ROADVIEW

Robust Automated Driving in Extreme Weather

Project No. 101069576

Deliverable 5.7

SW on Improved Localization Using High-density Map Updating – Final report

WP5 – Robust perception system

Authors	Petri Manninen (FGI), Tarmo Kivioja (FGI), Heikki Hyyti (FGI)
Lead participant	FGI
Delivery date	29 August 2025
Dissemination level	Public
Type	Document, report

Version 1.0

Co-funded by the European Union

Co-funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or European Climate, Infrastructure and Environment Executive Agency (CINEA). Neither the European Union nor the granting authority can be held responsible for them. Project grant No. 101069576.

UK participants in this project are co-funded by Innovate UK under contract No.10045139.
Swiss participants in this project are co-funded by the Swiss State Secretariat for Education, Research and Innovation (SERI) under contract No. 22.00123.

Revision history

Author(s)	Description	Date
Heikki Hyyti (FGI)	Draft deliverable	12/05/2025
Petri Manninen (FGI), Heikki Hyyti (FGI), Tarmo Kivioja (FGI)	Initial version	26/08/2025
Kimmo Kauvo (VTT), Fuat Akça (FORD)	Review comments	28/08/2025
Petri Manninen (FGI), Heikki Hyyti (FGI), Tarmo Kivioja (FGI)	Final version (v1.0)	28/08/2025
Johannes Ripperger (accelCH)	Final checks and formatting	29/08/2025

Contents

List of Figures

List of Tables

Partner short names

HH	Hogskolan Halmstad
LUA	Lapin Ammattikorkeakoulu OY
FGI	Maanmittauslaitos – Finnish Geospatial Research Institute
VTT	VTT Technical Research Centre of Finland Ltd
FORD	Ford Otomotiv Sanayi A. S.
accelCH	accelopment Schweiz AG
AVL	AVL Software and Functions GmbH

Abbreviations

ATE	Absolute Trajectory Error
D	Deliverable
EA-NDT	Environment-Aware Normal Distributions Transform
GNSS	Global Navigation Satellite System
HD	High Definition
ICP	Iterative Closest Point
INS	Inertial Navigation System
LOAM	Lidar Odometry and Mapping
NDT	Normal Distributions Transform
NMEA	National Marine Electronics Association
PPK	Post-Processed Kinematic
PPS	Pulse Per Second
SE-NDT	Semantic-assisted Normal Distributions Transform
SLAM	Simultaneous Localization and Mapping
WP	Work Package

Executive summary

This deliverable defines the final version of the software and algorithms developed for positioning an autonomous vehicle in all weather conditions on a High Definition (HD) map developed in WP5 Task 5.4. This deliverable extends the results obtained in the initial version of the software and algorithms published in Deliverable D5.6 and in the publication of the Normal Distributions Transform (EA-NDT) based HD-mapping method (Manninen, Hyyti et al., 2022).

EA-NDT uses semantic information to filter dynamic objects of the environment from the map and to conclude how to process the static objects. The proposed method is designed to automatically compress the information into two types of detected static objects: planar objects (such as walls, fences, traffic signs, and ground plane) and cylindrical objects (such as tree trunks and poles). Only the point cloud from these aforementioned static objects around the road environment is used to build a map for positioning the vehicle. The proposed method uses a probabilistic representation of the environment to withstand more noise and errors in the input point cloud collected with a lidar sensor mounted on top of the vehicle. Compared to the original NDT proposed by Magnusson et al. (2007), EA-NDT leverages semantic segmentation to filter and divide the point cloud, considering the geometry of the environment. It allows compression of the map representation because of a more optimal representation of the original data. Similarly, it means that less data is needed for an equally accurate map representation of the environment.

The work has two separate parts: 1) the building of the EA-NDT map, and 2) positioning on the map.

Objectives

This deliverable's objective is to test if the compressed map representation of EA-NDT will be justified by the positioning performance. We compare the positioning performance of EA-NDT against original NDT while simultaneously considering the memory usage of the map representation in real driving conditions.

Methodology and implementation

The map preprocessing part is tested with an entropy metric, giving a measure of point cloud noisiness. A smaller entropy indicates a better-fitting point cloud. A map file size is used to compare how much smaller amount of data EA-NDT uses with respect to the original NDT map. Finally, positioning accuracy is tested by comparing the map-based positioning methods against online GNSS positioning.

We use real data from a FORD Truck driving on Turkish highways to build the map and to position the vehicle on the map. All positioning methods are compared to the best possible estimate of the ground truth trajectory measured during the tests with a Novatel high-precision GNSS-INS device and post-processed afterwards using base-station data in Novatel Inertial Explorer software.

Outcomes

All compared map-based positioning methods were able to successfully measure the absolute position of the vehicle at least 82 % of the time. Unfortunately, none of the EA-NDT variants were able to position with the required accuracy 100% of the time, thus there is still room for improvement in the method. The best proposed EA-NDT method using a 4-meter source cell size for positioning is able to position on average with errors of 0.42 m and with average angular errors of 0.34 degrees. It is successful in positioning 94.2 % of the time in the tests. All NDT variants were able to position 100% of the time and got better results than EA-NDT in this test. The major difference between the methods is that the EA-NDT utilises semantic segmentation in map building. Unfortunately, the quality of semantic segmentation used in the test was not optimal.

The size of the EA-NDT map is only 8.6 MB in a computer memory, whereas the size of the original NDT is 27.3 MB. Compared to NDT, the EA-NDT map uses only 31,5 % of the needed space.

Next steps

We aim to demonstrate the EA-NDT mapping method in Demo 4 with the FORD Truck on Turkish highways, in Demo 2 with the VTT vehicle in Muonio Lapland, Finland, and in Demo 5 with the AVL demo vehicle in the final demonstration on the AVL test track, Germany. In these tests, we aim to use an environment where the semantic segmentation is expected to work better. In addition, we are preparing two publications on the topic. The first is related to the preprocessing methods needed to align the point cloud data precisely with lidar odometry and graph optimization. The second paper is related to a real-time capable and robust positioning method using online lidar data and inertial measurements to position the vehicle on the HD map.

1 Introduction

Autonomous vehicles need a precise position on the road. However, they cannot trust only GNSS positioning (Zaidi and Suddle, 2006; Wang, 2002). Positioning on a map is a valid alternative solution. In autonomous driving these maps are typically called High-Definition (HD) maps (Liu et al., 2020). HD map positioning should also work in changing environments and in all weather conditions, such as in rain, fog, and snow. In addition, saving a map from large areas uses a considerable amount of memory. Therefore, the map needs to compress information efficiently while robustly and accurately positioning the vehicle in difficult weather conditions in which the environment might appear different compared to the time instance when the map was built.

Lidar scan registration is an accurate technique to fit lidar point clouds, and it can be used to define the position of a lidar scan in an HD map. Iterative Closest Point (ICP) by Besl and McKay (1992) is one of the first registration methods, using unprocessed point cloud data. However, the memory usage of an uncompressed point cloud map used by ICP registration is too high to be stored in the memory of the vehicle. Feature-based methods, such as Lidar Odometry and Mapping (LOAM) by Zhang and Singh (2014), have the challenge of finding features that are robust against changes in the environment due to weather. Normal Distributions Transform (NDT) (Magnusson et al., 2007) provides a probability-based approach for the registration and divides the point cloud into equally sized 3D grid cells to model each cell with a unique normal distribution. NDT compresses the information significantly and includes the uncertainty in the registration, but it does not consider the environment at all. Zaganidis et al. (2017) proposed semantic-assisted NDT (SE-NDT), that includes the use of semantic information to improve the data association in NDT registration. However, SE-NDT doesn't consider the geometry of the environment either, thus, the distributions at the borders of the objects can have undesired effects.

