Deliverable D4.5

HD Mapping Modules

Dissemination level	**PU**
Version	**1.0**
Lead contractor	**VTT**
Due date	**30.09.2022**
Version date	**08.11.2022**

Document information

Authors

Kimmo Kauvo – VTT

Matti Kutila – VTT

Pasi Pyykönen – VTT

Titta Kemppi – VTT

Marcus Bartels – Ibeo

Lauri Lehti – Unikie

Funding

Co-labelled PENTA and EURIPIDES2 project endorsed by EUREKA, National Funding Authorities:
Austrian Research Promotion Agency (FFG)
Business Finland
Federal Ministry of Education and Research (BMBF)
National Research Council of Canada Industrial Research Assistance Program (NRC-IRAP)

Contact

Dr. Werner Ritter
Manager Vision Enhancement Technology
Environment Perception
Mercedes-Benz AG
RD/AFU
Werk/Plant 05919 - HPC G005-BB
D-71059 Sindelfingen, Germany
Mobile +49 (160) 863 8531
werner.r.ritter@mercedes-benz.com

Table of contents

1 Executive summary 5

2 Introduction 6

3 Framework 7

4 Interfacing with HD maps 10

5 Demonstration 12

6 Conclusion 14

List of abbreviations 15

References 16

List of figures

Figure 3.1: Simulator – Digital Twin – Real World workflow sketch 7

Figure 3.2: Location engine –Map server – Real World workflow sketch 8

Figure 3.3: Location engine architecture 9

Figure 5.1: Vehicle demonstration sensor system with gated camera (1) and LiDAR 12

Figure 5.2: HD map information connected real driving environment 13

List of tables

Table 3.1: Comparison between conventional and HD maps (TomTom) 9

Table 4.1: Layered structure of the HD maps 10

Table 4.2: Industry standards for the HD maps 10

1 Executive summary

This deliverable is related to the AI-SEE project *WP4 (Fusion & AI)* and *task 4.5 (HD mapping)* which aims to enhance automated vehicle electronic horizon with providing map database for supporting recognition of road elements in adverse weather. The aim is to collect new elements with using SLAM technique for increasing geographical coverage of the map data and adapt to the actual weather conditions which limits visibility.

The purpose of this document is to describe elements of autonomous driving network and how the digital 3D virtual environment connects with the real world using the simulation environment. This document also presents the structure of the HD maps with available industry standards. Finally, demonstrations chapter connects digital 3D virtual environment together with demonstrations in real world.

Chapter 3 depicts the main elements of the whole autonomous driving framework, i.e., how the digital 3D virtual environment relates to the real world using the simulation environment. The chapter points out that there are two major 3D computer graphics game engine environments in the market to use. As a sub problem of the SLAM, the chapter presents the point cloud localizing. It also presents the general Ibeo location engine architecture. The chapter concludes with the table comparing conventional and HD maps.

Chapter 4 presents the generally accepted layered structure of the HD maps. Also, a table presenting currently available HD map industry standards is presented. It also points out that a remarkable benefit is gained when HD data sources are provided via Web Feature Services.

Chapter 5 presents demonstrations which connect digital 3D environment together with real world environment and scenarios. Furthermore, this chapter introduces an example how to enable HD maps with real time measurement with vehicle and simulator environment.

2 Introduction

In this deliverable, we explore the applications of HD maps and AI-SEE use cases [1], and how one would approach creating one themselves. We introduce why HD maps are in the rise and increasingly used in simulated worlds and data collection, especially what they are and consist of. After that we discuss different ways of accumulating the data for a map, what kind of frameworks can be used to create one, what are the differences between those approaches, and why we chose the ones we did. Finally, we demonstrate our own solution for HD mapping: what worked well and what the solutions are especially good for, without forgetting what could be improved. These results can then be used to learn about the more intricate possibilities of HD mapping and help broaden our horizons with developing HD maps further in the future.

3 Framework

Figure 3.1 depicts the whole autonomous driving workflow (Simulator - Digital Twin – Real World). The High-Definition map (HD map) is a vital part, needed by the Digital Twin as well as by the Simulator & Autonomous software (sw.). Virtual world is an important element in developing autonomous driving. Used virtual world simulator can be run with the Unreal or Unity game engine. Autonomous sw. stack can be connected to simulator via e.g., a ROS2 (Robot Operating System) bridge, or the simulator can have some software ready-built for use. Thus, the simulator is depicted twice, as its own block and as a part of the autonomous sw.

