Research article

Autonomous vehicle self-localization based on abstract map and multi-channel LiDAR in urban area

Ehsan Javanmardi, Yanlei Gu *, Mahdi Javanmardi, Shunsuke Kamijo

The University of Tokyo, Japan

ARTICLE INFO

A R T I C L E  I N F O

Article history:
Received 28 July 2017
Received in revised form 25 April 2018
Accepted 9 May 2018
Available online xxxx

Keywords:
Self-localization
Abstract map
Lidar
Velodyne
Planar surface map
Vector map
Urban area

A B S T R A C T

Accurate vehicle self-localization is significant for autonomous driving. The localization techniques based on Global Navigation Satellite System (GNSS) cannot achieve the required accuracy in urban canyons. On the other hand, simultaneous localization and mapping (SLAM) methods suffer from the error accumulation problem. State-of-the-art localization approaches adopt 3D Light Detection and Ranging (Lidar) to observe the surrounding environment and match the observation with a prior known 3D point cloud map for estimating the position of the vehicle within the map. However, storing the massive point cloud needs immense storage on the vehicle, or it should be stored on servers, which makes the simultaneous downloading of the map by multiple vehicles another challenge. In this study, rather than employing the point cloud directly as the prior map, we focus on the abstract map of buildings, which is easy to extract, and at the same time apparently observable by Lidar. More especially, we proposed vehicle localization methods based on two different abstract map formats representing urban areas. The first format is the multilayer 2D vector map of building footprints, which represents the building boundaries using vectors (lines). The second format is the planar surface map of buildings and ground. These map formats share the same idea that the uncertainty (deviation) of each vector or planar surface is calculated and included in the map. Later in the localization phase, the observed data from Lidar is matched with the abstract map to obtain the precise location of the vehicle. Experiments conducted in a dense urban area of Tokyo show that even though we significantly shrank the map size, we could preserve the mean error of the localization.

© 2017 International Association of Traffic and Safety Sciences. Production and hosting by Elsevier Ltd. This is an open access article under the CC BY-NC-ND license (http://creativecommons.org/licenses/by-nc-nd/4.0/).

1. Introduction

Autonomous vehicle technology has been considered as one of the key components of the intelligent transportation systems since their first introduction. In recent years, significant progress has been made in this field, and many companies already started their field tests. One of the critical requirements of the autonomous driving is the ability to know the ego-position (self-localization) within decimeters [1,2]. Although the Global Navigation Satellite Systems (GNSS) can achieve this level of accuracy in open sky, its positioning quality degrades significantly in dense urban areas due to the signal blockage and multipath effect [3–6]. One of the alternatives or auxiliary technology to GPS is vision. Although vision-based methods have achieved considerable progress [7–11], still this technology suffers from weather conditions, illumination changes, and shadows.

Since many companies are working on getting its price as low as possible, Light Detection and Ranging (Lidar) has once again gotten attention recently as the primary technology for the autonomous vehicle’s perception. Lidar’s measurements are accurate, it has a wide field of view, long range, and it is irrelevant to different light conditions. Many car companies have already started equipping their commercial vehicles with this technology for different driving assistance applications.

Lidar-based localization can be divided into SLAM (simultaneous localization and mapping) and map-based techniques. SLAM methods can be divided into two main categories: feature-based [12] and scan-based [13]. Typically, localization based on SLAM suffers from an error accumulation problem [15]. To overcome this problem, global features should be used to update the location frequently. For example, authors in [14] used building corners to rectify the error accumulation.

There are several environments that GNSS systems cannot meet the demanding requirements for autonomous vehicle’s self-localization. These environments can be built up urban environments or under tree cover including rural areas. In this paper, we focus on the vehicle localization in urban areas using Lidar and priori known map due to the significance of localization accuracy for autonomous driving in complex urban scenarios.
With the development of remote sensing technologies which are being used for airborne laser scanning (ALS) and mobile mapping system (MMS), the large-scale and high accuracy maps have become widely available [2]. The availability of the high-precision maps has increased the interests on the map-based vehicle localization techniques. In most of these methods, the high-definition 3D point cloud is used as the prior map, and in each sequence, the observed Lidar scan is matched to the map to obtain the ego-vehicle position. In [1], the authors use the Velodyne laser scanner in addition to GPS, IMU, and odometer to generate the map offline. Then the localization within the map is performed using the particle filter. The authors extended their work in [16], developing a probabilistic approach. In [17], the authors develop an iterative closest point (ICP) algorithm for local matching of the scans and incorporate a histogram-feature representation for the registration of submaps. In [18], lane markers are used as local features, which are extracted from reflectivity values of the Lidar scans. In [19], a Monte Carlo Localization (MCL) method utilizing the curb-intersection features is introduced, and the road observation is fused with odometry information. Pauly [20] proposed a height map encoding in point cloud by spectrally analyzing the point cloud.

