IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received December 16, 2020, accepted January 1, 2021, date of publication January 6, 2021, date of current version January 14, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3049482

Updating Point Cloud Layer of High Definition (HD) Map Based on Crowd-Sourcing of Multiple Vehicles Installed LiDAR

CHANSOO KIM[1], (Member, IEEE), SUNGJIN CHO[1], (Student Member, IEEE), MYOUNGHO SUNWOO[1], (Member, IEEE), PAULO RESENDE[2], (Member, IEEE), BENAZOUZ BRADAÏ[2], (Member, IEEE), AND KICHUN JO[3], (Member, IEEE)

[1]Department of Automotive Engineering, Hanyang University, Seoul 04763, South Korea
[2]Driving Assistance Research Center, Valeo, 93012 Bobigny Cedex, France
[3]Department of Smart Vehicle Engineering, Konkuk University, Seoul 05029, South Korea

Corresponding author: Kichun Jo (kichun.jo@gmail.com)

This work was financially supported by the BK21 plus program (22A20130000045) under the Ministry of Education, Republic of Korea, the Industrial Strategy Technology Development Program (No. 10039673, 10060068, 10079961), the International Collaborative Research and Development Program (N0001992) under the Ministry of Trade, Industry and Energy (MOTIE Korea), and National Research Foundation of Korea (NRF) grant funded by the Korean government (MEST) (No. 2011-0017495, 2019R1G1A1099806, 2020R1C1C1007739). This research was also funded and conducted under 'the Competency Development Program for Industry Specialist' of the Korean Ministry of Trade, Industry and Energy (MOTIE), operated by Korea Institute for Advancement of Technology (KIAT). (No. N0002431 and No. N0002428).

ABSTRACT A high-definition (HD) map is becoming an integral component of future mobility systems such as autonomous and connected vehicles. Advances in computing systems, LiDAR technologies, and vehicle communication technologies have enabled the HD map to directly treat a point cloud map (PCM), modeling road environments as LiDAR signal-level data. However, if actual road environments are changed, the PCM, modeling the environments before changes, can not be used for vehicle applications. Accordingly, the PCM has to stay up-to-date states by reflecting the environment changes continuously. This paper presents a crowd-sourcing framework to update the PCM from environment changes continuously using LiDAR and vehicle communication. Multiple intelligent vehicles installed with the LiDAR sensors download the PCM from a map server via wireless vehicle communication. To minimize the effects of environment changes, a robust localization based on a hierarchical Simultaneous Localization and Mapping (SLAM) estimates the pose (position and direction). The estimated pose is used to detect the differences between the PCM and environments, which are defined as map changes. The map changes are detected by the probabilistic and evidential theory considering the LiDAR characteristics, such as beam divergence and multi-echo. The detected map changes are uploaded to the map cloud server and merged into the PCM. The proposed crowd-sourcing framework to keep the PCM up-to-date is verified and evaluated via simulations and experiments in sites with road environment changes.

INDEX TERMS High-definition (HD) map, point cloud, crowd-sourcing, light detection and ranging (LiDAR), localization.

I. INTRODUCTION

A High-definition (HD) map is an essential component for the era of autonomous cars with more accurate and detailed data (10-20 cm accuracy) than current navigation maps [1]. The HD map is used for the key functions of autonomous cars, such as localization [2]–[5], perception [6], decision [7], and control. A lot of map companies (such as HERE [8], TomTom [9], and CARMERA [10]) are currently starting to provide the HD map service to keep pace with the innovation of autonomous cars. The HD map can be divided into two types: a landmark-level HD map and a signal-level HD map based on LiDAR. The landmark-level HD map includes readable landmarks such as road geometry [11], road surface marking

The associate editor coordinating the review of this manuscript and approving it for publication was Shaohua Wan.

[12], [13], and traffic signs [14]. On the other hand, the signal-level HD map based on LiDAR includes signal-level raw data such as point cloud map (PCM) [15]–[18] and occupancy grid map [19]–[22].