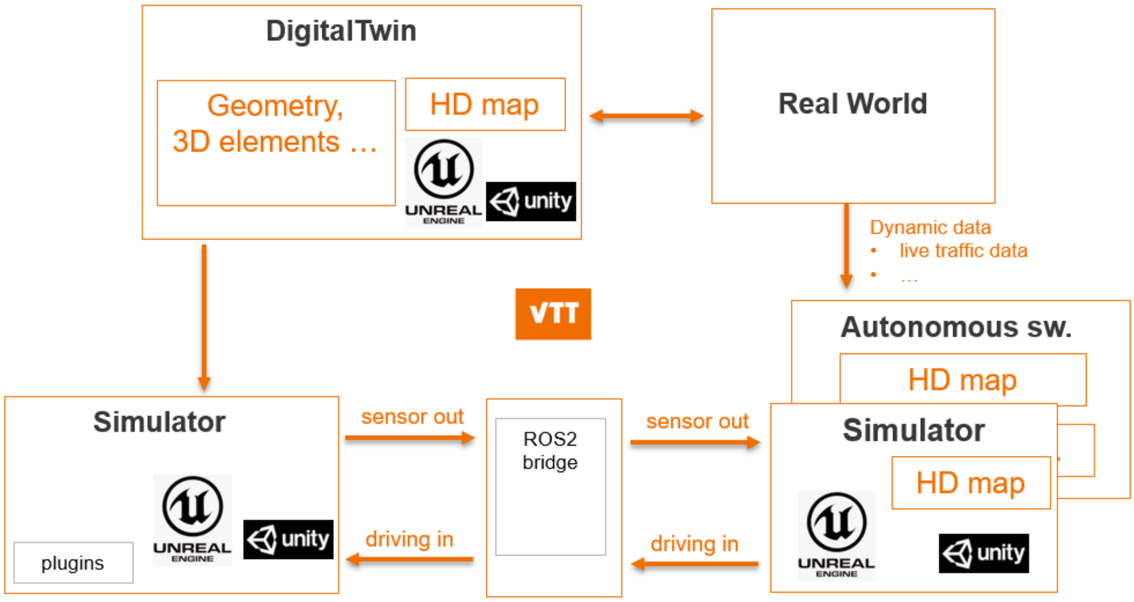

Figure 3.1: Simulator – Digital Twin – Real World workflow sketch

A digital twin is a virtual representation of a real-world object, environment or a system, typically built so it can be used to simulate how the thing would interact in the real world. In this deliverable, a digital twin refers to an accurate, recognisable virtual representation of a town, especially in the context of simulating autonomous driving within it.

HD map is a major component in creating a digital twin. For digital twins to be useful in autonomous driving simulations, they need to be an accurate representation of a real-world physical environment, which is why more than just graphical components are needed. There needs to be knowledge of, for example, how different components interact with each other, what are the rules and laws of the surroundings (traffic lights, lanes, signs) and which parts of the surrounding can be driven or walked on. Essentially, the digital twin consists of an HD map and graphical components.

Simulators create virtual environments and maps for simulations with an HD map as a core. The simulated ego vehicle interacts with and reacts to its surroundings via the information provided by the HD map. Each simulator requires the HD map to be in a particular industry standard, for example lanelet2 or OpenDRIVE format. This needs to be considered either in HD map creation or when choosing a simulator.

Along with the simulator and digital twin, the HD map is also needed by the autonomous software, so it can make decisions according to the information about the surroundings of the vehicle being moved. Therefore, in Figure 3.1 the HD map is illustrated thrice, along with underlining how significant part of automated driving simulation it is.

Figure 3.2 illustrates the location engine workflow. The location engine utilizes perception information such as traffic signs and poles from the real world, and real-time HD map data from the cloud, to calculate an accurate position of the vehicle. The location engine relays on the centimetre accuracy of the HD map for certain elements which can be also detected from the point cloud generated by the LiDAR to make the positioning calculation [2]. The location engine can also recognize new elements such as snow piles that are not part of the current HD map data. These kinds of new elements are updated on the HD map located on the HD map server to keep the map information real-time.

