Research Article

Cloud Update of Geodetic Normal Distribution Map Based on Crowd-Sourcing Detection against Road Environment Changes

Chansoo Kim, Sungjin Cho, Myoungho Sunwoo, Paulo Resende, Benazouz Bradaï, and Kichun Jo

1Department of Intelligent Mobility, Chonnam University, Gwangju 61186, Republic of Korea
2Smart Mobility R&D Center, ACELAB, Seoul 06222, Republic of Korea
3Department of Automotive Convergence, Korea University, Seoul 02841, Republic of Korea
4Driving Assistance Research, Valeo, Créteil 94000, France
5Department of Smart Vehicle Engineering, Konkuk University, Seoul 05029, Republic of Korea

Correspondence should be addressed to Kichun Jo; kichun@konkuk.ac.kr

Received 1 January 2022; Accepted 30 March 2022; Published 25 April 2022

Copyright © 2022 Chansoo Kim et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

LiDAR-based localization has been widely used for the pose estimation of autonomous vehicles. Since the localization requires a sustainable map reflecting environment changes, a map update framework based on crowd-sourcing measurements has been researched. Unfortunately, a point cloud map occupies too large data size to transmit data in the uploading and downloading of the map update framework. To realize the LiDAR map update framework by reducing the data size, we proposed a novel map update framework using a Geodetic Normal Distribution (GND) map that compresses the point cloud to the normal distributions. The proposed GND map update framework comprises two parts: map change detection based on crowd-sourcing vehicles and map updating based on a map cloud server. GND map changes are detected based on an evidence theory considering geometric relationships between the GND map and crowd-sourcing measurements and uploaded to the map cloud server. Uploaded map changes reproduce representative map changes based on a similarity-based clustering, which are updated into the GND map. The proposed framework was evaluated in simulations and real environments on construction sites. As a result, although partial map changes occurred, the GND map was kept up-to-date through the proposed framework and the localization for autonomous driving was performed successfully.

1. Introduction

Recently, map-matching localization has been globally researched for localization of autonomous vehicles [1–3]. After a high-definition (HD) map has been constructed by a mapping vehicle equipped with a mobile mapping system (MMS), autonomous vehicles with affordable sensors can estimate their poses (position and heading) by matching their measurements with HD map information. One of the widely used sensors for precise positioning based on map-matching is an LiDAR sensor [4–6]. Localization can be achieved by matching geometric shapes measured by the LiDAR sensor with a LiDAR map which is constructed by the mapping vehicle. Since the LiDAR sensor provides a three-dimensional precise point cloud within 10 cm accuracy and the LiDAR map provides precise surrounding information within 20 cm accuracy, a localization accuracy within 30 cm can be achieved [7].

Although LiDAR sensors can offer very accurate localization performance, they do encounter a critical problem. The problem is an adaptation to changes in real environments. High-accuracy LiDAR-based localization can be achieved in environments with no changes in their geometric shapes. However, it is easy for the geometric shapes in the environments to be changed. For example, vegetation, such as trees, plants, and grass, can be grown; semistatic objects, such as parked vehicles, can be moved; and construction sites can appear. Accordingly, for LiDAR-based
localization, it is essential for environment changes to be periodically updated into the existing HD map for minimization of the differences between the environments and the map.

However, change updates based on conventional mapping methods using a mapping vehicle with precise sensors have several problems: (1) Mapping vehicles must be re-driven on all roads to acquire new data, which has a substantial cost. (2) Latency necessarily occurs in map updates because it takes lots of time for a few mapping vehicles to acquire all necessary data. Therefore, a novel approach to update road environment changes is required.

To solve the problems for the adaptation to environment changes, numerous companies (HERE [8], TomTom [9], Mobileye [10], Bosch [11], and Daimler [12]) and academies ([13–15]) have researched a map update framework based on crowd-sourced features (i.e., lanes and traffic control devices), which are measured from camera sensors of intelligent vehicles driven on roads. After crowd-sourced features are uploaded into a map cloud server, the features are merged into the representative changes, and the representative changes are updated into the existing HD map. As a result, the updated HD map information is downloaded and used in intelligent vehicles. The sequential map update framework has tried to overcome the inaccuracy of the features measured by camera sensors through the merging of crowd-sourced information. As a result, the framework based on crowd-sourcing vehicles on roads can provide some advantages such as reducing costs and rare latency for map updates.

Unfortunately, while concepts of map update frameworks based on visual features such as lanes and traffic control devices have been open to the public, most frameworks have not clearly provided performances and results for the HD map and vehicle pose estimation based on the cameras yet. In addition, the camera-based map update frameworks cannot be performed either in the alleys or in the large intersections without lanes and traffic control devices.

Different from the camera-based localization and map update framework, the LiDAR-based localization and map update framework has two advantages. The LiDAR-based approach provides good positioning performances and map accuracy due to the accuracy of the LiDAR sensors. Next, the approach enables vehicle positioning wherever regardless of the absence of visual features such as lanes and traffic control devices. However, it is difficult for the framework to be directly applied for an update of the LiDAR map because map-relevant LiDAR data are too large to be uploaded into the map cloud server and downloaded to intelligent vehicles. While map-relevant information based on camera sensors occupies 10 kB/km [10], raw point cloud map data occupies over 500 MB/km on real roads [7]. Accordingly, a LiDAR map structure for the map update framework based on crowd-sourced measurements is required to occupy a low data size. In addition, researches such as change detection and merging of crowd-sourced measurements are required for the map update framework to adopt the map structure.

To compress the map-relevant LiDAR data between the crowd-sourcing vehicles and the map cloud server, a concept of a Normal Distribution Transform (NDT) map is used in the paper. Especially, a Geodetic Normal Distribution (GND) map structure [16], which is extended from the NDT map in our previous research, is adopted because it supports a unified map structure for multiple vehicles. Based on the worldwide management property of the GND map, the map data are interpreted in the same manner in individual vehicles regardless of the coordinate conversion errors. In order to update the GND map periodically, the paper proposes a cloud update framework of the GND map based on crowd-sourcing detection of road environment changes. There are two problems for the GND map update framework to be considered for realization in the real roads. First, map changes must be detected in crowd-sourcing vehicles. In order to detect map changes, each change probability in each cell of the GND map is estimated based on the evidence theory and the ray-casting approach between the measured points and the normal distributions. Second, it is essential to merge and reproduce representative changes from crowd-sourced map changes, and then the representative changes are updated into the existing GND map. In order to merge the crowd-sourced map changes, a similarity-based clustering algorithm is applied.

The objective of this paper is to keep the GND map up-to-date for continuous localization of autonomous driving. To achieve the objective, the contributions of the paper are the following:

(i) The proposed framework reduces the map-relevant data size by applying the GND map structure in the overall map update process

(ii) The change detection algorithm based on the evidence theory and the ray-casting approach between the measured points and the normal distributions detects the GND map changes from crowd-sourcing vehicles

(iii) After the GND map changes detected by crowd-sourcing vehicles are merged into representative map changes based on the clustering algorithm using a similarity of map changes, the representative map changes update the existing GND map in the map server

To explain the framework of the GND map update, this paper is organized as follows. Section 2 explains some works related to map update system based on the LiDAR sensor. Section 3 describes an overall framework to update the existing GND map. In Section 4, crowd-sourcing vehicles can detect changes from GND map structure. Section 5 explains extraction of a representative change from the crowd-sourcing map changes in the map cloud server. In Section 6, the performance of the proposed framework is evaluated in simulation environments. In Section 7, experiments are performed for analyzing the performance of the proposed framework in real environments. Finally, the paper is concluded with Section 8.
2. Related Works

In an autonomous driving field, there are mainly three LiDAR map structures: point cloud (PCD) map [17–20], grid map [21–24], and geometric map [25–28]. Since the PCD map is constructed easily by accumulating point clouds acquired by LiDAR sensors based on the vehicle moving trajectory, it is widely applied as a fundamental map type for LiDAR. The grid map is also widely used as the map structure for LiDAR. In particular, since the occupancy grid map consists of lots of grids with occupancy probabilities distinguishing based on the probability theory and the LiDAR ray-casting, the map type can remove the moving objects in the mapping process easily. Finally, the geometric map structure, such as the NDT map [25, 26] and Gaussian mixture map [27, 28], can be used for the LiDAR map. To reduce the map size, after points in the PCD map are split into lots of voxels similar to the grid map, the points are converted to the normal distribution or the Gaussian mixture distribution.

Map update approaches can be generally determined based on the properties of the map structure. As shown in Table 1, the approaches to updating map changes can be split into two categories: standalone-based and crowd-sourcing-based. In researches on standalone-based map update, there are two main parts: map change detection/update and instant map update. The PCD map structure can be generally updated after changes have been detected because the PCD map structure has definite point information around environments. The first approach to map change detection for the PCD map involves determining the minimum distances between sensor points and map points [29, 30]. The other approach determines changes by checking whether map points are traversed by sensor points [31–34]. Among them, Xiao et al. applied the Dempster–Shafer theory to integrate multiple inferences from multiple rays to estimate the change states of map points precisely [33, 34].

An occupancy grid map can be updated based on the instant map update strategy because the map structure models the occupied probabilities in spaces that do not contain definite information. In addition, the occupancy grid map structure supports a basic function to instantly update the occupied and free probabilities of cells based on the ray-casting of LiDAR measurements. In this way, the occupancy grid map can be easily updated in a map-changing environment [35]. Particularly noteworthy is the frequency map enhancement platform proposed by Krajnik et al., which provides and updates the occupancy grid map frequently [36, 37]. However, the occupancy grid map cannot distinguish the unknown grids (no measurement region) and conflict grids (different measurements in the same region) explicitly. In order to solve the problem, Trehard et al. applied the evidence theory to the occupancy grid map [38, 39].

A NDT map has been researched for both the change detection/update strategy and instant map update strategy. Katsura et al. detected changes in the NDT map by comparing the normal distributions in the NDT map with the normal distributions constructed by measurements [40]. To determine the differences of distributions from two sources, they compared geometric shapes consisting of sheets, planes, and lines determined by eigendecomposition. However, this approach cannot distinguish deleted cells and unmeasured cells because the changes cannot distinguish free space based on ray-casting. On the other hand, an NDT map integrated with an occupancy grid map can be updated using the instant map update strategy [41–43]. This approach instantly updates the occupied probability in the cells of the NDT map based on ray-casting. Simultaneously, the normal distribution in the cell is updated based on the recursive covariance sample update. However, in this approach, the cell information is not definite within the transient state, which is different from the deterministic state in a real environment, where one state must be allocated. In addition, because all measurements are required to update the map, they may occupy too large size.

