lenz, C. Stiller, et al. : Vision meets Robotics: The KITTI Dataset, in *International Journal of Robotics Research*, pp. 1229-1235 (2013).
(14) H. Pu : Dynamic eHorizon with Traffic Light Information for Efficient Urban Traffic, in *Advanced Microsystems for Automotive Applications*, pp. 15-24 (2015).
(15) S. Patil, J. Van den Berg and R. Alterovitz : Estimating Probability of Collision for Safe Motion Planning under Gaussian Motion and Sensing Uncertainty, in *International Conference on Robotics and Automation*, pp. 3238-3244 (2012).
(16) J. Eggert : Predictive Risk Estimation for Intelligent ADAS Functions, in *Intelligent Transportation Systems Conference*, pp. 711-718 (2014).

Copyright © 2017 Society of Automotive Engineers of Japan, Inc. All rights reserved