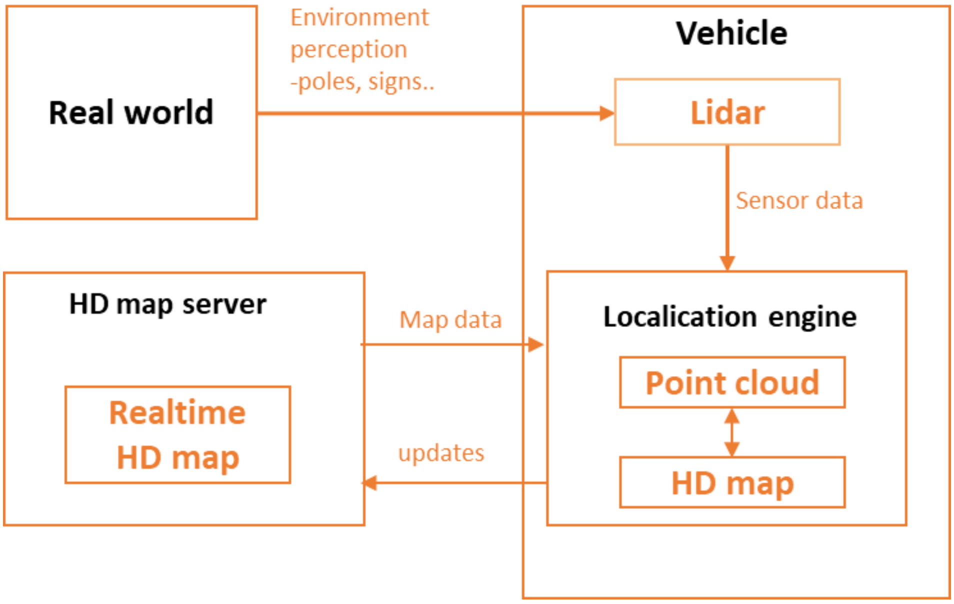

Figure 3.2: Location engine –Map server – Real World workflow sketch

Figure 3.3 depicts the architecture of the Ibeo Localization Engine. Central components are landmark detection, map reader, HD map fusion and the localization core module. The latter hosts the localization algorithm, which uses the HD map together with landmark detections from LiDAR point clouds and motion/position sensing data (GNSS, inertial measurements and vehicle odometry) to find matches between detected real-world landmarks and HD map features. The HD map fusion module uses the localization result and a sensor visibility estimation to identify discrepancies between the perceived real world and the HD map (map deltas). These results are used to update the HD map internally, and they can be provided to an external map database. The map reader acts as input interface for the HD map, loading static map data from map definition files or receiving map data from an external database. These data are converted into an internal format and provided to the localization and map fusion modules. The interface towards a real-time map database is only partially implemented so far.

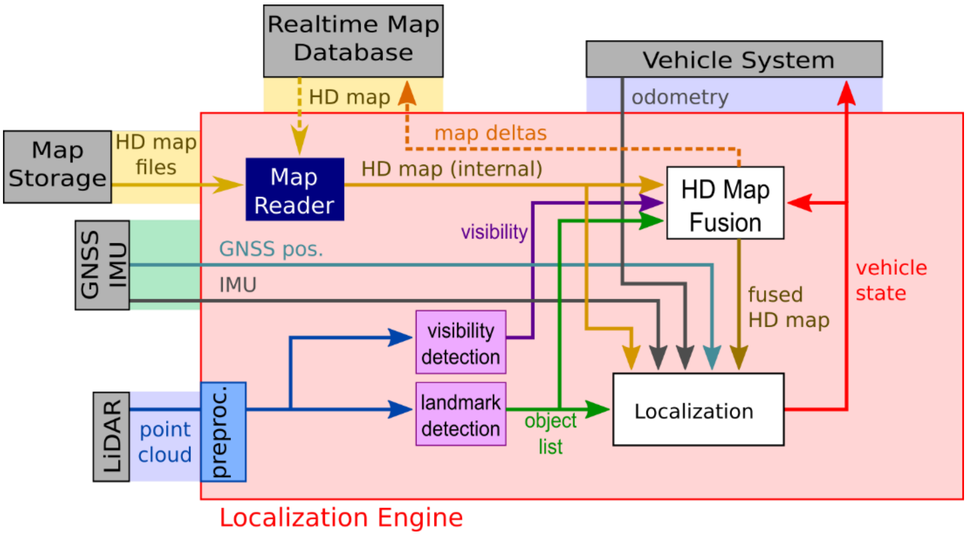

Figure 3.3: Location engine architecture

High-Definition maps (HD-maps) are maps to support autonomous driving. HD maps help autonomous vehicles to understand the environment precisely beyond the sensor range and help vehicles to localize itself on centimeter level by providing accurate road furniture e.g. poles and curb stones. It provides detailed enough information about the environment that an autonomous vehicle can plan a route and help to navigate to the destination based on lane-level description.