On the other hand, a crowd-sourced map update system for LiDAR has been widely researched. The approach used crowd-sourced sensor information transmitted to a cloud server. To update the PCD map structure, Kim et al. applied both probabilistic and evidential theories to the map update based on the ray-casting approach [44]. In addition, Xue et al. used the infrastructure units for point cloud map update [45]. However, the PCD map structure still occupies a too large size to upload and download the map data. For the probabilistic occupancy grid map structure, there are several ways to update the map based on crowd-sourced data [46–48]. However, the occupancy grid map cannot explicitly deal with unknown states that are not measured from the sensors. To overcome this problem, Jo et al. proposed updating a worldwide 3D environment based on an evidential occupancy grid map using multiple vehicles [49]. However, to the best of our knowledge, no crowd-sourced data-based map update systems currently exist for other geometric map structures.

3. Framework of GND Map Update Based on Crowd-Sourcing Detection

Figure 1 illustrates an overall framework of the GND map update system based on crowd-sourced data in changing environments. The system has two physical parts: a collection of intelligent vehicles and a map cloud server. Intelligent vehicles detect differences between a GND map and point clouds measured by in-vehicle LiDAR sensors, which are denoted as map changes. The map changes detected by multiple vehicles are uploaded to the map cloud server.

| Agent          | Map type | Methodology                  | Research |
|----------------|----------|------------------------------|----------|
| Standalone     | PCD      | Change detection/update      | [29–34]  |
| Grid           |          | Instant map update           | [35–37]  |
| NDT            | Grid     | Change detection/update      | [40]     |
| Crowd-sourcing | Grid     | Map merging                  | [46–49]  |
server merges the crowd-sourced map changes into representative map changes, which update the existing GND map, denoted as the base GND map. The process of the crowd-sourced map update framework has five steps, which are conducted as follows:

(i) **Download.** The GND map, which stores the geometric shape information about road environments as multiple normal distribution models, is used as the LiDAR map for the localization of intelligent vehicles. Because the vehicle approximately knows its location from its GNSS, the GND map information around the location of the ego-vehicle can be downloaded to the vehicle in advance before the vehicle enters into the region through vehicle wireless networks. By the characteristics of the GND map structure compressing the map data, the burden on the wireless network is reduced over the PCD map widely used.

(ii) **Robust Localization.** The pose of the vehicle is essential for detecting the map changes between the GND map and the measured point cloud. The vehicle’s pose can be estimated based on map-matching localization using the geometric matching relationship between measurements and the base GND map. However, changes in the environments may cause poor performance of geometric matching for localization. To reduce the degradation of localization, this paper applies an off-line hierarchical approach based on a submap concept into the GND map-matching process [44]. Because the map update framework does not require the vehicle pose in real-time but the precise vehicle pose, the off-line approach is used for the elaborate vehicle localization.

(iii) **Change Detection.** Based on the pose estimated from robust localization, point clouds measured by in-vehicle LiDARs are used to detect changes within the downloaded GND map. To detect these changes, this paper considers the geometric relationship between LiDAR beam characteristics and normal distributions in the GND map. To estimate the normal distribution changes precisely, an evidence theory is employed. The process of map change detection is explained in Section.

(iv) **Upload.** The map changes detected by multiple intelligent vehicles are uploaded to the map cloud server by vehicle wireless networks. To reduce network costs by minimization of the transmitted data, only changed parts are uploaded. In addition, because the changed parts are formatted by the GND map structure, the data size can be reduced more than the PCD map changes [44].

(v) **Update.** The map changes uploaded by multiple intelligent vehicles merge representative map changes, which update the base GND map in the map cloud server. Because the map changes detected by the vehicles may have some errors due to their low-cost sensors and inaccurate positioning (tens of centimeter level), rules for merging the crowd-sourced data are required. This paper proposes similarity check methods and merging rules between crowd-sourced data based on linear algebra; this is explained in Section.

To implement the proposed GND map update system based on crowd-sourced data, the initial GND map must be constructed by a mapping vehicle equipped with an MMS in advance. The inaccurate crowd-sourcing data measured by affordable sensors cannot be used for the precise mapping but for the map update framework. In addition, dynamic points reflected against dynamic objects (including moving pedestrians, bikes, and vehicles) are classified. In order to classify the moving states of the points, a LiDAR point

---

*Figure 1: Overall framework of map update system based on crowd-sourced data.*
motion segmentation algorithm based on a combination of probabilistic and evidential approaches is used [50]. In addition, it is assumed that real-time updates to the GND map are not necessary; the map update is performed periodically (e.g., daily). It is also assumed that Partial map changes in the interval between periodic map updates can be supported by robust localization. Based on preliminaries and assumptions, the paper mainly focuses on steps of Change detection and Update processes in section and, respectively.

4. GND Map Change Detection in Individual Vehicles

The major objective of this paper is to keep the base GND map up-to-date based on crowd-sourced data from multiple intelligent vehicles. The map change update algorithm has two parts: map change detection in individual vehicles and map change updating based on crowd-sourced data in the cloud. First of all, differences between the base GND map and the real environment, which are defined as map changes, are detected from the intelligent vehicles to keep the base GND map up-to-date. In this section, the change detection algorithm for the GND map structure based on ray-casting is presented.

4.1. Voxel-to-Voxel (V2V) Comparison-Based Map Change Detection

As shown in Figure 2(a), a GND map structure, $m$, is composed of multiple GND tiles $\{m^q\}_{1 \leq q \leq Q}$ split by the same angle in geodetic coordinates, where $Q$ represents the maximum tile id. As shown in Figure 2(b), the tile $m^q$ is the union of GND voxels $\{v^q\}_{1 \leq i \leq I}$, where $I$ is the maximum voxel index within the tile $m^q$. The GND voxel $v^q$ includes the number of points $n^q$, and the normal distribution $N(p^q, \Sigma^q)$ with mean $\mu^q$ and covariance $\Sigma^q$. Because each GND voxel $v^q$ is split by each spatial boundary, the probability of a point $p^q$ being located in the voxel $v^q$ is only related to $N(p^q, \Sigma^q)$. Accordingly, all voxels in the GND map are independent of each other. Due to the independence of each voxel, the change of the GND map can be determined within each voxel. Accordingly, in this section, the GND voxel $v^q$ is simply denoted as $v_{\text{map}}$ with $n_{\text{map}}$, $\mu_{\text{map}}$, and $\Sigma_{\text{map}}$.

As shown in Figure 3, the map change class $c_{\text{map} \rightarrow \text{new}}$ can be defined by five classes: normal, empty, new, modified, and deleted by comparison between GND map voxel $v_{\text{map}}$ and real environment voxel $v_{\text{real}}$. The blue normal class indicates that normal distributions of the voxels in the GND map and real environment are the same. The orange empty class indicates that the voxel does not have any confident normal distributions in both the map and real environment. The normal and empty classes are not map changes between the GND map and real environment. On the other hand, the red-colored new class indicates that the voxel has a new normal distribution that is not in the map. The orange-colored modified class indicates that the voxel has a normal distribution with a different shape to that of the normal distribution in the GND map. The green deleted class indicates that the normal distribution in the GND map is deleted in the real environment. The new, modified, and deleted classes can be determined as map changes from the GND map.

The voxel-to-voxel (V2V) comparison-based map change detection may detect the map change classes $c_{\text{map} \rightarrow i}$ by comparing voxels of the GND map with voxels of the real environment constructed by point clouds of an intelligent vehicle $i$ [40]. Unfortunately, in contrast to the definition of the map change class, the algorithm encounters some problems in the real road environment because it does not consider the measurement limitations. In a moving and changing environment with moving and parked vehicles, point clouds measured by an intelligent vehicle can be occluded by other objects, as shown in Figure 4(c). The occlusion of measurements in Figure 4 can cause two problems: (1) incomplete normal distributions and (2) ambiguity in unknown regions. First, even with no changes in the real environment, the occluded measurements can construct incomplete normal distributions, as shown in (1) of Figure 4. The incomplete normal distributions distinguish voxels as being modified incorrectly. In addition, unknown voxels (not measured by LiDAR in (2) of Figure 4) and deleted voxels (passed by the rays in (3) of Figure 4) cannot be distinguished from each other in the algorithm because both types of voxels have no points measured by LiDAR. This problem causes the deleted class to be assigned incorrectly, as shown in Figure 4(b).

4.2. Map Change Detection Algorithm Based on Ray-Casting

To solve two problems with V2V comparison-based map change detection, a novel change detection algorithm based on ray-casting is proposed in this paper. Different from the previous map change algorithm comparing with the constructed map, the new change detection algorithm estimates the map change state $s_{\text{map} \rightarrow i}$ using vehicle $i$’s individual point $p_i$.

The change state $s_{\text{map} \rightarrow i,j}$ can be inferred by a geometric relationship between the voxel in the GND map $v_{\text{map}}$ and the individual point $p_i$ measured by vehicle $i$. When the point is measured in the voxel by being blocked by a normal distribution, the normal distribution may be estimated as sustained, as shown in voxels (1) of Figure 5. Next, when the ray is traversed inside the normal distribution in the voxel, the geometric relationship can provide an inference for the voxel to be changed, as shown in voxels (2) of Figure 5. In addition, as the point is measured in the empty voxel, the voxel can be estimated as changed, as shown in voxels (3) of Figure 5. Accordingly, the geometric relationship between the voxel and the point can provide inference to determine whether the state is changed or sustained.

As shown in Figure 6, there are six cases for the geometric relationship between a point and a voxel, which are distinguished based on three criteria: whether there is a normal distribution or not in the voxel, whether a point is blocked or passed in the voxel, and whether the ray or point intersects the normal distribution or not. The first criterion can be checked by finding the normal distribution in the voxel. The second criterion, which is about whether the point...
is blocked or passed in the voxel, is straightforwardly determined by localizing the point in the voxel and ray-casting towards that point. In the blocked case, the point \( p_j \) is stored in the temporary point storage \( P \) with the same size as the map. The final criterion related to the intersection between a point and a normal distribution is explained as follows.