Among various map types of the HD map, the PCM, which is defined as a collection of 3d points reflected against environments, is widely used as a base map due to some advantages. First, the PCD map can be straightforwardly constructed by accumulating point clouds measured by LiDAR sensors. The PCM provides precise geometric shape information due to precise 3d point clouds measured by LiDAR sensors. Next, the PCM with the intensity can be easily converted to other map information such as road geometry, road surface marking, and traffic signs. The benefits of the PCM enables intelligent vehicles to localize precise poses (position and direction) by matching with the measured points [23], [24] and to plan the trajectory without any collisions [25].

However, there are common problems with the HD map including the PCM when actual road conditions are changed due to unexpected road construction and accidents. Since the map was built in advance by a special mapping vehicle installed with a mobile mapping system (MMS) with the LiDARs and a high-performance positioning system, the map, modeling the previous environments, is different from the present environments with the environment changes. The differences disturb the usage of the map for autonomous driving applications. Accordingly, the map has to stay up-to-date states by updating the differences caused by environment changes.

Previous studies used the mapping vehicle to update the environment changes to the map. However, the MMS-based map update process causes two problems. First, the tasks that the vehicles with MMS are continuously re-driven on all roads to acquire environment changes occur high maintenance costs such as the hiring of drivers and the running of MMS systems. Secondly, map update latency occurs until performing the map update process because it takes too much time for a few mapping vehicles with MMS to be driven on all roads.

To solve the previous problems such as costs and latency, methodologies based on crowd-sourced measurements, which are measured by affordable sensors installed in intelligent vehicles, are widely researched [26]–[30]. After the differences between the map and the environments can be detected through all measurement data uploaded from the intelligent vehicles to a map cloud server (managed by map maintainer), the detected differences update the existing map information in the map cloud server. The updated map in the map cloud server can be downloaded to the intelligent vehicles contributing to the map update framework and used for autonomous driving applications. The vehicles and maps are complementary in updating the map and helping the driving, respectively. Unfortunately, the crowd-sourced framework has mainly been researched for map update of the landmark-level HD map because relatively affordable camera sensors can measure the landmarks and data is small enough to transmit data by wireless network communications. In contrast, LiDAR measurements to update the PCM were not suitable as the crowd-sourced data because the point clouds were measured by too high prices of LiDAR sensors and were too bigger to transmit the data by wireless network communications than landmark measurements.

However, the recent trends for applying LiDAR sensors to mass-production and the advance of the wireless network communications draw out the potentials of the crowd-sourcing framework using LiDAR sensors. This paper presents a crowd-sourcing framework to update the PCM based on crowd-sourcing of multiple vehicles installed with LiDAR sensors. The crowd-sourcing cars download the PCM from the map cloud server via wireless network communications. The PCM is used to estimate the vehicle pose at the centimeter-level through the robust localization algorithm. Based on the estimated pose, the differences between the downloaded PCM and environments, which are defined as map changes, can be detected with consideration of the laser characteristics of LiDAR sensors (multi-echo and beam-divergence). The detected map changes are uploaded to the map cloud server and merged into the PCM based on an evidential approach. Similar to the crowd-sourcing landmark-map update system, the vehicles and the PCM have complementary relationships. The contributions of the proposed system are the following:

1) The crowd-sourcing algorithm applies both probabilistic and evidential theories to detect the map changes in the PCM, considering the LiDAR measuring characteristics (such as beam divergence and multi-echo).
2) A robust localization algorithm based on a hierarchical simultaneous localization and mapping (SLAM) approach estimates the crowd-sourcing vehicles' accurate pose in the partially changing environments.
3) Map changes detected by crowd-sourcing vehicles are merged back to the PCM through the proposed evidential approach.

This paper is organized as follows. Section II presents previous works related with the paper. Section III describes the overall architecture of the crowd-sourcing framework to update the PCM. Section IV presents the robust localization algorithm based on the hierarchical SLAM. The map change detection algorithm considering the LiDAR characteristic is presented in Section V. Section VI describes the evidential merging method of the detected map changes. Section VII and VIII show results from simulations and experiments to verify the proposed system, and the final section describes our conclusions.