Similarly to SE-NDT, EA-NDT by Manninen, Hyyti et al. (2022) also uses semantic segmentation, which is the classification of the points into predefined classes (Zhang et al., 2019). However, EA-NDT leverages semantic information even further to define planar and cylindrical objects of the environment for more optimal modeling of the data with normal distributions. Simultaneously, the use of planar and cylindrical objects instead of 3D grid division leads to higher compression of the map. An illustration of the EA-NDT HD map is shown in Figure 1. Moreover, the use of semantic information enables filtering of dynamic objects, and only the relevant information for positioning is included in the map. The use of semantic information is also beneficial since this allows the weighing of different semantic classes based on how confident they are considered for the registration.

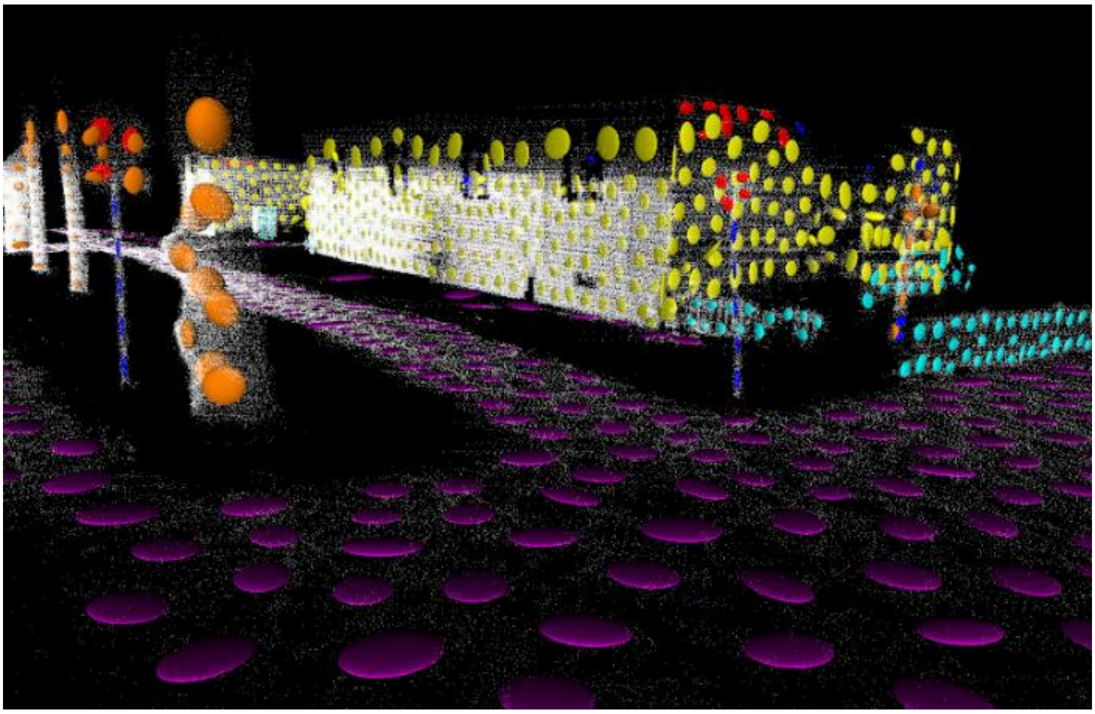

Figure 1. An illustration of an EA-NDT HD map.

2 Background

2.1 Survey of the relevant point cloud registration methods

Iterative closest point (ICP) is a family of point-based point cloud registration methods. A generic formulation of ICP is as follows. Given a source and target point cloud as well as an initial guess at a transformation that aligns the source with the target, first of all, optionally filter the input point clouds. Then, repeatedly transform the source point cloud, compute corresponding point pairs, compute weights for the pairs, and find the transformation that minimizes an error metric between the pairs. The procedure is repeated until some convergence criterion is met. Countless ICP variants exist, differing in how they implement these steps. A thorough review is provided by Pomerleau et al. (2015).

The first published ICP methods are point-to-point ICP by Besl and McKay (1992), and point-to-plane ICP by Chen and Medioni (1992). Segal et al. (2009) show that these can be viewed as special cases of their plane-to-plane variant called generalized ICP (GICP), which remains one of the most popular ICP variants to date.

KISS-ICP by Vizzo et al. (2023) represents the state of the art in the ICP family of registration methods. Its good performance across many different types of datasets is demonstrated using the KITTI (Geiger et al., 2012), MulRan (Jeong et al., 2018), NCLT (Carlevaris-Bianco et al., 2016), and Newer College (Ramezani et al., 2020) datasets. It extends point-to-point ICP with a few steps that improve the registration results in terms of accuracy, speed, and robustness to noise. Input point clouds are spatially downsampled with a voxel grid structure, which makes all further processing faster. The target point cloud used in each registration is a local map accumulated from previously successfully registered point clouds. This gives the map more detail, improving the registration accuracy. Furthermore, maintaining a local map drastically reduces drift over consecutive registrations. In order to mitigate the effects of wrong point correspondences, KISS-ICP adaptively maintains a search range for correspondences and substitutes the original point-to-point ICP cost function with a formulation more robust to outliers.

The weakness of ICP in the context of large-scale mapping is that the map needs to be stored as a point cloud, which is inefficient because a large and detailed point cloud requires a large amount of storage space. Furthermore, performing ICP registration in real time with such a cloud is infeasible or at least difficult, due to the large number of points. It is possible to sample or discretize the data before ICP registration, but the quality of the registration might degrade due to the loss of important information.

The normal distributions transform (NDT) provides a more compact representation for mapping and proposes to use normal distributions to model the discretized data to avoid information loss. It was originally presented in 2D by Biber and Strasser (2003) with a more rigorous mathematical treatment by Biber et al. (2004) and independently adapted to 3D by Takeuchi and Tsubouchi (2006) and Magnusson et al. (2007). In NDT, the point cloud is discretized into small regions called cells, each of which is approximated by a local normal distribution. The discretization of these cells is done by dividing the point cloud into fixed-size cells by a 3D voxel grid. While simple and computationally efficient, this approach can lead to issues at the cell boundaries. More sophisticated discretization strategies include segmented region growing (Das & Waslander, 2014), supervoxel segmentation (Kim & Lee, 2016), k-means clustering (Magnusson, 2009, p. 80 & 183; Jiang et al., 2019) and Expectation-Maximization (Eckart et al., 2016; Tabib et al., 2018). Note that some authors prefer to call their method Gaussian mixture model (GMM) registration instead of NDT registration, perhaps to highlight differences to the regular NDT registration method.

The traditional way to do NDT registration is point-to-distribution NDT (P2D-NDT), where the source point cloud is aligned with an NDT representation of the target point cloud. Stoyanov et al. (2012) proposed the distribution-to-distribution NDT (D2D-NDT) registration method, where an NDT representation is used for both the source and target point clouds, improving registration speed while still providing accurate registration results.