**Figure 2:** GND map structure in (a) geodetic tiles and (b) Cartesian voxels at the sea-level floor (1209 voxels).

**Figure 3:** Definition of five map change classes including \textit{normal, empty, new, deleted, and modified}.

**Figure 4:** V2V comparison-based map change detection compared with map constructed by intelligent vehicles.
4.2.1. Intersection between Point and Normal Distribution. To evaluate the intersection between a point and a normal distribution, we can check whether the measured point $p_j$ is located inside or outside the normal distribution $N(\mu_{\text{map}}, \Sigma_{\text{map}})$. Unfortunately, the normal distribution does not have a boundary because it is a three-dimensional probability density function, as shown in Figure 7. To determine whether the point $p_j$ is located inside or outside the normal distribution, a confidence interval is employed. The confidence interval $\eta_{\text{bound}}$ to determine the boundary is the physically expected probability that a sample in the voxel is located within the normal distribution. When the confidence interval $\eta_p$ computed by the point $p_j$ is smaller than $\eta_{\text{bound}}$, the point $p_j$ is located within the boundary of the normal distribution determined by the confidence interval $\eta_p$, as described by

$$ \begin{cases} 
\eta_{\text{bound}} \geq \eta_p &: \text{point is located inside distribution,} \\
\eta_{\text{bound}} < \eta_p &: \text{point is located outside distribution.} 
\end{cases} \quad (1) $$

To compute the confidence interval $\eta_p$ at the point $p_j$, the cumulative distribution function of the normal distribution is defined as the probability that a sample lies inside the ellipsoid determined by its Mahalanobis distance $r$ from the normal distribution in (2). Accordingly, the problem of finding the confidence interval $\eta_p$ is converted to the problem of finding the cumulative function of $r^2$. Because the square of the Mahalanobis distance $r^2$ is represented as the sum of the squares of three independent normally distributed Gaussian variables, the cumulative function of $r^2$ can be converted to a chi-square cumulative distribution function, as shown in (3).

4.2.2. Intersection between Ray and Normal Distribution. To check whether the ray by the measured point $p_j$ is traversed into the normal distribution $N(\mu_{\text{map}}, \Sigma_{\text{map}})$, a maximum likelihood point $p_{\text{ML}}$ is used. As shown in Figure 8, the maximum likelihood point $p_{\text{ML}}$ indicates a point with the maximum likelihood on the ray against the normal distribution $N(\mu_{\text{map}}, \Sigma_{\text{map}})$. The maximum likelihood point $p_{\text{ML}}$ can be calculated analytically [41]. When the confidence interval $\eta_{\text{ML}}$ computed by the maximum likelihood point is smaller than the boundary confidence interval $\eta_{\text{bound}}$, the ray is confirmed to have passed into the normal distribution, as shown in

$$ r^2 = (p_j - \mu_{\text{map}})^T \Sigma_{\text{map}}^{-1} (p_j - \mu_{\text{map}}), \quad (2) $$

$$ \eta_p = \int_0^{r^2} t^{(k-2)/2} e^{-t/2} \frac{1}{2^{k/2} \Gamma(k/2)} dt, \quad (3) $$

where $\Gamma(\cdot)$ is the Gamma function, and $k$ is set to 3 in accordance with the three-dimensional normal distribution. The confidence interval $\eta_p$ is determined by the chi-square cumulative distribution function, as described by equation (3). As a result, it is determined whether the point $p_j$ is located inside or outside the normal distribution.

4.3. Evidence Modeling for Geometric Relationship. As shown in Figure 6, the six cases from the geometric relationship between the measured point $p_j$ and the voxel $v_{\text{map}}$ can provide inference to determine whether the change state $s_{\text{map}}=i,j$ of the voxel $v_{\text{map}}$ is changed (C) or sustained (S). However, if the voxel $v_{\text{map}}$ is not measured by any points and passed by any rays, the map change state of the voxel cannot be estimated. To handle the unknown state issue explicitly, evidence theory is applied. In evidence theory, the two states form a frame of discernment $\Omega = S, C$. Additional states $\Omega, \phi$ can be managed explicitly by extending the frame of discernment $\Omega$ to the power set $2^\Omega = S, C, \Omega, \phi$, which is the set of all subsets of $\Omega = S, C$. The state $\Omega$ means that the voxel is sustained (S) or changed (C). However, because the state...
cannot be sustained (S) and changed (C) simultaneously, the state Ω represents an unknown state. The state ϕ represents that the voxel is not both sustained (S) and changed (C). However, because this scenario is physically impossible, the state ϕ represents a conflict situation. To quantify the evidence of each element of the power set, a mass function, denoted by m, is applied. Given a point p, measured by intelligent vehicle i, the mass functions of m_{map→i,j}(S) and m_{map→i,j}(C) represent the beliefs of the voxel being sustained (S) and changed (C), respectively. The mass function m_{map→i,j}(Ω) is the union of the beliefs of sustained and changed, and the mass function m_{map→i,j}(ϕ) represents the belief that the voxel is conflicted by different measurements. The sum of all masses in the power set must be one based on its definition in evidence theory. The mass function m_{map→i,j}(state), including m_{map→i,j}(S), m_{map→i,j}(C), m_{map→i,j}(Ω), and m_{map→i,j}(ϕ), can be modeled by the geometric relationship between the voxel v_{map} and point p measured by intelligent vehicle i.

4.3.1. Case 1: Blocking inside Normal Distribution (BI). In the BI case, the measured point in the voxel v_{map} is blocked inside a normal distribution. When the point is judged to be located inside the normal distribution by (1), the map change class can be either normal or modified, as shown in Figure 9. The measurement can provide a hypothesis for the map change state to be converted as sustained (S) or changed (C) for the map change model. The mass m_{map→i,j}(state) in the BI case can be modeled by

\[
\begin{align*}
m_{map→i,j}(ϕ) &= 0, \\
m_{map→i,j}(S) &= λ_{s,block}, \\
m_{map→i,j}(C) &= λ_{c,block}, \\
m_{map→i,j}(Ω) &= 1 - \sum_{A \notin Ω} m_{map→i,j}(A),
\end{align*}
\]

where the parameters $λ_{s,block}$ and $λ_{c,block}$ are configuration parameters. $λ_{s,block}$ represents the probability that the map change state is sustained, and $λ_{c,block}$ represents the probability that the voxel is changed by the blocked point information. Practically, $λ_{s,block}$ is higher than $λ_{c,block}$.

4.3.2. Case 2: Blocking outside Normal Distribution (BO). In the BO case, the point is blocked outside a distribution. If the point is judged not to be located inside the distribution by (1), the situation is determined as the BO case. When the point is located outside the distribution, there is only the case that the map change class is modified, as shown in Figure 9. Accordingly, the measurement can provide a hypothesis on the map change state to be changed. The mass $m_{map→i,j}(state)$ can be modeled by

\[
\begin{align*}
m_{map→i,j}(ϕ) &= 0, \\
m_{map→i,j}(S) &= 0, \\
m_{map→i,j}(C) &= λ_{c,block}, \\
m_{map→i,j}(Ω) &= 1 - \sum_{A \notin Ω} m_{map→i,j}(A),
\end{align*}
\]

where the parameter $λ_{c,block}$ is equal to the equivalent parameter in the BI case.

4.3.3. Case 3: Passing inside Normal Distribution (PI). In the PI case, the point is not located in the voxel and the ray is passing the distribution, as shown in Figure 9. The voxels passed by the ray are detected by the modified ray-casting algorithm [51]. Because Bresenham’s algorithm [51] assumes that the width, length, and height of all voxels are same, the algorithm is modified for variant voxel sizes in the GND map. In detected voxels passed by the ray, the passing of the distribution can be confirmed by (4). If the ray is judged to intersect the distribution, the voxel is treated as the PI case. In the PI case, there are two cases, where the map change class can be modified or deleted, as shown in Figure 9. Because both cases cause the map change state to be changed, the measurement can provide a hypothesis for the voxel to be changed. Therefore, the mass $m_{map→i,j}(state)$ in the PI case can be derived by

\[
\begin{align*}
m_{map→i,j}(ϕ) &= 0, \\
m_{map→i,j}(S) &= 0, \\
m_{map→i,j}(C) &= λ_{c,pass}, \\
m_{map→i,j}(Ω) &= 1 - \sum_{A \notin Ω} m_{map→i,j}(A),
\end{align*}
\]

where parameter $λ_{c,pass}$ represents a probability for a passed voxel to be changed.

4.3.4. Case 4: Passing outside Normal Distribution (PO). In the PO case, the point is not located in the voxel and the ray is not passing the distribution, as represented by Figure 9. This case can be determined by (4). This case causes three candidates: normal, modified, and deleted. The three cases are the same as all possible changes in a voxel with a normal distribution. This means that the PO case cannot
provide any inference to detect map changes based on the measurement. Accordingly, the mass \( m_{\text{map}}^{\text{PO} - i,j}(\text{state}) \) in the PO case is modeled by

\[
\begin{align*}
    m_{\text{map}}^{\text{PO} - i,j}(\phi) &= 0, \\
    m_{\text{map}}^{\text{PO} - i,j}(S) &= 0, \\
    m_{\text{map}}^{\text{PO} - i,j}(C) &= 0, \\
    m_{\text{map}}^{\text{PO} - i,j}(\Omega) &= 1.
\end{align*}
\] (8)

4.3.5. Case 5: Blocking in Empty Voxel (BE). The point is measured in the empty voxel in the BE case, as shown in Figure 9. In this case, only the new class is inferred. Accordingly, the mass \( m_{\text{map}}^{\text{BE} - i,j}(\text{state}) \) of the BE case is described by

\[
\begin{align*}
    m_{\text{map}}^{\text{BE} - i,j}(\phi) &= 0, \\
    m_{\text{map}}^{\text{BE} - i,j}(S) &= 0, \\
    m_{\text{map}}^{\text{BE} - i,j}(C) &= \lambda_{\text{block}}, \\
    m_{\text{map}}^{\text{BE} - i,j}(\Omega) &= 1 - \sum_{A \in \Omega} m_{\text{map}}^{\text{BE} - i,j}(A),
\end{align*}
\] (9)

where the parameter \( \lambda_{\text{block}} \) represents the probability for the voxel to be changed. The parameter is the same as the parameter in the BI case because it is modeled such that the effect of the blocked point provides the same inference for the map change.