II. RELATED WORKS

The HD map used for autonomous driving is classified as two map types: landmark-level and signal-level. The landmark-level map includes readable landmarks detected by the drivers such as road geometry, road surface marking, traffic signs, and traffic lights. Since the landmarks can be represented

as lightweight simple geometric models like points, polylines, polygons, and circles, map update based on crowd-sourced measurements have been widely researched. Kim *et al.* [31], [32] researched adding of new feature maps in the HD map based on crowd-sourced measurements; as a result, streetlights and traffic signs were added into the HD map as new features. Jo *et al.* [33] managed the traffic signs in the HD map by surveying whether traffic signs in the HD map were deleted or the new traffic signs were added based on the evidential approach. Pannen *et al.* [34] kept lane markings in the HD map up-to-date using the three pipelines (change detection, job creator, and map update). As well as academic fields, various companies (Daimler [35], NVIDIA [36], Bosch [37], TomTom [38], and HERE [39]) and consortia (SENSORIS and KoHAF) have researched the map update frameworks to keep the HD map up-to-date based on crowd-sourced data.

The signal-level map based on LiDAR measurements includes PCM [15]–[18] and occupancy grid map [19]–[22]. The signal-level map has been researched mainly related to the map change detection and update based on a standalone machine due to the high cost of the LiDAR and big data size of measurements. The changes of the PCM, modeling the environments by collecting points reflected from LiDAR, is detected by thresholding distances between the LiDAR points and the map points [40], [41]. In other approaches, the changes of the PCM can be evaluated and detected by using a ray-casting method based on the LiDAR laser beam [42]–[45]. Especially, an evidence theory is applied to relieve noises of the change detection from the ray-castings of the measurements [44], [45]. The detected changes can be updated into the existing map.

The occupancy grid map is one of the signal-level maps, which is composed of lots of occupancy grids splitting the World. The cell state of the occupancy grid map can be updated stochastically based on LiDAR ray casting [46]. The cells passed by the rays can be inferred to the free cells and the cells blocked by the LiDAR measurements can be inferred to the occupied cells. Especially, the Krajnik *et al.* provided the framework to update the occupancy grid map frequently [47], [48]. Unfortunately, since the cell state of the occupancy grid map is managed by the probabilistic theory, the map can not distinguish the difference between unknown cells (no measurements) and conflict cells (different measurements). To distinguish the difference, the evidence theory is applied to the occupancy grid map [49], [50].

On the other hand, crowd-sourced update systems for the signal-level map have been partially researched. For the occupancy grid map, the probabilistic theory-based map update system [51]–[53] and the evidence theory-based map update system [54] were researched. However, the previous studies for the map change detection and updating did not consider the entire crowd-sourcing framework and the system constraints such as uncertainty of localization and LiDAR's characteristics (multi-echo and beam-divergence).

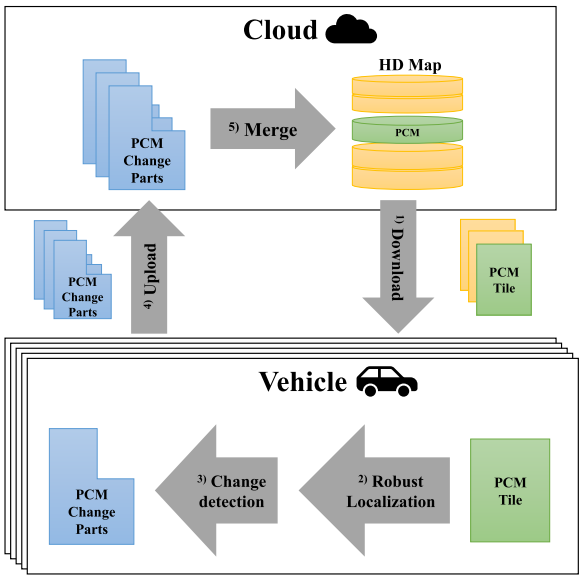

FIGURE 1. System architecture of the crowd-sourced PCM updating system.