Table 3.1 presents the differences between conventional digital map and HD map. HD maps provide high-quality a priori information for the object detection process especially in urban canyons, glare lighting conditions and under harsh weather conditions, e.g. when visibility is poor or lane markings are faded or covered with snow.

Table 3.1: Comparison between conventional and HD maps (TomTom)

Parameter	**Conventional Digital Map**	**HD Map**
Location accuracy	few meters	few centimeters
Lane-level information	turn info at intersections	all throughout the map
Additional information	road level • traffic signs • speed limits • curvature of turns at intersections • slope	lane level • traffic signs • speed limits • lane geometry • curvature and slope
Level of attribution	location and value/meaning	location, height, lane width, meaning, lane category, position relative to the lane, lane associations

4 Interfacing with HD maps

Table 4.1 depicts the structure of HD maps, which can be understood as layered structures; Geometric, Semantic, Road/Lane network, Experiental and Realtime. The Geometric layer is composed of raw sensor data collected from LiDAR, cameras, GNSS, and IMUs. The output is a 3D point cloud. The Semantic layer is adding semantic objects. These can be either 2D or 3D, such as lanes, lane boundaries, intersections, parking spots, traffic signs, traffic rules etc. that are used for driving safely. The Road/Lane network layer contains information on how to form a complete road network. The Experiental layer contains information collected from the experience; risky intersections based on previous accidents, red light violations, speeding etc. The Realtime layer is the topmost layer in the map: it is dynamically updated and contains real-time traffic information on traffic lights, traffic jams, current accidents etc. This data can also be shared real-time between vehicles.

Table 4.1: Layered structure of the HD maps

Realtime	traffic lights, traffic jams, accidents
Experiental	information collected from past experience
Road / Lane nw.	connection between lanes, how to form a road network
Semantic	lanes, traffic signs, speed limits, traffic rules
Geometric	pointcloud, linestrings, points

Table 4.2 lists four currently available HD map industry standards; Autoware Vector Map, Navigation Data Standard, OpenDRIVE and Lanelet2. From the openness point of view, OpenDRIVE and Lanelet2 are the real options. The storage definitions are the following: Autoware Vector Map (csv), Navigation Data Standard (SQLite), OpenDRIVE (xml), Lanelet2 (osm xml).

Table 4.2: Industry standards for the HD maps

	Autoware Vector Map	**NDS**	**OpenDRIVE**	**Lanelet2**
Easiness	**3** (multiple tools avail.)	**2** (seems difficult to create all information)	**1** (not many tools to write map)	4 (node based)
Tools	**3** (Read/Write for ROS software)	**4** (many tools are available after purchasing license)	**4** (many tools but mostly for simu., not for AD sw.)	**3** (many OSM tools but not much for Lanelet2)

	Autoware Vector Map	**NDS**	**OpenDRIVE**	**Lanelet2**
Adoption of format	**1** (not publicly used)	**5** (created by major automotive and tier-1 companies, Stand. under NDS Association)	**4** (many automotive and tier-1 companies)	**2** (some map vendors providing maps in Lanelet2 format)
Relation to prod. systems	**1** (no plans for connect to prod. systems)	**5**	**3** (not directly used in prod. env. But can be converted to NDS)	**1** (relatively new)
Expressiveness	**2** (limited)	**5** (detailed specification is closed, but is expected to be very high)	**4** (traffic sign id might be problem)	**3** (extendable format, but has less information compared to OpenDRIVE at the moment)
Interchange	**2** (conversion from OpenDRIVE and Lanelet)	**2** (bidirectional converter to OpenDRIVE)	**3** (bidirectional converter to NDS)	**3** (Lanelet1 and OpenDRIVE converter available)
Accessibility	**2**	**1** (licensed)	**4**	**5**

Scale (1=bad, 5=good)

Besides the mentioned HD map formats established in the automotive industry, various geographic data formats exist, which are used in geographic information systems (GIS) to manage geospatial information of application fields like infrastructure management or public administration. In the automotive HD map context, vectorial geodata formats can be relevant especially when using HD map data provided by infrastructure operators or public authorities. Such data can be either provided via online GIS services such as Web Feature Service (WFS) or as files. There are multiple common file formats such as GeoJSON or shape files (shp), which can be converted among each other using standard GIS software. However, the exact structure and content of map information in such geodata formats is not standardized, which might require provider-specific adaptations of the map interface.