4.3.6. Case 6: Passing through Empty Voxel (PE). When the ray is passing the empty voxel, the voxel is placed in the PE case. As shown in Figure 9, the candidates of the map change class are composed of the empty or new classes. Because the map change class of the voxel without a normal distribution must be either empty or new, the measurement does not provide any inference for the voxel. As a result, the mass \( m_{\text{map}}^{\text{PE} - i,j}(\text{state}) \) of the PE case can be represented by

\[
\begin{align*}
    m_{\text{map}}^{\text{PE} - i,j}(\phi) &= 0, \\
    m_{\text{map}}^{\text{PE} - i,j}(S) &= 0, \\
    m_{\text{map}}^{\text{PE} - i,j}(C) &= 0, \\
    m_{\text{map}}^{\text{PE} - i,j}(\Omega) &= 1.
\end{align*}
\] (10)

4.4. Map Change Detection Based on Integrating Masses of Map Changes. The six cases for the geometric relationship provide the mass \( m_{\text{map} - i,j}(\text{state}) \) for map change states as shown in Figure 5. Because the voxel \( v_{\text{map}} \) is measured iteratively by points \( \{p_j|1 \leq j \leq N_{j}\} \) from intelligent vehicle \( i \), the voxel \( v_{\text{map}} \) has multiple masses \( m_{\text{map} - i,j}(\text{state}) \) to provide inference for the map change. To detect the map change precisely, the measured masses \( m_{\text{map} - i,j}(\text{state}) \) are integrated into one mass. To integrate the masses, Dempster-Shafer’s combination rule is applied, as described by

\[
\begin{align*}
    m_{\text{v}_{12}}(A) &= \sum_{B \cap C = A|B,C \subseteq \Omega} m_1(B) \cdot m_2(C), \\
    m_{\text{v}_{10}}(A) &= \frac{m_{\text{v}_{12}}(A)}{1 - m_{\text{v}_{12}}(\phi)}, \quad \forall A \subseteq \Omega, \ A \neq \phi, \\
    m_{\text{v}_{10}}(\phi) &= 0.
\end{align*}
\] (11)

To merge the masses to the change state in each voxel, the masses in each voxel must be initialized. All voxels are initialized as unknown states as (12) because voxels do not have any inferences at first.
\[ \begin{align*}
m_{\text{map}}(\phi) &= 0, \\
m_{\text{map}}(S) &= 0, \\
m_{\text{map}}(C) &= 0, \\
m_{\text{map}}(\Omega) &= 1. \\
\end{align*} \] (12)

The result of all mass integration measured by intelligent vehicle \( i \) is denoted by \( m_{\text{map}^{-i}} \) (state), which includes \( m_{\text{map}^{-i}}(S) \), \( m_{\text{map}^{-i}}(C) \), \( m_{\text{map}^{-i}}(\Omega) \), and \( m_{\text{map}^{-i}}(\phi) \). The mass \( m_{\text{map}^{-i}}(\phi) \) must be 0 by (11). Finally, the maximum value within three masses \( m_{\text{map}^{-i}}(\text{state}) \) determines the ray-casting-based map change state \( s_{\text{map}^{-i}} \), composed of sustained, changed, and unknown.

Figure 10 shows the examples of six map change classes (unmeasured, empty, normal, new, modified, and deleted), which can appear in the situation for the map change detection. The unmeasured class is added into the basic five map change classes to consider the unmeasured situation due to limited LiDAR ranges. The unmeasured class is classified as the unknown state by the initial masses (12) because any rays do not reach the voxel. For the empty class, there is no normal distribution and no points are blocked in the voxel. It means that all measurements are allocated to the PE case. Accordingly, after the merging (11), the merged mass can be propagated to the unknown state. The two map change classes are not distinguished as map changes because the change states of the two cases are unknown state. On the other hand, for the normal class, the measurements from the LiDAR points are mostly the BI cases. Although there may be some mis-measurements such as the PE and BO cases, the merged mass can be propagated to the sustained state by evidence merging (10). Therefore, the normal class is not also distinguished as map changes. Different from the unmeasured, empty, and normal classes, the remaining three classes (new, modified, and deleted) must be detected as map changes. For the new class, most of points are blocked at the voxel (BE case); therefore, the voxel can be classified as the changed state. The voxel with the modified case has three BO cases, one BI case, and two PI cases as measurements. The result of merging of six measurements propagates the voxel to the changed state. The deleted case can be estimated as the changed state based on four PI cases and two PE cases.

As shown in Figure 11, the proposed ray-casting-based algorithm can solve two problems caused by the V2V comparison-based change detection algorithm in Figure 4. Voxel (1) is estimated as the sustained state updated by the measurements in the BI case, which means that the voxel is classified as the normal class different to voxel (1) of Figure 4. On the other hand, voxels (2) and (3) can be distinguished based on the ray-casting approach. Voxel (2) is classified as the unmeasured class because no measurement is reached in the voxel. Voxel (3) is estimated as changed state due to the measurement update in the PI case. Since there are no blocked points in the voxel, the voxel can be classified as the deleted class.

4.5. Uploading of Map Changes. When the voxel \( v_{\text{map}} \) is estimated as the changed state (new, modified, or deleted classes), the voxel \( v_{\text{map}} \) is uploaded to the map cloud server as map change information. The map change voxels are categorized into two groups: a new normal distribution group and no normal distribution group. If the voxel is classified as changed state by the blocking points as shown in new and modified cases of Figure 10, a new normal distribution is constructed and uploaded to the map cloud server with the voxel index information. On the other hand, if the voxel does not have any blocked points, the only voxel index is uploaded to the map server as the no normal distribution part. The regions to be uploaded are determined by the position of the ego-vehicle. When the vehicle has escaped from the boundary of a geodetic tile of the GND map, the map changes are serialized and stored in a file based on Google Protocol Buffers. The file with the map changes can be uploaded to the map cloud server through network protocols such as the http, FTP, WebDAV, and SAMBA.

5. GND Map Update Based on Crowd-Sourced Changes in Cloud Server

5.1. Clustering Based on Crowd-Sourced Data. When the differences between the environments and the GND map, the \( N_i \) intelligent vehicles can detect and upload the map change voxels \( \{ v_i \mid 1 \leq i \leq N_i \} \) for the voxel \( v_{\text{map}} \). The map change voxel \( v_i \) detected by \( i \)-th vehicle consists of the existence of the normal distribution \( \epsilon_i \), the number of points \( n_i \), mean \( \mu_i \), and covariance \( \Sigma_i \). If the map change voxel includes a normal distribution as a new normal distribution group, \( n_i \), \( \mu_i \), and \( \Sigma_i \) are defined. Otherwise, the parameters are not defined. However, because the map change voxels have been detected by low-cost sensors, they are not always precise, and inaccurate information can be uploaded. Therefore, it is essential to avoid inaccurate voxels from crowd-sourced voxels and merge crowd-sourced voxels into an accurate voxel to update the base map. To process two functions simultaneously, density-based spatial clustering of applications with noise (DBSCAN) is applied [52]. The DBSCAN algorithm generates clusters based on the distance between two points. Unfortunately, because the targets of the algorithm in the map change merging process are not points but crowd-sourced map change voxels (new, modified, and deleted classes), the distance-based DBSCAN algorithm cannot be directly applied. To solve this problem, the similarity between two voxels is applied instead of the distance between two points.

The similarity \( d_{i,j} \) between two voxels \( v_i \) and \( v_j \) is composed of three parts: existence similarity \( d_{i,j}^{\epsilon} \), L2 distance-based similarity \( d_{i,j}^{L2} \), and geometric similarity \( d_{i,j}^{g} \). The existence similarity \( d_{i,j}^{\epsilon} \) evaluates the existence between two voxels. The L2 distance-based similarity \( d_{i,j}^{L2} \) and geometric similarity \( d_{i,j}^{g} \) evaluate the similarities between two distributions of two voxels. If at least one of the two distributions does not exist, the L2 distance-based similarity \( d_{i,j}^{L2} \) and geometric similarity \( d_{i,j}^{g} \) are not calculated. Since the three similarities take values from 0 to 1, the total similarity
is represented as the product of three similarities to apply the DBSCAN algorithm in

\[ d_{i,j} = d_{i,j}^e \times d_{i,j}^{L2} \times d_{i,j}^d. \]  \hspace{1cm} (13)

The voxel without a normal distribution has an existence \( \varepsilon_j \) of false. To reflect the voxel without a normal distribution, the existence similarity is considered. The existence similarity compress the two existences \( \varepsilon_i \) and \( \varepsilon_j \) between two uploaded voxels \( v_i \) and \( v_j \) according to

\[ d_{i,j}^e = \begin{cases} 1, & \text{if } \varepsilon_i = \varepsilon_j = \text{true}, \\ 0, & \text{otherwise.} \end{cases} \]  \hspace{1cm} (14)

---

**Figure 10:** Examples of results of evidence merging.

**Figure 11:** Map change detection based on ray-casting approach.
The L2 distance-based similarity is calculated only in case that the two uploaded voxels have two distributions. The L2 distance-based similarity, used as a constraint for optimization of an distribution-to-distribution matching [25], is represented by the likelihood of the Mahalanobis distance between two distributions in

$$d_{i,j}^{L2} = \exp\left(-\frac{(\mu_j - \mu_i)^T(\Sigma_j + \Sigma_i)^{-1}(\mu_j - \mu_i)}{2}\right). \quad (15)$$

The similarity $d_{i,j}^{L2}$ takes a value from 0 to 1, with the value 1 representing the best similar relationship between two distributions.