III. SYSTEM ARCHITECTURE

Figure 1 shows the architecture of the proposed crowd-sourced PCM updating system. The system is divided into two parts: a cloud (server) for the map management and vehicles (clients) for the map change detection. The process of the crowd-sourced PCM update consists of five steps: 1) download, 2) robust localization, 3) change detection, 4) upload, and 5) merge.

1) Download: the high definition (HD) map is composed of several layers, such as the road topology layer, geometry layer, lane layer, and traffic sign layer, depending on what is included. The PCM layer is one of the layers in the HD map, which stores the point cloud information reflected against the road environments. The PCM consists of several tiles that divide the space into several areas. The vehicle can download the tiles that correspond to the region where the vehicle is currently located. The proposed algorithm is subject to quadtree tiling to provide a common tile structure that can be used around the world [54].

2) Robust localization: to recognize the environment changes through LiDAR, the vehicle's pose (position and orientation) on the PCM must be known. The vehicle pose can be estimated through a localization algorithm based on a geometric relationship between the real-time LiDAR point cloud and the PCM. The geometric relationship of two-point clouds can be obtained from registration techniques such as iterative closest points (ICP) and normal distribution transform (NDT). However, if partial environments are changed, it affects the localization's performance based on the point cloud registration due to the incorrect registration. Therefore, the proposed system needs a robust localization that can robustly estimate the position, even if there are partial environment changes. This paper proposes a robust localization

algorithm for robust pose estimation based on hierarchical SLAM and NDT matching.

3) Change detection: Based on the estimated pose from the robust localization, the LiDAR measurement is used to find map changes which mean the difference between the environments and the downloaded PCM. This paper proposes a change detection algorithm for recognizing map changes using the probability theory and the evidence theory based on the laser characteristics of LiDAR.

4) Upload: map changes detected in several cars are uploaded to the map cloud server. Since map changes includes only the changed point cloud part in the PCM (new and deleted points), data can be transmitted efficiently.

5) Merge: map changes detected in several cars are merged back to the PCM layer of the HD map based on the evidence theory. The merged PCM, which reflects environment changes, can be downloaded by other cars. If the cars revisit the environments with the previous map, the up-to-date map can be restored by downloading the PCM change parts only.

To implement the proposed crowd-sourced PCM updating system, the PCM layer of the HD map must be constructed in advance through a precise MMS with the LiDARs and a high-performance positioning system. Accurate positions of PCM's points in the geodetic coordinate systems cannot be obtained using sensors on the crowd-sourcing vehicles, such as low-cost GNSS data, on-board motion sensors, and LiDAR. Next, it is essential for the LiDAR points from the dynamic objects, such as moving vehicles and pedestrians, to be filtered out before localization and map change detection. In order to remove the dynamic points, the motion segmentation of LiDAR point algorithm is applied [55]. Based on preliminaries, the remaining of the paper will focus on steps 2, 3, and 5.

IV. ROBUST LOCALIZATION IN MAP CHANGING ENVIRONMENT

To detect the environment changes through LiDAR, we have to know the exact pose of LiDAR sensors. The LiDAR pose can be obtained from the vehicle pose and a geometric relationship between the vehicle frame and LiDAR frame (extrinsic calibration). The vehicle pose can be estimated through a localization algorithm based on the information fusion of vehicle motion, LiDAR measurement, and PCM.

A. GRAPH-BASED LOCALIZATION AND LIMITS

There are a lot of localization methods, such as Kalman filter (KF), extended KF, unscented KF, particle filter, and graph-based optimization. In the map update framework, the localization does not require real-time processing property but require better performance. Accordingly, we apply the graph-based optimization (Graph SLAM) as a basic localization framework because it is widely used for offline localization and mapping with good performance. The graph-based optimization consists of two-step processing: a front-end (construction of graph) and a back-end (graph optimization).

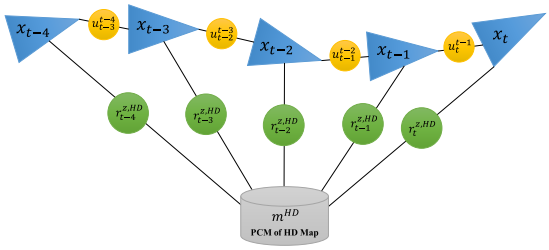

FIGURE 2. Graph structure for the localization problem.