Zaganidis et al. (2017) propose to incorporate semantic information extracted from the point cloud into the NDT registration pipeline. In semantic-assisted NDT (SE-NDT), the point cloud is first labeled with some method, then segmented into classes according to the labeling, and finally, an NDT representation is computed for each class separately. When registering the SE-NDT maps, only cells with the same class are considered for nearest neighbor search. The choice of the labeling method is important for the accuracy of SE-NDT. The authors use a geometric smoothness measure to segment the point cloud into two classes, resulting in improved registration accuracy compared with D2D-NDT, especially for point clouds with large overlap. Zaganidis et al. (2018) pair the same approach with classification based on a deep neural network, yielding further improvements. Cho et al. (2019) propose incorporating the uncertainty associated with the labels into the registration framework.

2.2 Graph-based SLAM

In robotics, simultaneous localization and mapping (SLAM) is the task of building a map of an unknown environment while at the same time maintaining an estimate of one's own pose in relation to that map. Grisetti et al. (2012) classify the many SLAM solutions that exist under two categories: filtering or online SLAM, where the map and pose estimate are continuously refined as new measurements come in, and smoothing or full SLAM, where the full set of measurements is available at once for trajectory estimation and map building. Graph-based SLAM methods, first proposed by Lu and Milios (1997), are smoothing methods where the SLAM problem is formulated as a directed graph whose nodes represent robot poses or landmarks, and the edges between the nodes represent measurements that somehow constrain the poses relative to each other. For example, an edge could represent a transformation between two poses, estimated using lidar odometry. Because measurements include uncertainty, the constraints may not be consistent. For example, chaining the transformations over a cycle in the graph should result in the identity transformation, but will likely be more or less off. The central problem in graph-based SLAM is then to find a configuration for the nodes that minimizes an error metric over the whole graph. GTSAM by Dellaert et al. (2022) is a popular C++ library for graph SLAM and was used in this work.

3 Methods

This section explains, first, the building of the Environment-Aware (EA)-NDT HD Map in Section 3.1, and second, the positioning on the proposed map using real-time lidar measurements from the vehicle in Section 3.2. EA-NDT improves compression of the traditional NDT map representation by leveraging semantic segmentation in a more optimal division of the point cloud into cells. Semantic segmentation is used to filter out dynamic objects of the environment. The rest of the objects are considered as planar (ground, building, fence, traffic sign) or cylindrical (poles, tree trunks) objects. Planes are fitted for planar objects, while cylindrical objects are treated as they are. The purpose of finding planes and cylinders is to minimize the number of cells for each object and to maximize the information of the distribution in each cell. The reasoning behind this is that a distribution containing a small variance at least in one direction is very informative, whereas a distribution with high variance in all directions contains very little information about the correct location of the points it represents.

3.1 Preprocessing

In this deliverable, we use lidar data recorded with Velodyne VLS-128 (Velodyne LiDAR Inc., 2020) and GNSS-INS measurements recorded with Novatel PwrPak7 (Novatel, October 2020) that are postprocessed with Novatel Inertial Explorer (Novatel, February 2020) to obtain a postprocessed kinematic (PPK) solution for GNSS and IMU measurements. To build a georeferenced point cloud from the data, as a preprocessing step, we perform motion compensation (see Section 3.1.1) of the lidar scans, lidar odometry (see Section 3.1.2) and graph optimization (see Section 3.1.3).

3.1.1 Motion compensation

The lidar scanner in the vehicle is rotating with a 10 Hz frequency, and therefore, each lidar scan is obtained within a period of 100 milliseconds. Motion compensation is used to compensate for the movement of the vehicle into the lidar scan within that 100 ms period. We perform the motion compensation of each lidar scan based on the Inertial Explorer PPK solution. Velodyne VLS-128 is time synchronized with the PPS signal and NMEA messages received from Novatel PwrPak7 to provide an accurate timestamp for each point in the lidar scan. Based on the timestamp of the lidar scan frame and each individual point, we interpolate the relative velocities and angular velocities from the PPK solution.

3.1.2 Lidar odometry

There are typically at least small errors in the PPK solution or even in the extrinsic calibration of the sensors. Therefore, the PPK solution is not precise for the accurate georeferencing of a point cloud alone. To improve the alignment of the data, we perform lidar odometry. First, we use HH semantic segmentation (see Deliverable D5.2) to acquire pointwise semantic segments for each lidar scan. The semantic segmentation is used to filter out dynamic obstacles from the point cloud before performing lidar odometry for the scans. For lidar odometry, we use KISS-ICP (Vizzo et al., 2023) registration, and we also register loop closures that are within 25 meters based on the PPK solution.

3.1.3 Graph optimization

Lidar odometry alone accumulates error along the trajectory, and the quality of PPK solution-based georeferencing alone is not good enough because the solution would drift out of the true position. Therefore, to improve the quality of a georeferenced point cloud, we use graph optimization to fuse the PPK solution and lidar odometry estimates. Namely, we used GTSAM by Dellaert et al. (2022) to obtain the final georeferenced point cloud. We modeled the XYZ-coordinates of the PPK solution as graph nodes and the lidar odometry with 6-DoF as edges between the nodes.

3.2 Building EA-NDT HD map

This section explains the required steps to build the EA-NDT HD map: semantic segmentation, instance clustering, primitive extraction, cell clustering, and computation of NDT parameters (mean and covariance) of each cell. The outputs of these steps are semantic information, instances, primitives, cells, and the EA-NDT HD map, as shown in the overall visualization in Figure 2. In this experiment, we assume that the trajectory to align the lidar scans into a point cloud map has already been solved, and it can be used to align the scans to build an EA-NDT HD map.

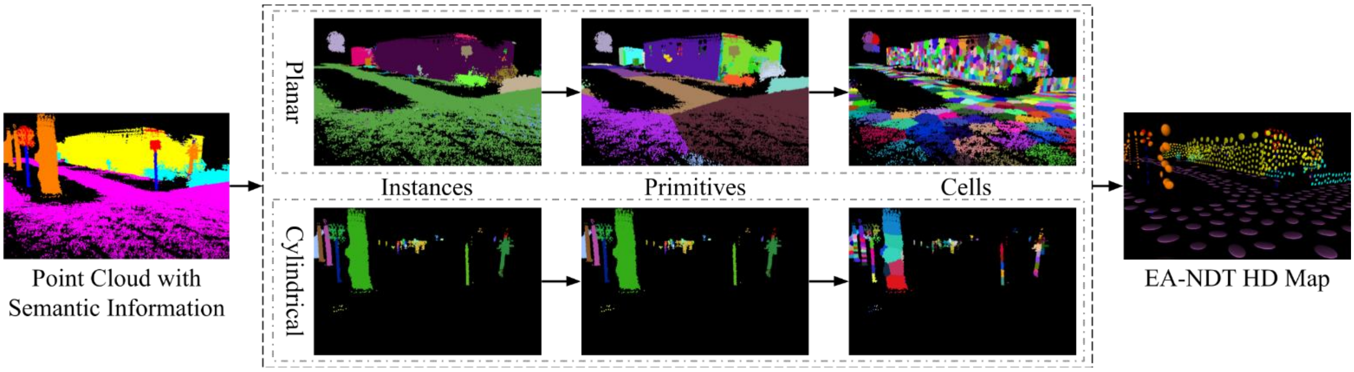

Figure 2. An illustration of the intermediate step results and final output of the process in building an EA-NDT HD map.