The Ibeo map reader currently supports OpenDRIVE and GeoJSON input files. For the latter, the mapping of information into the internal format can be configured according to the map source.

5 Demonstration

In chapter 3, Figure 3.1 we showed how a DigitalTwin connects the simulator environment with HD map into the real-world vehicle and sensors system. Figure 5.1 shows the real-world vehicle sensor system needed to connect HD map data with the road environment. The key elements of the vehicle system are LiDARs and a positioning system (2) calibrated together with a gated camera image (1).

A positioning system with RTK (Real-time kinematic positioning) GNSS, inertia unit and vehicle odometry enable the connection of landmarks from the vehicle's local coordinate system to the HD map global coordinate system.

Figure 5.1: Vehicle demonstration sensor system with gated camera (1) and LiDAR with positioning system (2)

Figure 5.2 shows an example how real road environment demonstration is connected between simulator and DigitalTwin HD map environments. Measured landmarks, i.e. traffic signs and poles, can be detected and measured in the real road environment and imported into the HD map. The simulator environment imports the HD map data as part of the simulation model for visibility analysis with visibility data collected by the vehicle. This data can also be imported from the HD map to back to an autonomous vehicle, where the information can be used to estimate the sensor system performance in adverse weather conditions.

Based on this information, the vehicle can update position data more reliability from different sensor systems.

Figure 5.2 also shows how traffic poles (1) and road railing positions can be imported from the HD map into vehicle systems. In this example, the vehicle measures landmark visibility with a LiDAR and gated camera system and updates visibility estimation back to the HD maps. With the updated HD map visibility model, the simulator weather model can be updated real-time.

Figure 5.2: HD map information connected real driving environment. Road elements like traffic poles (1) and road railing (2) can be downloaded HD map as targets to vehicle perception system

6 Conclusion

This chapter concludes the report by listing the key conclusions. Chapter 3 depicts the main elements of the whole autonomous driving framework, i.e. how the digital 3D virtual environment is connected with the real world using the simulation environment. This chapter points out that there are two major 3D computer graphics game engine environments in the market to use. Making the selection between these two on an early phase of the project outlines the options for decisions that must be done later on. Tools and software are typically developed either for Unreal or Unity. Format for the HD map itself is recommended to be either Lanelet2 or OpenDRIVE. As a sub problem of the SLAM chapter presents the point cloud localizing. It also presents the general Ibeo location engine architecture. The chapter concludes with the table comparing conventional and HD maps.

Chapter 4 presents the generally accepted layered structure of the HD maps. Currently it is agreed that HD map is made up of 5 or 6 layers. Here are mentioned the following layers: geometric, semantic, road/lane network, experiental and realtime (from lowest to top). Also, a table presenting currently available HD map industry standards is presented. It lists the main features of the following: Autoware Vector Map, NDS, OpenDRIVE and Lanelet2. Furthermore, it points out that a remarkable benefit is gained when HD data sources are provided via Web Feature Services. This enables transparent data flow from source to the end-user i.e. all new features e.g. on the geometric level are immediately in use in the field.

List of abbreviations

Abbreviation	Meaning	Explanation
Autoware	Autoware	Open source software stack for self-driving vehicles
GeoJSON	GeoJSON	Open standard format for describing geographical features
GIS	Geographic Information Systems	Database containing geographic data for estimating and visualizing features
GNSS	Global navigation satellite system	Satellite based navigation system for providing geo-spatial positioning
HD Map	High-Definition map	Highly accurate map data. Precise positions of lanes and other road elements
IMU	Inertial Measurement Units	Electronics which senses vehicle's orientation and driving vector
Lanelet2	Lanelet2	C++ library for handling map data in the context of automated driving
OpenDRIVE	OpenDRIVE	Open format specification to describe a road network's logic
ROS2	Robot Operating System	Dedicated software libraries and tools for robotic solutions
SLAM	Simultaneous localization and mapping	Updating mapping data with constructing new elements in unknown environment
WFS	Web Feature Service	Provides an interface requesting for geographical features across the web

References

[1] Ritter, W., Bijelic, M., et. al. 2021. Multi-Sensorial Perception System Architecture, Requirements and Use Cases. AI-SEE project deliverable 2.1.

[2] Ritter, W., Sclömicher, T. et. al. 2021. Individual Sensor Systems and Interfaces. AI-SEE project deliverable 2.2.