1) FRONT-END

The first step of graph-based optimization is to construct the graph. The graph is composed of nodes (random variables) and edges (constraints between the nodes), as shown in Figure 2. Here, the nodes of $x_{1:t} = [x_1, \cdots, x_t]$ represent a sequence of the poses from the time steps 1 to t. The map node m^{HD} represents the static environment stored as a point cloud structure on the HD map (PCM layer).

The edges represent the geometric relationship (translation and rotation) between the nodes. The inputs of vehicle motion (such as the yaw rate and speed) for each time step can be represented as $u_{1:t} = [u_1, \cdots, u_t]$. Based on the vehicle motion model with the inputs $u_{1:t}$, the edge constraints $[u_2^1, \cdots, u_t^{t-1}]$ between the consecutive pose nodes can be constructed. The motion constraint can be represented by the following equation:

$$\begin{aligned} x_t &= h(u_t^{t-1}, x_{t-1}) + \epsilon_{u,t}, \\ F_{u,t} &= \epsilon_{u,t}^T P_{u,t}^{-1} \epsilon_{u,t}, \end{aligned} \tag{1}$$

where the $h(u_t^{t-1}, x_{t-1})$ represents the transformation function that transfers the previous pose x_{t-1} to the present pose x_t with the the edge constraint u_t^{t-1} (translation and rotation) and the error of the vehicle motion model $\epsilon_{u,t}$. The vehicle model error is represented as a Gaussian distribution with zero mean and covariance $P_{u,t}$. $F_{u,t}$ represents a negative log-likelihood of the motion prediction, which is represented by a quadratic form of the transition error $\epsilon_{u,t}$ and covariance $P_{u,t}$.

The edges of $[r_1^{z,HD}, \cdots, r_t^{z,HD}]$ represent the geometric relationship (translation and rotation) between the HD map PCM and LiDAR measurements. The geometric constraints can be obtained by registration techniques such as iterative closest points (ICP) and normal distribution transform (NDT). Based on the geometric relationship, a map matching constraint can be obtained as the following equation:

$$\begin{aligned} x_t &= h(r_t^{z,HD}, m^{HD}) + \epsilon_{r,t}, \\ F_{r,t} &= \epsilon_{r,t}^T P_{r,t}^{-1} \epsilon_{r,t}, \end{aligned} \tag{2}$$

where the $h(r_t^{z,HD}, m^{HD})$ represents the map matching model to estimate the pose x_t based on the HD map m^{HD} and the geometric constraint $r_t^{z,HD}$. The $\epsilon_{r,t}$ is the map matching error of the registration algorithm, and it can be modeled as a Gaussian distribution with zero mean and covariance $P_{r,t}$. $F_{r,t}$ represents a negative log-likelihood of the map matching, which is represented by a quadratic form of the transition error $\epsilon_{r,t}$ and covariance $P_{r,t}$.

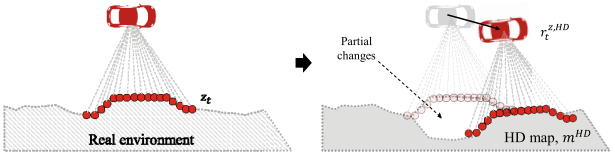

FIGURE 3. Incorrect matching issues due to partial environment changes.

2) BACK-END

A cost function of the graph-based optimization is obtained by sum of negative log-likelihood constraints (1) and (2), as described in equation (3). By minimizing the cost function, the entire poses $x_{1:t} = [x_1, \cdots, x_t]$ can be optimized. Here, we use a g2o library to minimize the cost function, which is c++ open source for the nonlinear least squares problems [56].

$$J = \sum_t \epsilon_{u,t}^T P_{u,t}^{-1} \epsilon_{u,t} + \sum_t \epsilon_{r,t}^T P_{r,t}^{-1} \epsilon_{r,t} \tag{3}$$