3.2.1 Semantic segmentation

In the process of generating an EA-NDT HD map for an area, the collected point cloud is first semantically segmented. In the work by Manninen, Hyyti et al. (2022), the semantic segmentation is achieved by using RandLA-Net model (Hu et al., 2020) pre-trained with SemanticKITTI dataset (Behley et al., 2019). SemanticKITTI is collected with Velodyne HDL-64 lidar (Velodyne LiDAR Inc., 2017), which is a 3D rotating lidar measuring with 64 vertical laser beams. In this deliverable, we instead use a semantic segmentation deep-learning model trained for Velodyne VLS-128, which measures with 128 vertical beams, provided and trained by HH in the ROADVIEW project and reported in Deliverable D5.2. The training data for this effort was collected in ROADVIEW with the FGI Arvo vehicle in Finland, and the data was labelled by LUA.

3.2.2 Instance clustering

Instance clustering is based on the Euclidean region growing algorithm (Trevor et al., 2013), which divides each semantic segment into spatially separate instances such as wall segments, poles, or tree trunks. The Euclidean region growing algorithm requires two parameters, distance threshold d_0 and minimum number of points n_p per instance.

Table 1. Values for the parameters used in EA-NDT instance clustering and primitive extraction.

		Other	Ground
Instance clustering	Distance threshold d_0	30 cm	50 cm
	Minimum points n_p	10	3000
Primitive extraction	Normal distance weight α	π/4	π/4
	Distance threshold d_1	15 cm	30 cm

We use the parameters defined by Manninen, Hyyti et al. (2022) and presented in Table 1. In addition to grouping points in the same objects or parts of objects, the instance clustering filters noise from the point cloud (Manninen, Hyyti et al., 2022).

3.2.3 Primitive extraction

In Primitive Extraction, previously found instances are modeled as primitives. Tree trunk and pole instances are modeled as cylindrical primitives, and traffic sign instances as planar primitives. All the other semantic labels are processed as planar primitives. These were extracted by using a Random Sample Consensus (Fischler & Bolles, 1981) based normal plane fitting algorithm after subsampling the instance with an averaging 10 cm voxel grid and estimating normals for each remaining point from 26 nearest neighboring voxels (i.e., 3×3×3 voxels area around the point). The plane fitting algorithm requires two parameters, normal distance weight α and a distance threshold d_t. We used the parameters defined by Manninen, Hyyti et al., 2022) and presented in Table 1. Although all planar primitives are processed similarly, for ground instances, an implementation of the K-means++ algorithm (Arthur & Vassilvitskii, 2007) was first used to divide the ground instances into primitives that are 100 m^2 roundish areas.

3.2.4 Cell clustering

In Cell Clustering, the primitives are divided into cells of about the same size. The K-means++ algorithm (Arthur & Vassilvitskii, 2007) is used to cluster the cells from the primitives. However, K-means++ requires the number of clusters as a predefined parameter. Manninen, Hyyti et al. (2022) have solved this by using a data-based approximation method to compute a predefined number of clusters for each primitive. The method to predefine a suitable number of clusters is based on approximating the area of the primitive's surface and dividing it by the area of the required cell size.

3.2.5 NDT computation

In the final stage, a mean and covariance are computed for the points in each cell found in the cell clustering stage. Then, similarly to NDT, the mean and covariance of each cell are stored into a hash map, in contrast to Manninen, Hyyti et al. (2022), which utilized an octree data structure to represent the EA-NDT HD Map. In addition to the mean and covariance, in EA-NDT, the average label vector is computed for each cell and included in the hash map.

3.2.6 Map search structure

The map search structure is based on a 3D voxel grid where each grid leaf contains all the EA-NDT cells located within that leaf's area. More specifically, EA-NDT cells that have their mean value inside a grid leaf are associated with that leaf. Since in EA-NDT the computation of NDT distributions is not based on a 3D grid, it is possible that more than one cell will exist in the same grid leaf.

3.3 Positioning on the HD map

This section explains how the positioning in the EA-NDT HD map is performed and how the positioning performance was compared.

3.3.1 Positioning on the EA-NDT map

Positioning is based on registering a lidar scan in the EA-NDT map. For each point in the lidar scan, the closest leaf in the voxel grid is searched together with all its direct neighboring leaves in a 3×3×3 neighborhood. For EA-NDT, only the cells that match the semantic label of the point are considered. The registration is then performed as described in Section 3.3.2. Figure 3 presents EA-NDT HD map positioning with a simple data flow diagram.

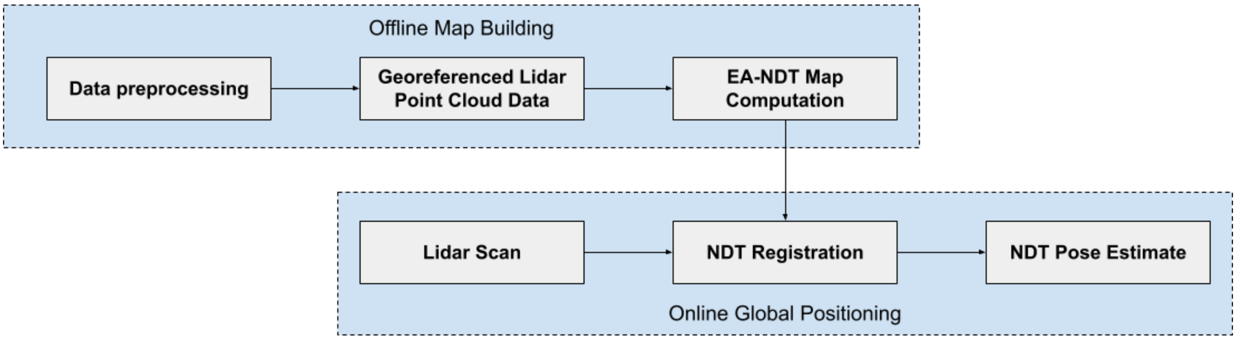

Figure 3. A simple data flow diagram of positioning with the EA-NDT HD map.

3.3.2 NDT registration

The NDT registration is performed using D2D-NDT for EA-NDT and for the NDT map. In EA-NDT, the semantic classes are taken into account such that corresponding NDT cells are only searched for among those with matching classes. D2D-NDT registration is carried out with three different values for source NDT cell size: 1 m, 2 m, and 4 m. The map (target for registration) is computed at 1 m resolution. The implementation, which allows asymmetric target and source cell sizes of D2D-NDT registration, is developed in-house at FGI.

3.3.3 Comparison metrics

Quality of the pre-processed point cloud for mapping

The quality of positioning is measured indirectly through point cloud entropy. All scans are accumulated into global coordinates based on the estimated poses, after which the entropy is computed. Entropy can be thought of as a measure of randomness present in the point cloud. The idea is that a correctly registered cloud minimizes the randomness as the points are heavily correlated with the underlying geometry, instead of being uniformly distributed in space. Therefore, a lower entropy implies better positioning. The entropy measure used is derived by Putkiranta (2020) as

$$E = -\sum_{i=1}^{n}\sum_{j=1}^{n} \exp\left(-\sigma^{-2}\left(x_i - x_j\right)^{\top}\left(x_i - x_j\right)\right), \tag{1}$$

where n Is the number of points in the cloud, x_i and x_j are points in the cloud, and σ is a free parameter. As computing every point-to-point distance is not feasible, and neighboring points give practically all of the contribution to the metric, E is approximated by evaluating the summand for the 80 closest points x_j for each point x_i. The value used for σ was 0.04.