After the ICP algorithm was proposed, Biber [21] introduced the idea of representing a two-dimensional environment by normal distributions for the scan registration. This method is known as Normal Distributions Transform (NDT). In NDT, map space is divided into fixed 2D cells, and for each cell, a corresponding normal distribution is generated. Then, rather than matching the scan data to the point cloud, it is matched to the set of normal distributions. Later, Magnusson et al. extended the NDT idea to a 3D domain and proposed many variants for NDT [22–25]. On the other hand, Tu et al. [26] was inspired by image compression techniques and proposed a method to compress the streaming point cloud using image and video compression methods. The authors also evaluated their compressed point cloud for the self-localization application. In our previous work [27], a multilayer 2D vector map was proposed for the vehicle self-localization which the map consisted of different heights of the building walls in 2D vector format. Therefore, the data size of the map was remarkably smaller than the conventional point cloud format. On the other hand, since the proposed vector map is multilayer and the normal distributions are generated from the vectors, we could achieve a better localization accuracy comparing to the conventional methods based on 2D point cloud maps.

City scale 3D point cloud data is too bulky to be stored in internal storages of the autonomous vehicles since each square kilometer consists of around 300 million points. For the self-localization applications, map structure should be small in size and, at the same time, capable of containing enough features for the localization. In an urban area, there are many structured objects such as buildings. These structures are made of many basic elements, such as 2D lines and 3D planes which are easy to extract and, at the same time, clearly observable by Lidar. In this study, rather than using the massive 3D point cloud as the prior map, we employ two abstract map formats for the 2D and 3D localization of the autonomous vehicle: multilayer 2D vector map and planar surface map. These maps are generated by extracting lines (vectors) and planes (planar surfaces) in addition to their uncertainties from the 3D point cloud. Therefore, we can abstract several thousands of points by a single element, and as a result, we can significantly shrink the map size (25 million points to around 1000 elements). In the localization phase, the observed data form Lidar is registered into the abstracted features to obtain the location. Fig. 1 shows the concept of localization using the proposed abstract map formats. The experimental result proves the effectiveness of the proposed conception of the abstract map-based localization.

The rest of this paper is organized as follows. Section 2 introduces the proposed localization method based on multilayer 2D vector map of buildings. Section 3 describes the planar surface map-based localization. In Section 4, we evaluate the proposed vehicle localization methods, and conclude the paper in Section 5.

2. Vehicle self-localization based on multilayer 2D vector map

2.1. Generating multilayer vector map

Buildings are the most available and stable features in cities comparing to trees, traffic signs, and poles. If a map is made based on buildings, updating and maintaining the map will be less challenging. On the other hand, Lidar can observe building surfaces clearly. Even though the lower parts of the buildings can be partially obscured by trees, cars or other...
dynamic objects but the upper layers of the building can be apparently captured by Lidar. Therefore, we only consider the buildings in the proposed map structures.

The 2D footprint of a building is generally considered as the area on a surface covered by the building at the ground level. However, as shown in Fig. 2, a building can have different footprints in different heights. In this case, if the 2D map is generated based on only a specific height of the buildings (i.e., footprints), self-localization of the vehicle might face a problem since the laser scanner installed on the vehicle might see a different height of the buildings which have different surface comparing to the map. In this section, a vehicle self-localization method based on the multilayer 2D vector map is proposed to reduce the map size while maintaining the localization accuracy. In the proposed structure, instead of storing and employing massive 3D or 2D point clouds as the prior map for the two-dimensional localization, 2D lines extracted from the 3D point cloud of the buildings are stored in the map as vectors. Thus, for each building, the footprints of the building at different height-levels are stored in the map, and as a result, a better localization can be expected. The term “multilayer” does not mean that a certain number of layers are considered in the map generation. In fact, it means that each building in the 2D vector map is generated from multiple footprints, if available (Fig. 2).