Although the L2 distance-based similarity $d_{i,j}^{L2}$ can provide a straightforward distance between two distributions, it encounters a problem. Although the two means $\mu_i$ and $\mu_j$ are located near each other and two shapes are very different from each other, the L2 distance-based similarity $d_{i,j}^{L2}$ is approximately 1 because the difference $\mu_j - \mu_i = 0$ makes the similarity to be 1. To solve this problem, geometric similarity, which is based on geometric shape evaluation, is applied [53]. Based on linear algebra, the geometric similarity can classify a distribution as one of three types: sphere, plane, and line. By the eigenvalues $\lambda_1$, $\lambda_2$, and $\lambda_3$ ($\lambda_1 < \lambda_2 < \lambda_3$) of normal distribution $N(\mu_i, \Sigma_i)$ derived by eigendecomposition of the covariance $\Sigma_i$, the type $t_i$ can be determined by

$$t_i = \begin{cases} 
\text{SPHERE}, & \lambda_3 \approx \lambda_2 \approx \lambda_1 \gg 0, \\
\text{PLANE}, & \lambda_3 \approx \lambda_2 \gg \lambda_1 \approx 0, \\
\text{LINE}, & \lambda_3 \gg \lambda_2 \approx \lambda_1 \approx 0. 
\end{cases} \quad (16)$$

When the ratio of $\lambda_2$ to $\lambda_1$ is larger than 10, $\lambda_2 \gg \lambda_1$ is achieved. If two distributions have the same type, the geometric shapes are compared. The normal vector $n_i$ at the PLANE type and the direction vector $d_i$ at the LINE type are used as feature vectors to compare geometric shapes. When the shapes of two distributions are different, the dot product of feature vectors of two distributions is lower. Using this concept, the geometric similarity can be derived through

$$d_{i,j}^{G} = \begin{cases} 
0, & t_i \neq t_j, \\
\langle n_i, n_j \rangle, & t_i = t_j = \text{PLANE}, \\
\langle d_i, d_j \rangle, & t_i = t_j = \text{LINE}. 
\end{cases} \quad (17)$$

5.2. Merging of Crowd-Sourced Map Information. The DBSCAN algorithm based on the similarity in (13) derives multiple clusters $\{c_j|1 \leq j \leq N\}$ from multiple map change voxels $\{|v|1 \leq i \leq N\}$, as shown in Figure 12. Each cluster $c_j$ is a union of multiple map change voxels. There are two types of clusters: empty type and distribution type. An empty type cluster example is $c_1$ in Figure 12, which means that all map change voxels in the cluster are empty voxels, i.e., $c_1 = \text{false}$. There can be only one cluster grouped as the empty type in a voxel because the empty voxel always has similarity 1 with other empty voxels. Accordingly, the representative cluster with the empty type is always set to an empty voxel. On the other hand, the distribution type clusters represented by clusters $c_2$ and $c_3$ in Figure 12 mean that clusters are grouped by the map change voxels with the distributions. The representatives of the clusters with distribution type can be derived by the recursive update of sample mean and covariance [54].

5.3. Publication of Definite Changed Map Information. There are several representatives constructed from each cluster, as shown in Figure 12(c). To publish the updated map information to the base map, there are several processes. First, the numbers of map change voxels in the clusters are compared, as shown in Figure 13. As a result, the representative normal distribution with the maximum number of merged map change voxels is selected as the final map candidate to be updated. Secondly, validation based on the ratio of uploaded information is evaluated. The ratio of the number of merged voxels to the number of information uploads must be higher than the configured parameter $\xi_{\text{update}}$ in Figure 13(b) to update the base map. The final process is a comparison of the final map candidate with the base map. The voxel in the base map and the voxel in the final map candidate are compared based on the similarity $d_{\text{map, candidate}}$ in (13), as shown in Figure 13. When the similarity is larger than $d_{\text{update}}$, as shown in Figure 13(b), the map candidate is published to the base map.

6. Simulations

6.1. Simulation Environments. To evaluate the map update framework, the environments with large changes such as construction sites are required. The site must also make the high-precision GNSS/INS information provide the precise position without any noises due to evaluation of robust localization in changing environments. Unfortunately, it is difficult to find the sites to satisfy both conditions for evaluation in practice. Therefore, the map update framework was evaluated in the simulations to construct the changing environments.

The simulation environments consisted of virtual perception data and real data measured by a test vehicle. The test vehicle installed with on-board sensors (yaw rate and wheel speed sensors), a low-cost GNSS receiver within 2.5 m positioning errors (U-Blox EVK-6T), and a high-precision GNSS/INS within 0.01 m positioning errors (OXTS RT3002) acquired the vehicle motion and positioning data as real data. The motion and positioning information measured by the on-board sensors and the low-cost GNSS receiver was used as input data of a vehicle part performing the map change detection algorithm in the proposed framework. In order to construct virtual perception data, an in-house LiDAR simulator generated some point cloud measurements based on a ray-casting approach from the vehicle pose to the virtually modeled environments. The positioning information measured by the high-precision GNSS was used as the ground truth of the vehicle pose to construct the LiDAR
measurements. As shown in Figure 14(a), the base GND map was constructed by the simulated LiDAR measurements along with the vehicle poses. In order to simulate the changing environments, the virtually modeled environments were modified by manual editing including adding/removing/transforming some objects. Accordingly, the editing of the environments provided the changed GND map after environment changes in Figure 14(b).

In the simulator, the LiDAR measurements were modeled considering the LiDAR specifications such as the number of layers, the horizontal resolution, and the vertical resolution. In order to simulate the crowdsourcing information measured from various vehicles, various sensor configurations including Velodyne VLP-16, HDL-32E, and HDL-64E, Robosense RS-LiDAR-16, RS-LiDAR-32, and Valeo Scala were applied in the simulations. For evaluating the accuracy of the GND map updated by the proposed framework, the simulations were accomplished in a local computer without a cloud computing system.

6.2. Robust Localization in Map-Changing Environments. For detecting map changes in the GND map, the exact pose of the vehicle is very important. In order to estimate the vehicle pose in changing environments, we applied a hierarchical algorithm [44, 55]. In the first process of the hierarchical architecture, a submap, which models the present environments in real-time, is constructed based on the Graph SLAM algorithm. Since our previous work [44] evaluated the performance of the algorithm in the PCD map only, we evaluated the performance of the localization with the submap approach in the GND map. To estimate the precise pose, we applied a Graph SLAM algorithm because the algorithm generally provides better performances than the Kalman filter algorithm and the map update does not require real-time processing.
Figure 14: GND maps (a) before environment changes and (b) after environment changes. The circles (1)–(3) indicate the differences between two maps.

Figure 15 shows the localization performances of the general Graph SLAM algorithm and the Graph SLAM with the submap concept through errors to the longitudinal, lateral, and heading directions. The errors were evaluated by the ground truth of the vehicle pose measured by the high-precision GNSS/INS. The differences between the GND maps as shown in circles (1)–(3) of Figure 14 are represented from 9 to 17 seconds in Figure 15. The hierarchical Graph SLAM algorithm with the submap concept had better performances than the general Graph SLAM algorithm without the submap concept because a matching ratio, which represents the ratio of matched regions over the measurements in the map-matching process, was higher by the submap.

To judge whether the localization can be used in the map update framework or not, localization performances were validated through localization requirements for autonomous driving proposed by Ford [56]. The Ford localization requirements consist of two criteria in three directions: maximum error boundary and 95% error boundary. In order to satisfy the maximum error boundary criterion, the longitudinal, lateral, and heading errors have to be located within the 0.44 m, 0.44 m, and 0.50° represented by yellow rectangles of Figure 15, respectively. On the other hand, to satisfy the 95% error boundary, the longitudinal, lateral, and heading errors, located in 95% error position from ascending ordering arrays, must be lower than the 0.15 m, 0.15 m, and 0.17° represented by green rectangles of Figure 15, respectively.

The maximum and 95% errors to the longitudinal, lateral, and heading directions from the general Graph SLAM and the hierarchical Graph SLAM are represented in Table 2. If the criterion is satisfied, the blank of the table is filled with the green color. Otherwise, the black is filled with the red color. Since the general Graph SLAM without the submap concept cannot satisfy the lateral and heading requirements, the algorithm cannot be used to the map update framework. On the other hand, the hierarchical Graph SLAM with the submap concept can be used to the map update framework because all criteria in Ford requirements are satisfied.

6.3. Map Change Detection and Map Update. The proposed map change detection algorithm can detect map changes from the base GND map using the relationship between the map and the measurements. The map changes, which are estimated as change states including new, modified, and deleted classes, are uploaded to the map cloud server. The map change information makes the changed map information be inferred as shown in Figure 16(a). The new, modified, and deleted classes are represented as red, yellow, and green ellipsoids, respectively. The normal class, represented by blue ellipsoids, is not uploaded to the map cloud server. The uploaded crowd-sourced map changes, detected by crowd-sourcing vehicles, are merged into the representative map changes through the map update algorithm in the proposed framework. As a result, the updated map can be represented in Figure 16(b). In order to validate the performance, the ground truth of map changes are represented in Figure 16(c), which is constructed by comparing the previous GND map (a) and the present GND map (b) of Figure 15.

By comparing with the ground truth of map changes, the performances of proposed algorithms are represented in confusion matrices, which are generally used to evaluate a classification problem, as shown in Table 3. The confusion matrices to represent performances of map changes include five map change classes: normal, new, modified, deleted, and empty. Since the outputs of the algorithm are compared with the ground truth of map changes, higher diagonal values mean better performances. Table 3 includes three confusion matrices, comparisons with (1) the base map, (2) results of map change detection, and (3) results of map update. The first one represents the confusion matrix compared with the base map without map change consideration, as shown in Figure 15(a). Since the base map has been not updated, there is no prediction for map changes (new, modified, and deleted) in the confusion matrix. The second one means the confusion matrix compared with the map inferred by the map change detection based on single driving, as shown in Figure 16(a). Although map changes can be detected, some errors still occur due to two reasons. The parked vehicles cause the inaccurate estimate from the true empty class to the predicted new class. Also, occlusion regions by parked vehicles cause misclassification from the true deleted class to the predicted normal class. The third one is the confusion matrix compared with the map updated by the map change update, as shown in Figure 16(b). The errors caused in the second confusion matrix are reduced to zero based on the crowd-sourcing measurements.

\[
F1 = 2 \times \frac{\text{precision} \times \text{recall}}{\text{precision} + \text{recall}}
\] (18)
Table 2: Table of robust localization performance in simulations.

| Experiments      | Longitudinal (m) | Lateral (m) | Heading (°) |
|------------------|------------------|-------------|-------------|
|                  | 95%: 0.15        | Max: 0.44   | 95%: 0.15   | Max: 0.44   | 95%: 0.17   | Max: 0.5    |
| Without submap   | 0.08412          | 0.2002      | 0.30124     | 1.4518      | 1.1388      | 6.2191      |
| With submap      | 0.031013         | 0.054553    | 0.012499    | 0.030573    | 0.052715    | 0.094034    |

Figure 15: Performance of robust localization in map-changing simulations.