Positioning

The registration error is defined as

$$\boldsymbol{T}_{err} = \boldsymbol{T}_R^{L^{-1}} \boldsymbol{T}_R^M \boldsymbol{T}_M^L, \tag{2}$$

where $\boldsymbol{T}_R^L$ is the pose of the lidar in the reference frame R, $\boldsymbol{T}_M^L$ is the estimated pose of the lidar in the map frame M, and $\boldsymbol{T}_R^M$ is the offset between the map and reference frames. The transformation chain to define $\boldsymbol{T}_{err}$ is shown in Figure 4.

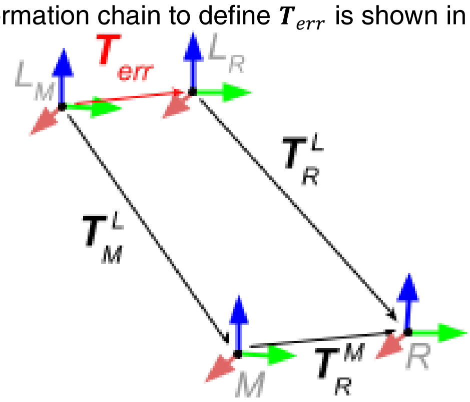

Figure 4. Illustration of the transformation chain to define the registration error after registering the scan. *R* defines the reference frame, and *M* defines the map frame.

To compare the scan registration performance, we define the absolute position error

$$p_{err} = \|\boldsymbol{t}_{err}\|_2 \,, \tag{3}$$

and absolute orientation error

$$\theta_{err} = \angle\boldsymbol{R}_{err} \,, \tag{4}$$

where ∠(.) denotes taking the angle of angle-axis representation for a rotation matrix (Zhang, Z., & Scaramuzza, D., 2018). Translation error vector $\boldsymbol{t}_{err}$ and rotation error matrix $\boldsymbol{R}_{err}$ are defined as parts of the registration error

$$\boldsymbol{T}_{err} = \begin{bmatrix}\boldsymbol{R}_{err} & \boldsymbol{t}_{err} \\ 1 & 0\end{bmatrix}, \tag{5}$$

The positioning errors are measured using absolute trajectory error (ATE), which is defined by Zhang & Scaramuzza (2018). For position, it is defined:

$$ATE_{pos} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left\|\boldsymbol{t}_{err,i}\right\|_2^2}\,, \tag{6}$$

and for rotation:

$$ATE_{rot} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left\|\angle\boldsymbol{R}_{err,i}\right\|^2}\,, \tag{7}$$

in which N is the number of data samples in the test set indexed by i.

Size of the map

The size of the EA-NDT map is computed through the memory requirement of the map. As defined in Deliverable D5.6, the size of each cell for NDT consists of x, y, z, intensity, covariance matrix upper triangle (6 values), and number of points. Each value uses 4 bytes, which results in 11 × 4 = 44 bytes in total. For EA-NDT, an extra label value of 4 bytes is added, resulting in 48 bytes per cell in total. The total size of the map is then computed as:

$$Size = N_{cells} * Size_{cell}, \quad \begin{cases} Size_{cell} = 44\ \mathrm{B} & \text{for NDT} \\ Size_{cell} = 48\ \mathrm{B} & \text{for EA-NDT} \end{cases}, \tag{8}$$

We used the same metrics here that are defined in the final KPI specifications in Deliverable D8.7.

4 Experiments

In this deliverable, we use data recorded with a Ford truck on the Turkish highway ROADVIEW demo site (Figure 5). The data consists of two recordings from the same highway in Turkey near Istanbul, recorded in February and June 2025. The first dataset from February is used to build an HD map to which the vehicle is positioned. The second dataset is used to test the positioning on the map. The data used in this experiment consisted of lidar data recorded with Velodyne VLS-128 (Velodyne LiDAR Inc., 2020) and GNSS-INS measurements recorded with Novatel PwrPak7 (Novatel, October 2020) that are postprocessed with Novatel Inertial Explorer (Novatel, February 2020) to obtain a postprocessed kinematic (PPK) solution for GNSS and IMU measurements. The IMU sensor in Novatel PwrPak7 is Epson EG320N. Both datasets are aligned as explained in Section 3.1.

In the experiments, we used Graph-SLAM optimized best estimate of the trajectory as a ground truth position. The best solution of the test trajectory (Ford 2, June 2025) data was used to generate the ground truth positions using the pre-processing methods. As shown in Table 2, the combined Graph-SLAM and post-processed GNSS-INS trajectory had the smallest entropy metric, indicating that it had the best available quality of point cloud and thus the best fitting trajectory at all positions.

The EA-NDT and NDT maps for map-based positioning were made using the lidar and GNSS+INS data collected in the first Ford data set (Ford 1, February 2025).

For comparing to online GNSS positioning accuracy, we use a real-time GNSS IMU solution in the Ford Truck from the test data (Ford 2, June 2025). The sensor is the same high-end Novatel PwrPak7 GNSS-INS sensor, but we used only online GNSS measurements without post-processing.