To generate the multilayer vector map, the entire building points are extracted from the calibrated MMS point cloud (Fig. 3(a)) by referring to ALS, firstly. To do this, ground regions are defined by extracting the ground points in ALS using the method presented in [29] and extending the points to 2.5D. Then, MMS points which are in non-ground regions are considered as building points (Fig. 3(b)). Secondly, the building points are projected onto the ground plane to form a multilayer 2D point cloud which contains the building footprints of different heights. From this 2D point cloud, vector segments (lines) are extracted using the Random Sample Consensus (RANSAC). During the vector extraction, the uncertainty of each vector is also calculated and stored. Assuming the length of an extracted vector \( \mathbf{v} \) which represents a building wall is \( l \), a 2D bounding box with the side lengths of \( l \) and \( l/10 \) is defined around the vector, as shown in Fig. 4. Given the points within this bounding box \( \{ \mathbf{y}_1, ..., \mathbf{y}_n \} \), the uncertainty of the vector \( \sigma \) is calculated as

\[
\sigma = \frac{1}{n} \sum_{k=1}^{n} \text{Distance}(\mathbf{v}, \mathbf{y}_k).
\]  
(1)

For each vector, the head and tail, and the uncertainty values are stored in the map. Fig. 3(c) shows the multilayer 2D vector map generated for the selected area, and Fig. 3(d) visualizes the vector uncertainties by the blue ellipses.

2.2. Scan-to-vector-map matching for localization

To estimate the current state of the vehicle, the laser scan should be registered to the prior abstract map. The flowchart of the proposed localization method is shown in Fig. 5. Firstly, the vector map is converted to a normal distribution (ND) map, and the laser scan is preprocessed before the matching. In the map matching phase, the scan is matched to the ND map using the point-to-distribution variant of NDT (P2D-NDT).

2.2.1. Generating normal distribution (ND) from 2D vector map

To perform the fast and robust matching between the Lidar scan and 2D vector map, the NDT-matching is used. In NDT-based point cloud registration, unlike the iterative closest point (ICP), the reference point cloud is converted to a set of Gaussian distributions, and then the matching is performed over these distributions [24]. In the conventional NDT, to calculate a set of NDs which represents the map, a process called discretization is performed which divides the map space into grids with a fixed size. Usually, the uncertainties in the reference ND map come from this discretization process which results in a considerable alignment error. Finding the suitable grid size is a difficult task and depends on many parameters such as the environment, sensors, and etc. The larger the grid size becomes, the more uncertainty appears in the NDs. If a small grid size is chosen to limit the uncertainty of the NDs, then a considerable number of the grids will not have enough point for making ND (5 points), and therefore, we will have a sparse map without enough features which is not suitable for the localization. Moreover, in the conventional NDT method, two close walls might fall into a single cell of the grid, and therefore can be abstracted by one ND. To avoid the problems mentioned and maintain the localization accuracy, this paper generates NDs based on the extracted vectors and their uncertainty values.

To generate ND from the 2D vectors in the map, we need to define the mean and covariance for each of them. The probability of observing a point \( p \) in \( \mathbb{R}^2 \) on a particular vector represented by a normal distribution is derived from the following equation:

\[
P(p) = \frac{1}{2\pi \sqrt{\text{det}(\Sigma)}} \exp\left(\frac{- (p - \mu)^T \Sigma^{-1} (p - \mu)}{2}\right)
\]  
(2)

where \( \Sigma \) is the covariance matrix, and \( \mu \) is the mean. In the case of vectors, their center points (Fig. 4) are selected as the mean values. To define the covariance matrices, Eigendecomposition is used. The covariance matrix for the vector \( \mathbf{v} \) can be defined as:

\[
\Sigma_v = \mathbf{U} \Lambda \mathbf{U}^T = \begin{bmatrix}
u_1x & u_2x \\ u_1y & u_2y \end{bmatrix} \begin{bmatrix}
\lambda_1 & 0 \\
0 & \lambda_2 \end{bmatrix} \begin{bmatrix}
u_1x & u_1y \\ u_2x & u_2y \end{bmatrix}.
\]  
(3)
where \( u_1 \) and \( u_2 \) are the Eigenvectors, and \( \lambda_1 \) and \( \lambda_2 \) are the Eigenvalues. The Eigen vectors are obtained from the head and tail points of the vector segments as follows:

\[
\begin{align*}
    u_1 &= \begin{bmatrix} v_{1x} - v_{2x} \quad v_{1y} - v_{2y} \end{bmatrix}, \\
    u_2 &= \begin{bmatrix} 0 & -1 \quad 1 & 0 \end{bmatrix} u_1,
\end{align*}
\]

where \( v_1 \) and \( v_2 \) are the head and tail points of the vector \( \vec{v} \). \( \lambda_1 \) is defined as half of the length of the vector, and \( \lambda_2 \) is set to the uncertainty of the vector \( (\sigma) \) which was calculated in Eq. (1). The NDS generated from the vectors are shown in Fig. 6. As each vector has its own ND, the uncertainties caused by the discretization in the conventional NDT can be resolved, and therefore, building corners and neighboring walls can be represented correctly.

2.2.2. Scan to normal distribution (ND) vector map matching

After obtaining the ND map from the multilayer 2D vector map, we have to localize the vehicle within the map. For the localization, a multi-layer (channel) Lidar mounted on the roof of the vehicle is used. Usually, the distortion of the Lidar scan which is caused by the vehicle motion during the experiment is eliminated using a relative movement acquired from the odometry sensors. In our algorithm, we proposed a method to remove the distortion inside the optimization process of the map matching. In this case, as can be seen in the flowchart of the map matching in Fig. 5, each time the transformation vector transforms the scan, the distortion elimination process is performed to calculate the distortion based on the newly estimated position.

To perform a 2D matching, the multi-layer laser scan should be converted to a 2D point cloud by projecting all layers to the ground plane. Employing a multi-layer laser scanner instead of a single-layer one brings two advantages for the map matching. Firstly, it increases the point density of the building footprints. And secondly, scanning different levels of buildings allows us to capture different footprints of a single building. Having access to a dense scan makes the registration robust against outliers. To avoid bias from uneven point distribution and to reduce the matching time, the input scan should be downsampled [23]. In the proposed method, to keep the benefits of the higher point density around building walls, the downsampling is done before the 2D projection. For the downsampling, we create a 2D grid of 30 cm × 30 cm over the input scan. And then, all points within each grid will be approximated with their centroid.

After preprocessing mentioned above, the 2D scan is transformed from the vehicle coordinate to the map coordinate (global) with an initial guess acquired from two previous subsequent positions. Then, the best alignment is obtained which is the transformation matrix \( M_t \). In fact, \( M_t \) is the optimal 2D transformation matrix that is applied to the initially aligned laser scan to further match the scan to the ND map, and defined as

\[
M_t = \begin{bmatrix} R_z(\theta) & T_{xy} \\ 0 & 1 \end{bmatrix}
\]

where \( R_z(\theta) \) describes \( \theta \) degrees of rotation around the Z-axis, and \( T_{xy} \) is 2D translation. This optimal matrix is obtained by maximum likelihood estimation [27].
Suppose that $X = \{x_1, \ldots, x_n\}$ is the initially transformed input scan in $\mathbb{R}^2$, and $P_j$ is the ND for the vector segment $j$. The optimal $M_t$ is the transformation that maximizes the likelihood function (score function)

$$L = \prod_{k=1}^{n} P_j(M_t x_k)$$

(6)

or, equivalently, minimizes the log-likelihood function of $L$:

$$-\log L = -\sum_{k=1}^{n} \log P_j(M_t x_k)$$

(7)

This cost function is less complex to be used for calculating gradient and Hessian matrix in the optimization process, and thus, Eq. (7) is used for obtaining $M_t$. In Eq. (7), outlier points significantly decreases the matching score. Therefore, instead of the normal distribution, a mixture of uniform and normal distribution $\tilde{P}_j$ is used:

$$\tilde{P}_j = \xi_1 P_j + \xi_2 P_0$$

(8)

where $P_0$ is the expected rate of the outliers, and $\xi_1$ and $\xi_2$ are constants such that Eq. (8) integrates over the $j$th vector to one. By applying Eq. (8) to the log-likelihood function in Eq. (7) and approximating it for the sake of simplicity, the score of $k$th point can be defined as:

$$\text{Score}(M_t, x_k, j) = d_1 \exp \left( -d_2 \frac{(M_t x_k - C_j)^T \Sigma_j^{-1} (M_t x_k - C_j)}{2} \right)$$