Figure 16: Map change information (a) detected by the map change detection algorithm based on single driving, (b) updated by the proposed framework based on crowd-sourced data, and (c) representing ground truth information.
Table 3: Confusion matrices about non-update, change detection based on single driving, and map update based on crowd-sourced data in simulations.

| Truth | Base map | Recall (%) | Map change detection | Recall (%) | Map update | Recall (%) |
|-------|----------|------------|----------------------|------------|------------|------------|
|       | Normal   | 100        | Normal               | 99.78      | Normal     | 99.98      |
| Normal| 475      | 0          | 0                    | 0          | 0          | 0          |
| New   | 0        | 0          | 0                    | 0          | 0          | 0          |
| Modified| 17     | 0          | 0                    | 0          | 0          | 0          |
| Deleted| 135      | 0          | 0                    | 0          | 0          | 0          |
| Empty | 0        | 0          | 0                    | 0          | 0          | 0          |
|       | 0        | 0          | 0                    | 0          | 0          | 0          |

Precision (%)  
75.75  0  0  0  99.50  F1-score  96.34  99.13  99.98  F1-score  99.62

F1-score  37.36  88.05

16 Journal of Advanced Transportation
To evaluate the overall performance based on a unit index, we adopt the F1-score (18). The F1-score is widely used in the situation that the number of voxels in each class is not balanced (i.e., empty ≫ normal ≫ new, modified, deleted). The F1-score by the base map, results of map change detection, and results of map update are represented as 37.36%, 88.05%, and 99.62%, respectively.

Figure 17 represents the effects of the number of vehicles to join the map update. The blue line of Figure 17 means the F1-score by the base map. The yellow line represents the F1-score by results of the map change detection based on single driving. The green line represents the results of the map update based on the number of crowd-sourcing vehicles. More crowd-sourcing vehicles construct a more precise map by considering more map change information, and the performance of the map update is saturated after 9 vehicles.

7. Experiments

7.1. Experimental Environments. The test vehicle, used in the simulations, was also used in the experiments. Differently with the usage of the virtual perception sensors in simulations, the experiments used two real LiDAR sensors (Velodyne VLP-16). Although two LiDAR sensors were used to construct the base map, only one LiDAR sensor was used to update the map information based on the proposed system. The high-precision GNSS/INS information was not used for the proposed system but used for the base map construction and ground truth positioning. The low-cost GNSS was only used to solve the kidnapped problem for localization.

As the paper said in Section 6.1, there were two requirements from a test site for evaluation of the proposed map update system. In order to evaluate the map update system, the environment must be changed from the base map. In addition, to use the high-precision GNSS/INS as ground truth information, the quality of the GNSS must be good in the test site. The chosen place that met the requirements was Wangsimni-ro in Korea, as shown in Figure 18. In the test site, the test vehicle was driven to construct the base map on August 1, 2019. Next, the test vehicle was driven 38 times on the same road to update the base map on October 12, 2019.

The raw PCD map was constructed by accumulating point clouds measured by two LiDAR sensors based on the trajectory of the ego-vehicle measured by high-precision GNSS/INS. To convert the PCD map to the GND map, the points in the PCD map were allocated to each voxel with 1 m basic size. Based on the points, the normal distribution in each voxel could be constructed to form the GND map structure.

7.2. Robust Localization in Map-Changing Environment. The real environment confronts the changes of the road over time, such as the construction site, growing trees, and parked vehicles. Accordingly, it is essential to estimate the precise pose considering the map-changing situations. To reflect the map-changing environments, postprocessing localization based on hierarchical Graph SLAM with the submap concept is applied [44].

To evaluate variations regarding the submap, the errors of localization algorithms without the submap and with the submap are shown in Figure 19. The brown and red lines represent the localization performance without and with the submap, respectively. The yellow and green rectangles in Figure 19 represent the maximum and 95% error boundaries for the localization requirements, respectively. As shown in Figure 19, the notable environment changes occur from 20 to 31 seconds.

Localization without the submap has particularly varying errors in the partial map changes because incorrect map-matching constraints can be constructed by mismatching between the LiDAR measurements and the unmodified GND map. Although precise accuracy of localization is essential for map change detection that meets the localization requirements, the general Graph SLAM without the submap does not satisfy the requirements. To relieve the LiDAR mismatching between measurements and the GND map, the hierarchical Graph SLAM algorithm is applied with the submap concept. After applying hierarchical Graph SLAM, localization error can be reduced. The maximum and 95% errors are represented in Table 4.

The 95%/maximum errors of hierarchical Graph SLAM to the longitudinal and lateral directions are 0.1478/0.19742 m and 0.13104/0.20154 m, respectively. Since the errors are located within 0.15/0.44 m, the localization with the submap can satisfy the localization requirements. In addition, the 0.14799/0.2351° of 95%/maximum heading errors satisfy the localization requirements. Therefore, the robust localization algorithm based on hierarchical SLAM with the submap meets the requirements for localization, and it means that the localization algorithm can be used for the proposed framework.

7.3. Map Change Detection in GND Map Structure. The proposed map change detection and map update processes were evaluated in real environments. Different from simulations, the real environments, which have more noise, can
Figure 18: Test site for experiment: Wangsimni-ro in Korea. (a) Global trajectories, (b) base map environment at August 1, 2019, and (c) present map environment at October 12, 2019.

Figure 19: Performance of robust localization in map-changing experiments.
generate more positioning errors in the robust localization process. Especially, the rotation errors can affect the performance of the map change detection and the map update processes rather than the translation errors. In order to relieve effects of rotation errors, the map change detection is performed within 50 m from the ego-vehicle. The results based on the map change detection and the map update are represented in Figures 20(a) and 20(b). Figure 20(c) represents the ground truth of map changes constructing by comparison between the previous GND map and the present GND map.

The performances of proposed algorithms are evaluated using the confusion matrices compared with the ground truth of map changes (Figure 20(c)), as shown in Table 5. The evaluation was performed in the region from the ground to a height of 1.5 m because most of the changes are in that region. The first confusion matrix represents the results of comparison with the base map. Because the base map does not have any map change information, the precision rates and the recall rates related with the map change classes (new, modified, and deleted) were set to 0. Therefore, the F1-score based on the base map was 37.36%, which is a very low value. The second confusion matrix means the results of the map inferred by the map change detection based on single driving, as shown in Figure 20(a). The F1-score of map change detection was 78.34%. There are several reasons for degrading the accuracy of the map change detection algorithm. First, the precision rate of the new class is 35.92% because the dynamic points that are misclassified as static are propagated to the new class, as shown in the circles of Figure 20(a). Secondly, the precision rate of the modified class is 25.37% because the true normal class is misclassified as modified class. Because low precision rate is caused by incorrectly constructed new normal distributions in the map.

| Experiments   | Longitudinal (m) | Lateral (m) | Heading (°) |
|---------------|------------------|-------------|-------------|
|               | 95%: 0.15 | Max: 0.44 | 95%: 0.15 | Max: 0.44 | 95%: 0.17 | Max: 0.5 |
| Without submap | 0.2299 | 0.36875 | 0.28506 | 0.66626 | 0.41892 | 0.94196 |
| With submap    | 0.1478 | 0.19742 | 0.13104 | 0.20154 | 0.14799 | 0.2351 |

Figure 20: Map change information (a) detected by the map change detection algorithm based on single driving, (b) updated by the proposed framework based on crowd-sourced data, and (c) representing ground truth information.
Table 5: Confusion matrices about nonupdate, change detection based on single driving, and map update based on crowd-sourced data in experiments.

| Truth  | Base map | Recall (%) | Map change detection | Recall (%) | Map update | Recall (%) |
|--------|----------|------------|----------------------|------------|------------|------------|
|        | Normal   | 1999       | 0                    | 0          | 0          | 0          |
|        | New      | 0          | 0                    | 345        | 0          | 0          |
|        | Modified | 20         | 0                    | 0          | 0          | 0          |
|        | Deleted   | 291        | 0                    | 0          | 0          | 0          |
|        | Empty     | 0          | 0                    | 38507      | 100        | 0          |

| Precision (%) | 86.53 | 0 | 0 | 0 | 99.11 | F1-score 98.70 | 35.92 | 25.37 | 83.70 | 99.99 | F1-score 98.61 | 85.67 | 80.95 | 99.61 | 99.89 | F1-score 92.80 |
change detection process, the misclassification can be overcome after the crowd-sourced data-based evaluation process in the map cloud server. The third reason is the 83.7% precision rate of the deleted class, which is low. This means that the normal states in the previous base map are incorrectly removed. The misclassified voxels can be overcome by applying the change ratio $\xi_{\text{update}}$ to the merging process.

Based on the crowd-sourcing detection, the confusion matrix, representing the results of the map update based on crowd-sourcing detection (Figure 20(b)), made the F1-score be 92.8%. The precision rate of the new class increases dramatically from 32.92% to 85.67% due to the merging process in the map update. Similarly, the precision rate of the modified class compared with the precision rate of the map change detection algorithm increases from 25.37% to 80.95%. Because the class predicted as modified by the map change detection algorithm is compared with each other, only clear voxels remain. The precision rate of the deleted class also increases from 83.7% to 99.61%.

7.4. Comparison with the PCD Map Update. In order to compare the performance of the GND map update over the other approach, the framework of the PCD map update based on crowd-sourcing detection was used [44]. The PCD map, used for constructing the base GND map, was used as the base map. The changed information can be detected by the ray-casting approach in each vehicle. The detected map changes, which were composed of only changed points (new and deleted points), were uploaded to the map cloud server. Using the map change information, the PCD map was finally updated, as shown in Figure 21(a). For the fair evaluation, the same indicator, F1-score based on comparison of normal distributions, was used. Accordingly, the updated PCD map was converted to the updated GND map, as shown in Figure 21(b). The updated GND map was evaluated by comparing with the ground truth of map changes (Figure 20(c)).

Figure 22 shows the results of the PCD map update and the GND map update. The blue line represents the F1-score by the base map. The F1-scores of the base PCD map and the base GND map are the same. The yellow and green lines represent the F1-scores from the map change detection and the map update based on the GND map. On the other hand, the orange and purple lines represent the F1-score detected by the map change detection and the map update based on the PCD map. In using single driving data, the F1-score based on PCD map update is higher than the F1-score based on the GND map update because the ray-casting based on the PCD map can check the changes more in detail than the ray-casting based on the GND map. Although the degradation of the change detection based on the single driving data, the results of the map update were similar to each other after 16 vehicles due to the crowd-sourcing data.