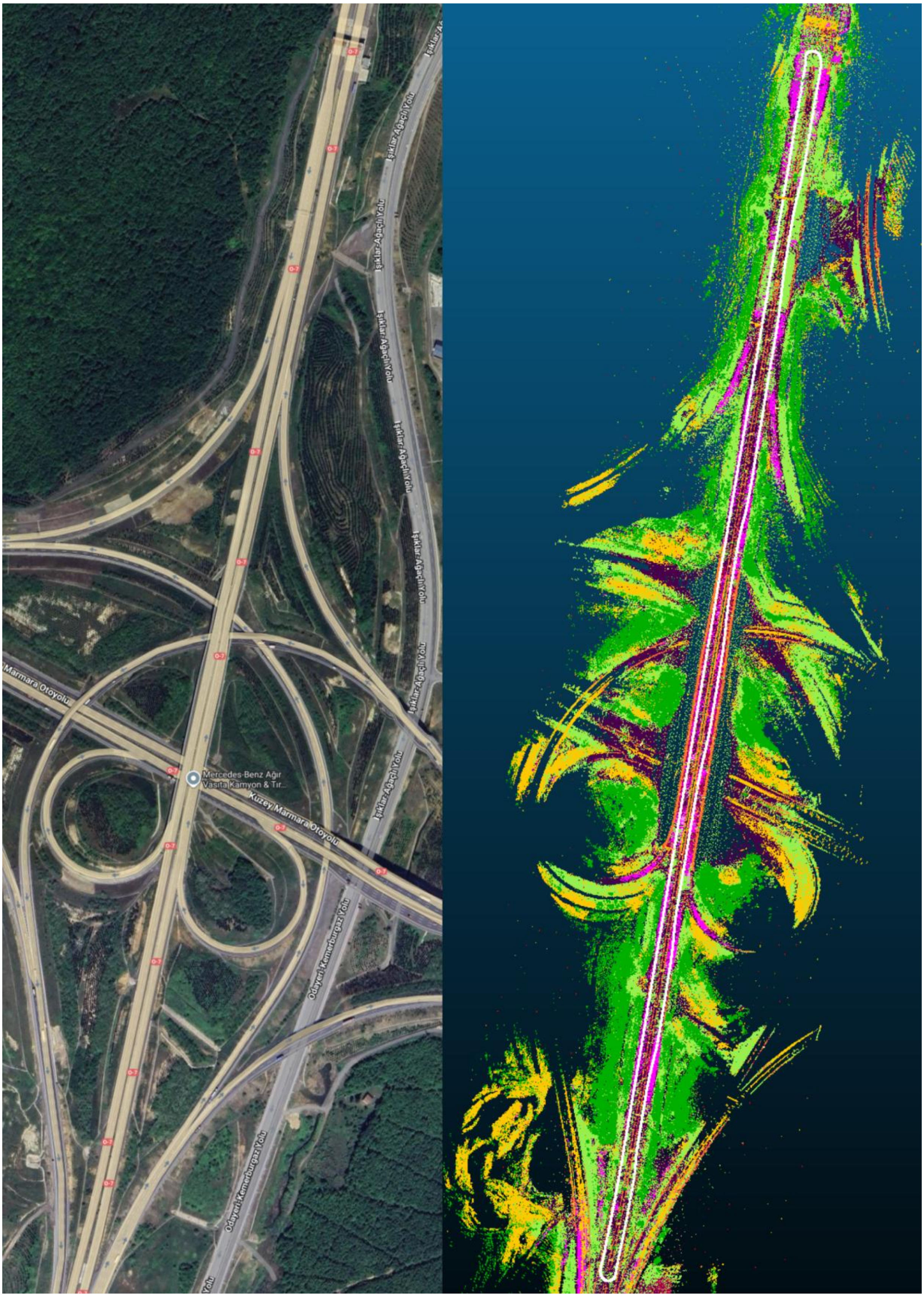

Figure 5. A map of the Turkish Highway is shown on the left side (© Google) and graph-optimized semantically segmented point cloud data of the same area is shown on the right. The driven test trajectory is depicted in white color.

5 Results

5.1 Data entropy

Entropy metrics of the data set after different pre-processing steps are shown in Table 2. As defined in Section 3.3.3, smaller entropy metric values describe a better point cloud fit, indicating less randomness in the data. For both datasets, the PPK GNSS-INS-based solution results in the highest entropy metric value, indicating the worst point cloud fit. For both datasets, the lidar odometry solution significantly reduces the entropy score, improving the point cloud fit. By incorporating the loop closures, the entropy score is reduced by the Graph-SLAM solution. For the first dataset, the entropy score slightly increases when the PPK GNSS-INS solution is fused with the lidar odometry (Graph-SLAM + PPK GNSS-INS in Table 2) estimates indicating that the fused solution is not optimal. However, for the second dataset, the entropy decreases when the PPK GNSS-INS solution is fused with the lidar odometry estimates. This difference can be explained because, for the first dataset, the PPK GNSS-INS solution was visually less accurate compared to the second dataset. This can also be observed in the entropy score of the first dataset PPK GNSS-INS solution. In general, it can be observed that the entropy score, and therefore the quality of the georeferenced point cloud, significantly improves with the suggested preprocessing steps.

Table 2. Entropy metric computed for both datasets with different trajectory estimates. The smallest entropy value is highlighted in blue, indicating the best point cloud quality. The Graph-SLAM solution uses lidar odometry and loop closure estimates to align the data. The Graph-SLAM + PPK GNSS-INS fuses lidar odometry and PPK GNSS-INS solution.

Dataset	PPK GNSS-INS	Lidar odometry	Graph-SLAM	Graph-SLAM + PPK GNSS-INS
Ford 1st, February 2025	-2 265 538 911	-3 628 044 909	-4 180 984 897	-3 992 169 168
Ford 2nd, June 2025	-3 379 753 179	-4 005 081 829	-4 217 934 665	-4 477 880 532

5.2 Positioning accuracy

The accuracy of positioning is compared between different EA-NDT versions at different resolutions and the NDT implementation at the same resolutions. In all of these comparisons, distribution to distribution (D2D) variant of the NDT has been used. The statistics of the tests between different methods are shown for horizontal and height errors in Figure 6.

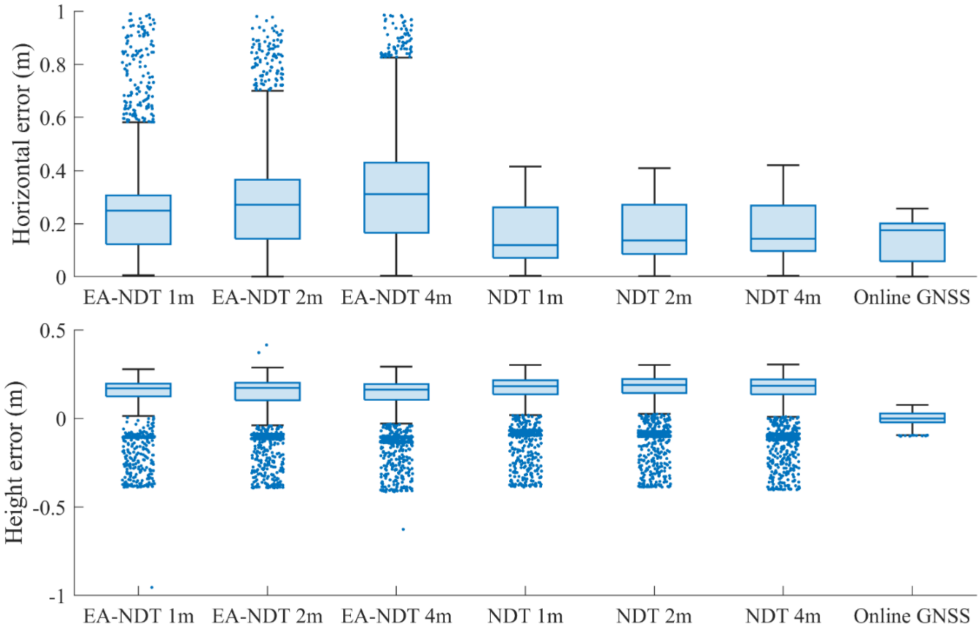

Figure 6. Box-and-whiskers plots of the positioning results for horizontal positioning and for height error, separated for each of the compared methods, EA-NDT-based positioning with 1-, 2-, and 4-meter source cell size, and NDT for comparison using the same cell sizes. Outliers above 1.0 meters are cut out, and their amount is shown in Table 3.

The results for positioning tests are presented in Table 3. The rotation results for the online GNSS are estimated by the Novatel internal sensor fusion algorithm, using also inertial measurements.

Table 3. Results for positioning Ford 2, June 2025 measurements on a map computed from Ford 1, February 2025 data collection. Since none of the NDT methods were successful in all measurements, a success percent is computed for all, and ATEs were computed for all measurements as well as for only successful measurements in which the position error is less than 1.0 meter.