(9)

where $j$ is an index of the closest vector to the transformed point $M_t x_k$, $\Sigma_j$ and $C_j$ are covariance and center of the $j$th vector, and $d_1$ and $d_2$ are obtained from $\xi_1$ and $\xi_2$. Finally, the cost function of the transformation matrix $M_t$ for the input scan $X$ is defined as:

$$\text{Cost}(M_t) = -\sum_{k=1}^{n} \text{Score}(M_t, x_k, j)$$

(10)

To find the optimal $M_t$, the Newton optimization method is employed. Newton’s method optimizes the cost function by

$$H \begin{bmatrix} \Delta x \\ \Delta y \\ \Delta \theta \end{bmatrix} = -g,$$

(11)

where $H$ and $g$ are the Hessian matrix and the gradient of the cost function, and $x$, $y$, and $\theta$ are the translation and rotation parameters of the Matrix $M_t$. After the Newton’s process is merged, the optimum $M_t$ is obtained and the ego-position of the vehicle can calculated by multiplying $M_t$ to the initial translation matrix. Fig. 7 shows a few examples of the proposed map matching.

3. Vehicle self-localization based on probabilistic planar surface map

The vector map format proposed in the previous section is suitable for 2D localization of the autonomous vehicle, but it cannot be used for the 3D localization. Planes are considered as an abstract three-dimensional representation of building walls. Thus, we extend the idea

Fig. 5. The flowchart of the vector map-based self-localization. Inputs are the vector map and input point cloud acquired by the laser scanner, and the output is the current position of the vehicle.

Fig. 6. Normal distributions generated from the vectors.

Please cite this article as: E. Javanmardi, et al., Autonomous vehicle self-localization based on abstract map and multi-channel LiDAR in urban area, IATSS Research (2017), https://doi.org/10.1016/j.iatssr.2018.05.001
from Section 2 and propose a 3D planar surface map format for the vehicle localization. The flowchart of the proposed planar surface map-based localization is shown in Fig. 8. As can be seen in the figure, the output of the planar surface map-based localization is a 6D state of the vehicle.

3.1. Generating probabilistic planar surface map

In contrast to the 2D vector map-based localization which was proposed in the previous chapter where the ground points were removed in the vector mapping step, the ground surface plays an essential role in the 3D localization using the planar surface map, especially for defining the pitch and height of the vehicle position. Therefore, instead of removing the ground surfaces, we actively take them into account for the localization. Thus, the primary components representing a planar surface map will be building walls and ground surface.

The first step of creating a planar surface map is to define the ground surface in the 3D point cloud map (Fig. 9(a)). The ground points are obtained using the cloth simulation filter algorithm proposed in [29] which works well for large-scale ground detection. Fig. 9(b) and (c) shows the ground extraction procedure. Typically, the ground surface does not include so many details and also does not change so often. Therefore, a local surface of the ground can be abstracted by a plane without losing information. To generate the ground map, the ground surface is first divided into large grids (10 m × 10m), and then a plane detection algorithm is applied to the points within each cell. For each detected plane, the uncertainty of the points forming the plane is also estimated and stored. Therefore, for each ground surface, the center, width, length, and normal of each plane are stored beside the mean and sigma of the corresponding distribution. The method of calculating ND from each surface is described after the building surface extraction, for both the ground and building surfaces. Fig. 9(d) shows the normal distributions of the ground surface.

After the ground extraction from the MMS point cloud, remaining points are used to extract the building walls, as shown in Fig. 9(e). The objective of the plane extraction is to define the main features that likely to be observed by the laser scanner in the localization phase, not 3D building reconstruction. The planar surfaces are extracted by efficiently fitting plane to the off-ground points using RANSAC [30]. RANSAC randomly generates plane candidates in each iteration by subsampling the point cloud, and estimates the plane with maximal score. The plane is accepted only if the deviation of the points is less than a predefined threshold. We empirically defined the angular threshold for extracting the planes within the experimental area which is three degree. The remaining RANSAC parameters used for the plane extraction are as follows; maximum distance to plane which is set to 1.0 m, the minimum number of points per plane which is set to 200, and the overlooking probability which is set to 0.01. Results of the plane fitting are shown in Fig. 9(f), where each plane represents a building wall or a part of it. For each extracted plane, just like the ground part, the center, width, length, and normal of the plane, are stored with the mean and covariance of the corresponding ND of the plane.