7.5. Traffic of Wireless Network Communications. The uploaded size is dramatically reduced by 96.06% in the proposed framework.

The analysis of the traffic of the wireless networks is represented in Figure 23. The blue, orange, yellow lines in Figure 23 mean the data size which can be transmitted during the vehicle driving through average speeds of 3G, 4G, and 5G wireless networks in Korea. The purple and green lines in Figure 23(a) show the downloaded sizes of the PCD
map and the GND map during the vehicle driving. While the downloaded PCD map size was 19.37 MB, the downloaded GND map size is 3.36 MB, due to the compression based on the normal distributions. The map size was compressed by 82.6% using the GND map. Since the green line for the GND map is always lower than the blue line of the 3G wireless network, the 3G network can be used for downloading the GND map.

The purple and green lines in Figure 23(b) represent the uploaded size of map changes based on the PCD map and the GND map during the vehicle driving. The sky-colored line represents all measurement data (in-vehicle sensors, the low-cost GNSS, and the LiDAR measurements) used to update the map changes similar to the conventional camera-based map update approaches. The size of uploaded PCD map changes, which were composed of new and deleted points, was 2.06 MB, which was lower than all measurements. In the map change uploading, the adaptation of normal distributions with new, modified and deleted classes dramatically reduced 2.06 MB to 0.11 MB. A similar tendency was represented in Figure 23(c), which shows the total uploaded data from crowd-sourcing vehicles. While the data size for uploading all measurements and PCD map changes were used until 1713 MB and 89.24 MB during driving of 38 vehicles, the proposed framework used only 3.511 MB. Compared with the PCD map update framework, the uploaded size is dramatically reduced by 96.06% in the proposed framework.

7.6. Localization Performance after Map Update. The final goal of the GND map update is to perform precise online localization by keeping the GND map up-to-date. To evaluate the online localization performance, two data are compared, as shown in Figure 24. The brown line is the same as the brown line in Figure 19. Online localization with the previous map cannot satisfy the localization requirements. On the other hand, the red line represents the online localization performance based on the matching with the GND map after the map update. The results of localization before and after the map update were reinterpreted as Table 6. The maximum longitudinal and lateral errors were 0.17868 m and 0.20133 m,
respectively, which were within the maximum boundary of 0.44 m. The maximum heading error was 0.20966°, which was within the maximum boundary of 0.5°. In addition, the longitudinal, lateral, and heading 95% errors were 0.11053 m, 0.14431 m, and 0.20966°, respectively. Since the online localization algorithm with the updated map can satisfy the localization requirements, the localization based on the proposed framework can be used for autonomous driving.

8. Conclusion

There are some problems that LiDAR-based localization has a critical issue to be solved in order to realize autonomous driving in the future. Since LiDAR-based localization uses the LiDAR map modeling static environments, it can be fragile by the environment changes of static information (i.e., parked vehicle and construction site). Accordingly, it is essential for changes in the real environment to be periodically updated in the LiDAR map for minimization of the difference between the environment and the LiDAR map. However, it is difficult to apply directly to the periodic update because the LiDAR map is too large to be downloaded or uploaded.

In order to solve the problems, the paper proposed the GND map update framework based on crowd-sourcing data. The proposed framework consists of five steps: downloading, robust localization, change detection, uploading, and map update. The main contributions of the proposed framework are summarized as follows:

1. The paper proposed a GND map update framework based on crowd-sourcing detection. Based on the framework, the LiDAR map can be kept up-to-date to support the localization for autonomous driving. While the performance of the proposed framework is similar to the performance of the state-of-the-art framework based on crowd-sourcing data [44], the data size transmitted in the proposed framework is dramatically reduced by 82.6% in the downloading process and 96.06% in the uploading process. Using the updated map, the online LiDAR localization was successfully performed in changing environments.

2. The proposed change detection algorithm finds the probability that the environments are changed based on the relationship between the normal distribution and the LiDAR measurements. Accumulation of change information based on the evidence approach detects the deterministic map changes to be uploaded into the map server. Since the only map changes (new, modified, and deleted normal distributions) are uploaded into the map server, the uploaded data size
is reduced than the data size of uploading the changed points.

(3) The proposed map merging algorithm detects the representative map changes from the crowd-sourcing change information through a DBSCAN algorithm based on the proposed normal-distribution similarity. The representative map changes can update the GND map in the map server by replacing the changing parts.

This framework keeps map information up-to-date without the additional costs generated by professional mapping vehicles with mobile mapping systems. In addition, the map update framework reduced the wireless network burdens dramatically. Despite these innovations, there are some limitations that should be resolved as future works. First, the data size to update the map based on a camera (10 kB/km) is still less than the uploaded data size of the GND map changes (56.32 kB/km). To compress the size of the GND map more, the authors plan to research the GND map compression based on the deep learning approach. Second, the map update framework does not consider the semantic segmentation information of point clouds researched in deep learning fields. To improve the performance of the map update framework, the authors plan to research a map update approach considering the semantic information.

Data Availability

The dataset cannot be opened because the data are company properties.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported in part by the BK21 Plus Program (22A20130000045) under the Ministry of Education, South Korea, in part by the Industrial Strategy Technology Development Program under Grant 10039673, Grant 10060068, and Grant 10079961, the International Collaborative Research and Development Program (N0001992) under the Ministry of Trade, Industry, and Energy (MOTIE Korea), and in part by the National Research Foundation of Korea (NRF) Grant funded by the Korean Government (MEST) under Grant 2011-0017495 and 2020R1C1C1007739. Following are results of a study on the “Future Transportation Equipment Development of Regional Innovation Platform for Gwangju and Jeollanamdo” Project, supported by the Ministry of Education and National Research Foundation of Korea.

References

[1] N. Mattern, R. Schubert, and G. Wanielik, “High-accurate vehicle localization using digital maps and coherency images,” in Proceedings of the 2010 IEEE Intelligent Vehicles Symposium (IV), pp. 462–469, La Jolla, CA, USA, June 2010.
[2] K. Jo, Y. Jo, J. K. Suhr, H. G. Jung, and M. Sunwoo, “Precise localization of an autonomous car based on probabilistic noise models of road surface marker features using multiple cameras,” IEEE Transactions on Intelligent Transportation Systems, vol. 16, no. 6, pp. 3377–3392, 2015.
[3] X. Chen, W. Hu, L. Zhang, Z. Shi, and M. Li, “Integration of low-cost gns and monocular cameras for simultaneous localization and mapping,” in Sensors, vol. 18, no. 7, p. 2193, Multidisciplinary Digital Publishing Institute, 2018.
[4] J. Levinson, M. Montemerlo, and S. Thrun, “Map-based precision vehicle localization in urban environments,” in Robotics: Science and Systems, IEEE, 2007.
[5] J. Levinson and S. Thrun, “Robust vehicle localization in urban environments using probabilistic maps,” in Proceedings of the 2010 IEEE International Conference on Robotics and Automation (ICRA), pp. 4372–4378, IEEE, Anchorage, AK, USA, May 2010.
[6] J. Choi, “Hybrid map-based SLAM using a Velodyne laser scanner,” in Proceedings of the 2014 17th International IEEE Conference on Intelligent Transportation Systems (ITSC), pp. 3082–3087, IEEE, Qingdao, China, October 2014.
[7] S. Kuutti, S. Fallah, K. Katsaros, M. Dianati, F. Mccullough, and A. Mouzakitis, “A survey of the state-of-the-art localization techniques and their potentials for autonomous vehicle applications,” IEEE Internet of Things Journal, vol. 5, no. 2, pp. 829–846, IEEE, 2018.
[8] S. Sood, “Self-healing maps for autonomous driving,” in Proceedings of the 2017 NVIDIA GPU Technology Conference, San Jose Convention Center, Silicon Valley, San Jose, CA, USA, 2017.
[9] TomTom, “Qualcomm and TomTom to Crowdsource High-Definition Map Data,” https://www.roadtraffic-technology.com/news/newqualcomm-and-tomtom-to-crowd-source-high-definition-map-data-5752324/.
[10] Mobileye, “Road experience management (REM),” [Online]. Available: https://www.mobileye.com/our-technology/rem/.
[11] Bosch, “Bosch road signature,” https://www.bosch-mobility-solutions.com/en/products-and-services/passenger-cars-and-light-commercial-vehicles/automated-driving/road-signature/.
[12] Daimler, “Daimler and HERE to bring HD Live Map to future Mercedes-Benz models,” https://media.daimler.com/marsMediaSite/en/instance/ko/Daimler-and-HERE-to-bring-HD-Live-Map-to-future-Mercedes-Benz-models.xhtml?oid=33472763.
[13] C. Kim, K. Jo, B. Bradai, and M. Sunwoo, “Multiple vehicles based new landmark feature mapping for highly autonomous driving map,” in Proceedings of the 2017 14th Workshop on Positioning, Navigation and Communications (WPNC), vol. 2018, pp. 1–6, Bremen, Germany, 2017.
[14] C. Kim, S. Cho, M. Sunwoo, and K. Jo, “Crowd-sourced mapping of new feature layer for high-definition map,” in Sensors, vol. 18, no. 12, p. 4172, Multidisciplinary Digital Publishing Institute, 2018.
[15] K. Jo, C. Kim, and M. Sunwoo, “Simultaneous localization and map change update for the high definition map-based autonomous driving car,” Sensors, vol. 18, no. 9, p. 3145, 2018.
[16] C. Kim, S. Cho, and M. Sunwoo, “A geodetic normal distribution map for long-term LiDAR localization on earth,” IEEE Access, vol. 9, pp. 470–484, Bremen, Germany, 2020.
[17] A. D. Stewart and P. Newman, “Laps-localisation Using Appearance of Prior Structure: 6-dof Monocular Camera Localisation Using Prior Pointclouds,” in Proceedings of the 2012 IEEE International Conference on Robotics and
[18] W. Maddern, G. Pascoe, and P. Newman, “Leveraging experience for large-scale LiDAR localisation in changing cities,” in Proceedings of the 2015 IEEE International Conference on Robotics and Automation (ICRA), pp. 1684–1691, Seattle, WA, USA, May 2015.