Name	Success %	ATE for position, all measurements (m)	ATE for rotation, all measurements (deg)	ATE for position, succesful (m)	ATE for rotation, succesful (deg)
EA-NDT 1m	81,9 %	5,42	0,80	0,36	0,42
EA-NDT 2m	90,3 %	2,92	0,42	0,38	0,33
EA-NDT 4m	94,2 %	1,93	0,37	0,42	0,34
NDT 1m	100,0 %	0,27	0,28	0,27	0,28
NDT 2m	100,0 %	0,28	0,27	0,28	0,27
NDT 4m	100,0 %	0,28	0,26	0,28	0,26
Online GNSS	100,0 %	0,16	1,37	0,16	1,37

As shown in Table 3 and in Figure 6, the online GNSS is accurately working in this experiment on an open highway scenario. Unfortunately, GNSS does not give a reliable rotation of the vehicle in the world when used only with a single antenna. All HD-mapping methods are working to give a position to the vehicle at least most of the time. The D2D implementation of NDT is really good on this highway data in this test, giving 100% reliability and similar error statistics as the other HD-map-based methods. The best EA-NDT method has the largest cell size of 4m, giving an acceptable solution at 94% reliability. The benefit of HD-mapping-based methods over GNSS methods is that EA-NDT and NDT are able to provide really precise 3D rotation in addition to the position estimate.

All HD-mapping methods (NDT and EA-NDT rows in Table 3) suffer a larger average error compared to the online GNSS solution. We hypothesise that this error is caused by the errors present in the pre-processing step of georeferencing the point cloud. If an error is observed in the positioning in the data from which the map is made, it will be shown as a bias in our results where we position on the map and assume that the map reflects the positions of the objects in the world precisely.

As seen in Table 3, EA-NDT occasionally fails to find the correct location in this test. We hypothesize that this is caused by the limited quality of the semantic segmentation data used in this test. We utilized the semantic segmentation model provided and trained by the HH team in D5.2. However, their deep learning network was trained with data collected in Finland in different conditions using an FGI passenger vehicle during winter. Now the model was used in non-optimal conditions with a Ford truck in a different environment in dry weather. In future tests of ROADVIEW, we hope to have more suitable conditions and better quality of semantic segmentation of the point cloud.

5.3 Map size

The results for the map size are shown in Table 4. As seen in the table, the EA-NDT map is significantly smaller than the equivalent NDT map used to compare the results in the previous section. EA-NDT uses only 31,5 % of the data space.

Table 4. Size of the map for EA-NDT and NDT.

Dataset	EA-NDT (1 m)	NDT (1 m)	Share of EA-NDT size from NDT map of the same size
Ford 1, February 2025	8.6 MB	27.3 MB	31,5 %

6 Conclusions

In this deliverable, we presented the final version of the software and algorithms developed for positioning an autonomous vehicle in all weather conditions on an HD map developed in WP5 Task 5.4. This deliverable extended the results obtained in the initial version of the software and algorithms published in Deliverable D5.6 and in the publication of Normal Distributions Transform (EA-NDT) based HD-mapping method (Manninen, Hyyti et al., 2022).

The proposed EA-NDT uses semantic information to filter dynamic objects of the environment from the map and to conclude how to process the static objects. The proposed method was designed to automatically compress the information into two types of detected static objects: planar objects (such as walls, fences, traffic signs, and ground plane) and cylindrical objects (such as tree trunks and poles). Only the point cloud from these aforementioned static objects around the road environment was used to build a map for positioning the vehicle. The proposed method uses a probabilistic representation of the environment to withstand more noise and errors in the input point cloud collected with a lidar sensor mounted on top of the vehicle. Compared to the original NDT proposed by Magnusson et al. (2007), EA-NDT leverages semantic segmentation to filter and divide the point cloud, considering the geometry of the environment. It allowed significant compression of the map representation because of a more optimal representation of the original data. In our test, the EA-NDT map was only 31,5 % of the size of the compared NDT map. Both of these methods use only a really small fraction of the original point cloud data. The whole mapped highway segment fitted into 8.6 MB of data using EA-NDT.

We tested whether the compressed map representation of EA-NDT will be justified by the positioning performance. We compared the positioning performance of EA-NDT against NDT and online GNSS positioning. The tests were performed on a Turkish highway with a FORD truck. All compared map-based positioning methods were able to position the vehicle on the map most of the time. The best proposed EA-NDT method using a 4-meter source cell size for positioning is able to position on average with errors of 0.42 m and with average angular errors of 0.34 degrees. It is successful in positioning 94% of the time in the tests. The compared NDT methods were able to position the vehicle at all times, proving to be superior in this test.

The limited quality of the EA-NDT positioning leaves plenty of room for improvement in the future. We hypothesize that the limited quality of the semantic segmentation network provided by HH was not optimal. It might have made some areas of the EA-NDT map erroneous, and thus the positioning failed occasionally. Additionally, there seems to be a small offset in the positions given by the HD-map-based positioning methods (both EA-NDT and NDT) when compared to online GNSS positioning, which was marginally better in giving the 3D position. However, map-based methods were better for estimating the rotation of the vehicle. As a future work, we aim to improve the semantic segmentation and the pre-processing steps to remove the errors in the positions of the mapped data.

Next, we aim to demonstrate the EA-NDT mapping method in Demo 4 with the FORD Truck on Turkish highways, in Demo 2 with the VTT vehicle in Muonio Lapland, Finland, and in Demo 5 with the AVL demo vehicle in the final demonstration on the AVL test track, Germany. In addition, we are preparing two publications on the topic. The first is related to the preprocessing methods needed to align the point cloud data precisely with lidar odometry and graph optimization. The second paper is related to the real-time capable and robust positioning method using online lidar data and inertial measurements to position the vehicle on the HD-map.

7 References

Arthur, D., & Vassilvitskii, S. (2007, January). k-means++: The advantages of careful seeding. In *Soda* (Vol. 7, pp. 1027-1035).

Behley, J., Garbade, M., Milioto, A., Quenzel, J., Behnke, S., Stachniss, C., & Gall, J. (2019). SemanticKITTI: A Dataset for Semantic Scene Understanding of LiDAR Sequences. In *Proceedings of the IEEE/CVF international conference on computer vision* (pp. 9297-9307).

Besl, P. J., & McKay, N. D. (1992). Method for registration of 3-D shapes. *In P. S. Schenker (Ed.), Sensor fusion IV: control paradigms and data structures* (Vol. 1611, pp. 586–606). SPIE Publications. https://doi.org/10.1117/12.57955

Biber, P., Fleck, S., & Strasser, W. (2004). A probabilistic framework for robust and accurate matching of point clouds. *Joint Pattern Recognition Symposium*, 480–487. https://doi.org/10.1007/978-3-540-28649-3_59

Carlevaris-Bianco, N., Ushani, A. K., & Eustice, R. M. (2016). University of Michigan North Campus long-term vision and lidar dataset. *The International Journal of Robotics Research*, 35(9), 1023-1035.

Chen, Y., & Medioni, G. (1992). Object modelling by registration of multiple range images*. Image and Vision Computing*, 10(3), 145–155. https://doi.org/10.1016/0262-8856(92)90066-c

Cho, S., Kim, C., Park, J., Sunwoo, M., & Jo, K. (2020). Semantic point cloud mapping of LiDAR based on probabilistic uncertainty modeling for autonomous driving. *Sensors*, 20(20), 5900. https://www.mdpi.com/1424-8220/20/20/5900/pdf

Dellaert, F., & GTSAM contributors. (2022). borglab/gtsam (Version 4.2a8). doi:10.5281/zenodo.5794541

Eckart, B., Kim, K., Troccoli, A., Kelly, A., & Kautz, J. (2016, June). Accelerated Generative Models for 3D Point Cloud Data. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Fischler, M. A., & Bolles, R. C. (1981). Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, *24*(6), 381-395.