As mentioned before, to use the planar surfaces extracted from both the ground and building surfaces for the self-localization, we have to...
generate NDs which will be used in the scan matching phase. Therefore, the uncertainty of the points forming each plane should be calculated and stored for making the ND for the plane.

Suppose that a set of points \( Y = \{ y_1, y_2, \ldots, y_n \} \) in \( \mathbb{R}^3 \) is forming the planar surface entity \( P_i \). The uncertainty of the plane \( P_i \) is defined by the mean \( \mu \) and covariance matrix \( \Sigma \) as follows:

\[
\mu = \frac{1}{n} \sum_{k=1}^{n} y_k
\]

\[
\Sigma = \frac{1}{n-1} BB^T, \quad B = [y_1 - \mu, \ldots, y_n - \mu].
\]

By having the mean and covariance matrix, the probability of observing a point \( p \) in \( \mathbb{R}^3 \) on a particular wall surface or ground surface represented by a normal distribution is derived from the following equation:

\[
P(p) = \frac{\exp\left(-\frac{1}{2}(p-\mu)^T\Sigma^{-1}(p-\mu)\right)}{\sqrt{2\pi} |\Sigma|^{\frac{1}{2}}}
\]

In the planar surface map, we only store the planes and their uncertainty values for both the ground and building surfaces, and neglect all other details to reduce the map size. While those details might be helpful for the self-positioning, especially in the longitudinal direction, the acquired surface information can provide fairly enough features for the map matching in urban areas. Fig. 10 shows different map formats and abstraction level of a particular urban area. As can be seen, a dense 3D point cloud consists of higher details which lead to massive data size. In the conventional NDT, the grid size and localization accuracy are tightly correlated. While increasing the grid size can reduce the map size, the positioning accuracy is affected and degraded. The proposed 3D surface map can provide the smallest map size and comparable localization accuracy with the original point cloud.

### 3.2. Scan-to-planar-map-matching for localization

#### 3.2.1. Making normal distribution (ND) from planar surface map

In the previous step, the planar surface map of the environment including the ground and building surfaces was generated and stored. In this step, the planar surface map should be converted to an ND map to perform the NDT matching for the localization. If we directly use the uncertainty associated with each surface in the planar surface map as the...
ND (Eq. (14)) and perform the NDT matching, we will face two challenges as written below.

In the NDT matching technique, to calculate the matching score, we should define a corresponding ND for each point in the input scan. This is done by searching and selecting the ND with the nearest centroid to the target point. This method works well if the domain of NDs is relatively small. However, in the planar surface map, typically NDs representing the planes are large. Therefore, distances to the ND centroids are not a good estimate for defining the nearest ND, as shown in Fig. 11(a). In the figure, the closest surface to the point C is P1. However, as the centroid of P2 is closer to C than the centroid of P1, P2 is considered as the corresponding ND for the C. The same problem happens for points A and B, and it causes a significant matching error. To solve this, point to plane distance should be considered to define the nearest plane. However, calculating the distance of all points in the input scan to all planes in the map is time-consuming and makes the matching slow.

The second challenge is score calculation. If each plane is represented by only one ND, the input points with a same distance to the plane might get totally different scores. In Fig. 11(a), points A and D have an almost same distance to the plane P1, but the score of A is much lower than D.

To overcome the problems mentioned above without increasing the size of the planar surface map, each plane is subdivided into a fixed smaller size and then represented by multiple NDs in the localization phase, as shown in Fig. 11(b). By doing this, the nearest ND to the input points can be estimated by searching the centroids of NDs by k-d-tree nearest neighbor algorithm which is fast, and also the score function will become more uniform. Since this procedure is done in the localization phase, the stored map size is not increased.

3.2.2. Scan to normal distribution (ND) plane map matching

Before the map matching, the input scan is preprocessed to remove the distortion caused by vehicle motion using the method described in Section 2.2.2. After excluding the distortion, scans are downsampled using a similar method described in Section 2.2.2. Then, the downsampled scan is transformed from the vehicle coordinate to the map coordinate by an initial guess. The initial guess of the first frame is obtained from GPS, but after that, it is estimated by a simple prediction performed based on two previous positions of the vehicle.