[19] Y. Xu, V. John, and S. Mita, “3D point cloud map based vehicle localization using stereo camera,” in Proceedings of the 2017 IEEE Intelligent Vehicles Symposium (IV), pp. 487–492, Los Angeles, CA, USA, June 2017.

[20] Z. Liu, H. Chen, H. Di et al., “Real-time 6d lidar slam in large scale natural terrains for ugv,” in Proceedings of the 2018 IEEE Intelligent Vehicles Symposium (IV), pp. 662–667, Changshu, China, June 2018.

[21] S. Thrun, W. Burgard, and D. Fox, “A real-time algorithm for mobile robot mapping with applications to multi-robot and 3D mapping,” in Proceedings of the 2000 IEEE International Conference on Robotics and Automation (ICRA), vol. 1, pp. 321–328, San Francisco, CA, USA, April 2000.

[22] G. Grisetti, C. Stachniss, W. Burgard, and others, “Improved techniques for grid mapping with rao-blackwellized particle filters,” IEEE Transactions on Robotics, vol. 23, no. 1, p. 34, 2007.

[23] C. Kim, S. Cho, C. Jang, M. Sunwoo, and K. Jo, "Evidence filter of semantic segmented image from around view monitor in automated parking system," IEEE Access, vol. 7, Article ID 92791, 2019.

[24] S. Cho, C. Kim, K. Jo, and M. Sunwoo, “A GPU accelerated particle filter based localization using 3D evidential voxel maps,” SAE Technical Paper 2019-01-0491, 2019.

[25] T. Stoyanov, M. Magnusson, H. Andreasson, and A. J. Lilienthal, “Fast and accurate scan registration through minimization of the distance between compact 3D NDT representations,” The International Journal of Robotics Research, vol. 31, no. 12, pp. 1377–1393, Sage Publications Sage UK, London, England, 2012.

[26] E. Javanmardi, M. Javanmardi, Y. Gu, and S. Kamijo, “Adaptive resolution refinement of NDT map based on localization error modeled by map factors,” in Proceedings of the 2018 IEEE 21st International Conference on Intelligent Transportation Systems (ITSC), pp. 2237–2243, IEEE, Maui, HI, USA, November 2018.

[27] R. W. Wolcott and R. M. Eustice, “Fast LiDAR localization using multiresolution Gaussian mixture maps,” in Proceedings of the 2015 IEEE International Conference on Robotics and Automation (ICRA), vol. 2015, pp. 2814–2821, IEEE, Seattle, WA, USA, May 2015.

[28] R.W. Wolcott and R.M. Eustice, “Robust LiDAR localization using multiresolution Gaussian mixture maps for autonomous driving,” The International Journal of Robotics Research, vol. 36, no. 3, pp. 292–319, 2017.

[29] D. Girardeau-Montaut, M. Roux, R. Marc, and Guillaume Thibault, “Change detection on points cloud data acquired with a ground laser scanner,” The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, vol. 36, 2005.

[30] L. Wellhausen, D. Renaud, G. Abel, S. Roland, and C. Cesar, “Reliable real-time change detection and mapping for 3D LiDARs,” in Proceedings of the 2017 IEEE International Symposium on Safety, Security and Rescue Robotics (SSRR), pp. 81–87, IEEE, Shanghai, China, October 2017.

[31] J. P. Underwood, D. Gillisjo, T. Bailey, and V. Vlaskine, “Explicit 3D change detection using ray-tracing in spherical coordinates,” in Proceedings of the 2013 IEEE International Conference on Robotics and Automation (ICRA), pp. 4735–4741, Karlsruhe, Germany, May 2013.

[32] F. Ferri, M. Gianni, M. Menna, and F. Pirri, “Dynamic obstacles detection and 3D map updating,” in Proceedings of the 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 5694–5699, IEEE, Hamburg, Germany, October 2015.

[33] W. Xiao, B. Vallet, and N. Paparoditis, “Change detection in 3D point clouds acquired by a mobile mapping system,” ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences, vol. II, pp. 331–336, 2013.

[34] X. Wen, B. Vallet, M. Brédif, and N. Paparoditis, “Street environment change detection from mobile laser scanning point clouds,” ISPRS Journal of Photogrammetry and Remote Sensing, vol. 107, pp. 38–49, Elsevier, 2015.

[35] W. Burgard, C. Stachniss, and D. Hähnle, “Mobile robot map learning from range data in dynamic environments,” Autonomous Navigation in Dynamic Environments, vol. 35, pp. 3–28, 2007.

[36] T. Krajnik, J. P. Fentanes, M. Hanheide, and T. Duckett, “Persistent localization and life-long mapping in changing environments using the frequency map enhancement,” in Proceedings of the 2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 4558–4563, Daejeon, Republic of Korea, October 2016.

[37] T. Krajnik, J. P. Fentanes, J. M. Santos, and T. Duckett, “Fremen: frequency map enhancement for long-term mobile robot autonomy in changing environments,” IEEE Transactions on Robotics, vol. 33, no. 4, pp. 964–977, IEEE, 2017.

[38] G. Trehard, Z. Alsayed, E. Pollard, B. Bradaï, and F. Nashashibi, “Credibilist simultaneous localization and mapping with a LiDAR,” in Proceedings of the 2014 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 2699–2706, Iros, Chicago, IL, USA, September 2014.

[39] G. Trehard, E. Pollard, B. Bradaï, and F. Nashashibi, “On line mapping and global positioning for autonomous driving in urban environment based on evidential SLAM,” in Proceedings of the 2015 IEEE Intelligent Vehicles Symposium (IV), pp. 814–819, Montreal, QC, Canada, May 2015.

[40] U. Katsura, K. Matsumoto, A. Kawamura, and T. Ishigami, “Spatial change detection using voxel classification by normal distributions transform,” in Proceedings of the 2019 IEEE International Conference on Robotics and Automation (ICRA), pp. 2953–2959, 2019.

[41] J. P. Saarinen, H. Andreasson, T. Stoyanov, and A. J. Lilienthal, “3D normal distributions transform occupancy maps: an efficient representation for mapping in dynamic environments,” The International Journal of Robotics Research, vol. 32, no. 14, pp. 1627–1644, 2013.

[42] J. Saarinen, H. Andreasson, T. Stoyanov, J. Ala-Luhtala, and A. J. Lilienthal, “Normal distributions transform occupancy maps: application to large-scale online 3D mapping,” in Proceedings of the 2013 IEEE International Conference on Robotics and Automation (ICRA), pp. 2233–2238, IEEE, Karlsruhe, Germany, May 2013.

[43] E. Einhorn and H. Michael Gross, “Generic NDT mapping in dynamic environments and its application for lifelong SLAM,” in Robotics and Autonomous Systems, vol. 69, no. 1, pp. 28–39, Elsevier, 2015.

[44] C. Kim, S. Cho, M. Sunwoo, P. Resende, B. Bradaï, and K. Jo, “Updating point cloud layer of high definition (HD) map
based on crowd-sourcing of multiple vehicles installed LiDAR,” IEEE Access, vol. 9, p. 1, 2021.

[45] D. Xue, N. Yang, X. Zhao, and Z. Wang, “Point-cloud map update for connected and autonomous vehicles based on vehicle infrastructure cooperation: framework and field experiments,” in Proceedings of the 2021 IEEE International Intelligent Transportation Systems Conference (ITSC), pp. 2062–2067, Indianapolis, IN, USA, September 2021.

[46] F. Abrate, B. Bona, M. Indri, S. Rosa, and F. Tibaldi, “Multi-robot map updating in dynamic environments,” Distributed Autonomous Robotic Systems, vol. 83, pp. 147–160, Springer, Berlin, Germany, 2012.

[47] M. Labbé and F. Michaud, “Long-term online multi-session graph-based SPLAM with memory management,” in Autonomous Robots, vol. 42, pp. 1133–1150, Springer US, no. 6, 2018.

[48] N. Shaik, T. Liebig, C. Kirsch, and H. Müller, “Dynamic map update of non-static facility logistics environment with a multi-robot system,” in Advances in Artificial Intelligence, G. Kern-Isberner, J. Fürnkranz, and M. Thimm, Eds., Lecture Notes in Computer Science, vol. 10505, pp. 249–261, Springer, Cham, 2017.

[49] K. Jo, S. Cho, C. Kim et al., “Cloud update of tiled evidential occupancy grid maps for the multi-vehicle mapping,” in Sensors, vol. 18, no. 12, p. 4119, Multidisciplinary Digital Publishing Institute, 2018.

[50] K. Jo, S. Lee, C. Kim, and M. Sunwoo, “Rapid motion segmentation of LiDAR point cloud based on a combination of probabilistic and evidential approaches for intelligent vehicles,” Sensors, vol. 19, no. 19, p. 4116, 2019.

[51] J. E. Bresenham, “Algorithm for computer control of a digital plotter,” IBM Systems Journal, vol. 4, no. 1, pp. 25–30, IBM, 1965.

[52] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, “A density-based algorithm for discovering clusters in large spatial databases with noise,” in Proceedings of the Second International Conference on Knowledge Discovery and Data Mining, vol. 96, no. 34, pp. 226–231, Portland Oregon, August 1996.

[53] M. Magnusson, H. Andreasson, A. Nuchter, and A. J. Lilienthal, “Appearance-based loop detection from 3D laser data using the normal distributions transform,” in Proceedings of the 2009 IEEE International Conference on Robotics and Automation (ICRA), pp. 23–28, Kobe, Japan, May 2009.

[54] T. F. Chan, "Gene Howard Golub and Randall J LeVeque. "Updating formulae and a pairwise algorithm for computing sample variances”" in Proceedings of the COMPSTAT 1982 5th Symposium held at Toulouse, pp. 30–41, Toulouse, Franch, 1982.

[55] S. Cho, C. Kim, M. Sunwoo, and K. Jo, “Robust localization in map changing environments based on hierarchical approach of sliding window optimization and filtering,” IEEE Transactions on Intelligent Transportation Systems, vol. 23, pp. 1–7, 2020.

[56] T. G. R. Reid, S. E. Houts, R. Cammarata et al., “Localization requirements for autonomous vehicles,” pp. 1–16, 2019, 10.4271/12-02-03-0012 https://arxiv.org/abs/1906.01061.