Geiger, A., Lenz, P., & Urtasun, R. (2012, June). Are we ready for Autonomous Driving? The KITTI Vision Benchmark Suite. *In 2012 IEEE conference on computer vision and pattern recognition* (pp. 3354-3361). IEEE.

Grisetti, G., Kümmerle, R., Stachniss, C., & Burgard, W. (2011). A tutorial on graph-based SLAM. *IEEE Intelligent Transportation Systems Magazine*, 2(4), 31-43.

Hyyppä, E., Manninen, P., Maanpää, J., Taher, J., Litkey, P., Hyyti, H., ... & Hyyppä, J. (2023). Can the Perception Data of Autonomous Vehicles Be Used to Replace Mobile Mapping Surveys?—A Case Study Surveying Roadside City Trees. *Remote Sensing*, *15*(7), 1790.

Hu, Q., Yang, B., Xie, L., Rosa, S., Guo, Y., Wang, Z., ... & Markham, A. (2020). Randla-net: Efficient semantic segmentation of large-scale point clouds. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (pp. 11108-11117).

Jeong, J., Cho, Y., Shin, Y. S., Roh, H., & Kim, A. (2018, May). Complex urban lidar data set. In *2018 IEEE international conference on robotics and automation (ICRA)* (pp. 6344-6351). IEEE.

Jiang, M., Song, S., Tang, F., Li, Y., Liu, J., & Feng, X. (2019). Scan registration for underwater mechanical scanning imaging sonar using symmetrical Kullback–Leibler divergence. *Journal of Electronic Imaging*, 28(1), 13026. https://doi.org/10.1117/1.JEI.28.1.013026

Koide, K. (2017). ndt_omp. (Github repository). https://github.com/koide3/ndt_omp

Liu, R., Wang, J., & Zhang, B. (2020). High definition map for automated driving: Overview and analysis. *The Journal of Navigation*, *73*(2), 324-341.

Lu, F., & Milios, E. (1997). Globally consistent range scan alignment for environment mapping. *Autonomous robots*, 4(4), 333-349.

Maanpää, J., Taher, J., Manninen, P., Pakola, L., Melekhov, I., & Hyyppä, J. (2021, January). Multimodal end-to-end learning for autonomous steering in adverse road and weather conditions. In *2020 25th International Conference on Pattern Recognition (ICPR)* (pp. 699-706). IEEE.

Magnusson, M. (2009). *The three-dimensional normal-distributions transform: an efficient representation for registration, surface analysis, and loop detection* (Doctoral dissertation, Örebro University).

Magnusson, M., Lilienthal, A., & Duckett, T. (2007). Scan registration for autonomous mining vehicles using 3D-NDT. *Journal of Field Robotics*, 24(10), 803–827. https://doi.org/10.1002/rob.20204

Manninen, P., Hyyti, H., Kyrki, V., Maanpää, J., Taher, J., & Hyyppä, J. (2022, October). Towards High-Definition Maps: A Framework Leveraging Semantic Segmentation to Improve NDT Map Compression and Descriptivity. In *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (pp. 5370-5377). IEEE.

Novatel, Inertial Explorer, February 2020. D18034 Version 9 Brochure.

Novatel, PwrPak7-E1, October 2020. D18496 Version 7 datasheet.

Pomerleau, F., Colas, F., & Siegwart, R. (2015). A review of point cloud registration algorithms for mobile robotics. Foundations and Trends in Robotics, 4(1), 1–104. https://doi.org/10.1561/9781680830255 Biber, P., & Straßer, W. (2003). The normal distributions transform: A new approach to laser scan matching. *Proceedings 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2003)*, 3, 2743–2748. https://doi.org/10.1109/iros.2003.1249285

Putkiranta, P. (2020). Geometric calibration of rotating multi-beam lidar systems. (Master's thesis, Aalto University). https://urn.fi/URN:NBN:fi:aalto-202001261822

Ramezani, M., Wang, Y., Camurri, M., Wisth, D., Mattamala, M., & Fallon, M. (2020, October). The newer college dataset: Handheld lidar, inertial and vision with ground truth. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (pp. 4353-4360). IEEE.

Reid, T. G., Houts, S. E., Cammarata, R., Mills, G., Agarwal, S., Vora, A., & Pandey, G. (2019). Localization Requirements for Autonomous Vehicles. *arXiv preprint arXiv:1906.01061*.

Segal, A., Haehnel, D., & Thrun, S. (2009). Generalized-ICP. *Robotics: Science and Systems*, 2, Article 4. https://doi.org/10.15607/rss.2009.v.021

Stoyanov, T., Magnusson, M., Andreasson, H., & Lilienthal, A. J. (2012). Fast and accurate scan registration through minimization of the distance between compact 3D NDT representations. *The International Journal of Robotics Research*, 31(12), 1377–1393. https://doi.org/10.1177/0278364912460895

Tabib, W., O'Meadhra, C., & Michael, N. (2018). On-Manifold GMM Registration. *IEEE Robotics and Automation Letters*, 3(4), 3805–3812. https://doi.org/10.1109/LRA.2018.2856279

Takeuchi, E., & Tsubouchi, T. (2006). A 3-D scan matching using improved 3-D normal distributions transform for mobile robotic mapping. *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 3068–3073. https://doi.org/10.1109/iros.2006.282246

Trevor, A. J., Gedikli, S., Rusu, R. B., & Christensen, H. I. (2013). Efficient organized point cloud segmentation with connected components. *Semantic Perception Mapping and Exploration (SPME)*, *10*(6), 251-257.

Velodyne LiDAR Inc. (2017, December). HDL-64E S3 High Definition LiDAR Sensor, User's manual and programming guide [Datasheet].

Velodyne LiDAR Inc. (2020, August). Alpha Prime User Manual, 63-9483 Rev. 3 [Datasheet].

Wang, J. (2002). Pseudolite applications in positioning and navigation: Progress and problems. *Positioning*, *1*(03).

Zaganidis, A., Magnusson, M., Duckett, T., & Cielniak, G. (2017). Semantic-assisted 3D normal distributions transform for scan registration in environments with limited structure. *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 4064–4069. https://doi.org/10.1109/IROS.2017.8206262

Zaganidis, A., Sun, L., Duckett, T., & Cielniak, G. (2018). Integrating Deep Semantic Segmentation Into 3-D Point Cloud Registration. *IEEE Robotics and Automation Letters*, 3(4), 2942–2949. https://doi.org/10.1109/LRA.2018.2848308

Zaidi, A. S., & Suddle, M. R. (2006, September). Global navigation satellite systems: a survey. In *2006 international conference on advances in space technologies* (pp. 84-87). IEEE.

Zhang, J., & Singh, S. (2014, July). LOAM: Lidar odometry and mapping in real-time. *In Robotics: Science and systems* (Vol. 2, No. 9, pp. 1-9).

Zhang, J., Zhao, X., Chen, Z., & Lu, Z. (2019). A review of deep learning-based semantic segmentation for point cloud. *IEEE access*, *7*, 179118-179133.

Zhang, Z., & Scaramuzza, D. (2018). A tutorial on quantitative trajectory evaluation for visual (-inertial) odometry. *In 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (pp. 7244-7251). IEEE.