Fig. 11. Challenges in defining the corresponding ND for each point in the input scan: (a) misdetection of the nearest ND due to the large ND domain, and the score reduction due to the distance of the input points from the center of the ND; (b) the positive effect of representing the planar surfaces using smaller NDs.

Fig. 12. The experimental area in the vicinity of Hitotsubashi, a typical urban area in the Chiyoda-ku area of central Tokyo, Japan. The red line shows the experimental route and the green dots show the position acquired by GPS for the route.
Fig. 13. The experimental setup: (left) the MMS used for the 3D mapping; (right) the experimental vehicle for the localization. Velodyne VLP-16 is mounted on the top of the car at the height of 2.3 m.

Fig. 14. Evaluation of the proposed vector map format: (a) number of the map elements in different map formats; (b) comparison of the localization accuracy using the vector map and single layer 2D point cloud map with different heights.
Suppose that the transformation vector for the initial guess is \( \bar{p} = [t_x, t_y, t_z, \theta_x, \theta_y, \theta_z]^T \) where \( t_x, t_y, t_z \) are the translation, and \( \theta_x, \theta_y, \theta_z \) are the rotation parameters. Assuming that \( X = [x_1 \ldots x_m] \) is the downsampled laser scan and \( \hat{X} \) is the downsampled scan after applying the transformation \( \bar{p} \), goal of the matching is optimizing the transformation \( \bar{p} \) to maximize the matching score. Therefore, we can assume that accuracy of the transformation is increasing during the optimization process. If the vehicle motion is known, distortion of the sensor reading can be easily corrected. Thus, in each iteration, for the estimated transformation \( \bar{p} \), the motion of the vehicle is estimated and the distortion of the input scan is reduced by improving the accuracy of estimated motion.

Assuming that \( T \) is the transformation function and \( D \) is the distortion elimination function, \( \hat{X} \) can be defined as

\[
\hat{X} = T(\bar{p}, X), \quad \hat{X} = D(X) = D(T(\bar{p}, X)).
\]

(15)

Similar to Section 2.2.2, the cost function of \( \hat{X} \) is defined as

\[
\text{Cost}(\hat{X}) = -\sum_{k=1}^n \text{Score}(\hat{x}_k).
\]

(16)

Therefore,

\[
\text{Cost}(\bar{p}) = -\sum_{k=1}^n \text{Score}(D(T(\bar{p}, X))).
\]

(17)

To find the optimal \( \bar{p} \) which is the best matching result, the Newton optimization method is applied. Newton’s method optimizes the cost function by iteratively changing the \( \bar{p} \) using the following equation:

\[
H_{ij}\Delta \theta_i \Delta \theta_j = -g_{ij}
\]

(18)

where \( H \) and \( g \) are the Hessain matrix and the gradient of the cost function, \( \Delta \theta_i \) and \( \Delta \theta_j \) are the translation, and \( \theta_i, \theta_j \) are the rotation parameters of \( \bar{p} \). After the Newton’s process is converged, the ego-position of the vehicle can be calculated using the optimum \( \bar{p} \). For the details of the optimization process, the reader is referred to [28].

4. Experimental Results

This section presents the experimental results of the proposed map formats and localization methods. To evaluate the performance of the localization methods using the proposed map formats, the experiments were conducted in the vicinity of Hitotsubashi, a typical urban area in the Chiyoda-ku area of central Tokyo, Japan. Fig. 12 shows the route of the experiment with the length of 650 m and accuracy of the GPS-based localization in the area. For the experiment, we drove the vehicle four times on the same route and compared the 2D and 3D localization accuracy using different map formats.

Fig. 13 (left) shows the MMS used for the mapping and Fig. 13 (right) shows the experimental vehicle and sensor setup for the localization. The 3D point cloud map is obtained by two single layer SICK laser scanners, and offline calibration using our previous work [2,31]. For the localization, our experimental vehicle is equipped with Velodyne VLP-16 laser scanner which has 100 m range, 360-degree horizontal and 30-degree vertical field of view. The laser scanner is installed at the height of 2.3 m and set to spin at 10 Hz which collects ten scans in each second. The maximum speed of the vehicle in the experiments was 47 km/h. Since defining the ground truth in this route is challenging, we adopted the 3D point cloud-based localization method as a quasi-ground truth.

To evaluate the proposed map structures, three parameters were evaluated: 1) map size, 2) localization accuracy, and 3) processing time.

4.1. Evaluation for multi-layer vector map-based localization

Fig. 14(a) shows the number of elements in each map structure representing the map size. As shown in the figure, using the vector map can dramatically reduce the map size comparing to the point cloud and other formats. Fig. 14(b) compares the self-localization accuracy (mean, max, and variance) of the proposed method and single layer 2D point cloud maps with different heights. Although the map size is reduced, the average error of the vector map is less than 20 cm,
which proves that the proposed format could preserve the information required for the localization. All methods in Fig. 14(b) use the same multilayer laser scan as input with the same downsampling procedure. To perform the localization within the single layer 2D maps, 1.0 m NDT grid is employed.

In Fig. 15, 2D point cloud map is made from all heights to eliminate the effect of different layers in localization so that we can compare vector-based discretization and static grid-based discretization. Our method still has better performance. Fig. 16 shows a comparison of the proposed method with maps with different grid sizes. As shown in Fig. 16, the vector map outperforms the 2D point cloud based method with different grid size as well. Fig. 17 shows the longitudinal and lateral localization error during the route. The average processing time of the localization for the vector map was 44.66 ms while the frequency of input scan was 10 Hz. Moreover, more than 16 short and long vectors were employed in average during the localization using the vector map. This number tightly depends on the mapping resolution and parameters and varies in different places of the map.

4.2. Evaluation of planar surface map-based localization

Fig. 18 shows the map size of the experimental area for the planar surface map and other map formats. While the total number of points in the original point cloud data was around 25 million points, grid-based representation of the map could reduce the map size to 1043 elements which is much smaller than the original point cloud (25,000 times) and grid-based maps (160 times). Fig. 19 compares the accuracy of the localization using the planar surface map and other formats. As the figure shows, by increasing the grid size in the conventional map formats, the accuracy of the localization decreases, since the grid-based abstraction eliminates useful details of the map for the localization. The proposed method has lower error comparing to 2.0 NDT-based localization while its map size is around 47 times smaller. In NDT-based localization methods, the optimum size of the grid depends on the environment and there is no solid method for finding the optimum grid size in the literature. The mean and maximum localization error of the planar surface map is 43 cm and 120 cm, and the average processing time is 31.41 ms. These results demonstrate that, though we used an extremely smaller map structure, we could preserve the localization accuracy. As can be seen, in the proposed method, the smallest error is for the altitude. This is because the ground does not have so many details and can be abstracted without significant information loss. In addition, the distortion in the z-axis is negligible since car’s altitude does not change significantly during a scan. Furthermore, the results show that the longitudinal error is more than lateral error because the buildings on two side of the street are providing plenty of lateral features. Since in some part of the experimental route, neighboring buildings are almost connected, the walls which are perpendicular to the moving direction of the vehicle cannot be fully captured by MMS in the mapping phase. As a result, points on those walls become very few, and those parts of the buildings cannot be extracted in the planar surface map. Therefore, lack of those features results in a longitudinal error in positioning. This problem can be solved by adjusting the plane extraction parameters for such areas. One possible
future work is to find out such areas and add more details to the map to improve the longitudinal accuracy.

Fig. 20 shows the lateral and longitudinal error over the experimental route using the planar surface map.

Finally, the average number of the planes employed during the localization using the planar surface map was more than 34, while, this number was much higher in intersections. In contrast, in areas with long and flat buildings the number of features was lower.

5. Conclusions and future work

In this paper, we have proposed new map formats and corresponding localization methods based on multi-layer Lidar. The proposed abstract map is extremely small in size comparing to the 3D point cloud maps while providing enough features for accurate self-localization. For the same area, point cloud map had around 25 million points where the abstract map could reduce the number of map elements to around 1000 entities. To evaluate the capability of the abstract map and the corresponding localization methods, experiments have been conducted in a dense urban area of Tokyo with Velodyne VLP-16. The multilayer 2D vector map based localization has achieved a mean 2D error of 20 cm. Moreover, the planar surface map-based localization achieved about 43 cm of error, while the vehicle state could be estimated in 3D space. The paper proved that the abstract map could be alternative to the redundant point cloud map for the vehicle self-localization application. In the future, we will focus on the localization by integrating the multilayer 2D vector map and the planar surface map and extend the proposed framework to other scenarios such as rural environments.

References