3) PROBLEMS WITH THE MAP MATCHING CONSTRAINTS IN ENVIRONMENT CHANGING SITUATIONS

The graph-based optimization algorithm provides the best pose estimation when the nodes and edges are well constructed. However, there is a problem with the map matching edge configuration when the road environment is partially changed. Figure 3 shows a situation where the partial map changes affect the incorrect edge construction of the map matching constraints. The registration algorithm estimates the geometric relationship $r_t^{z,HD}$ between the LiDAR measurement z_t of red points reflected by the road environment in Figure 3-(a)and the HD map m^{HD} of gray regions in Figure 3-(b). As shown in Figure 3-(b), if the partial environment changes are not considered in the HD map, the registration algorithm will provide the wrong geometric relationship between the measurement and the HD map; therefore, an incorrect map match constraint will be constructed in the graph inevitably. The incorrect map matching constraints in the graph can affect the entire optimization process. Accordingly, the proposed map change detection needs a robust localization that can estimate the pose robustly even if there is a partial change in the road environment.

B. ROBUST LOCALIZATION BASED ON HIERARCHICAL OPTIMIZATION

For robust localization, a hierarchical pose graph optimization based on the NDT matching is proposed, consisting of two hierarchical processes: 1) a sub-map optimization and 2) a representative pose optimization. Since both processes are based on graph optimization, the back-end processes are no different, but the front-end processes which construct the nodes and edges are different.

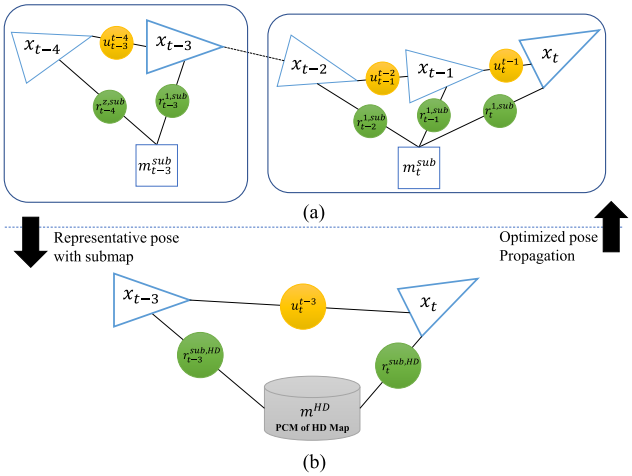

FIGURE 4. Hierarchical Graph SLAM for robust localization in environments with changes.

1) SUB-MAP OPTIMIZATION

The first step of the proposed robust localization is the sub-map optimization. For specific pose nodes within a specific distance window, the edges of the motion constraints and the sub-map geometric constraints are configured. An example of the sub-graph structure for the sub-map generation is shown in the Figure 4 (a). There are two sub-graph of the sub-maps m_{t-3}^{sub} and m_t^{sub}. For the sum-map m_t^{sub}, the pose nodes x_t, x_{t-1}, and x_{t-2} within a certain distance window are constrained by the motion constraints u_{t-1}^t and u_{t-2}^{t-1} and the geometric constrains $r_t^{z,sub}$, $r_{t-1}^{z,sub}$, and $r_{t-2}^{z,sub}$. The geometric constraints represents the geometric relationship between the sub-map m_t^{sub} and LiDAR measurement z_t, z_{t-1} and z_{t-2}. To form the geometric constraints (rotation and translation), a NDT matching technique is used. After forming the sub-graph for all poses $x_{1:t} = [x_1, \cdots, x_t]$, the graph optimization of the back-end process is applied to each sub-graph to obtain the sub-optimized poses and sub-maps.

2) REPRESENTATIVE POSE OPTIMIZATION

The second step of the robust localization is the representative pose optimization. This process also use the graph optimization, but the nodes and edges are different to the previous process. The graph structure for the representative pose optimization is shown in the Figure 4 (b). The nodes are consists of representative poses and the HD map m^{HD}. The representative poses x_t and x_{t-3} are obtained from the previous sub-map optimization. Each representative pose has the motion constraints u_t^{t-3} based on the vehicle motion model and sub-optimized pose relationship. The representative poses also has the geometric constraints $r_{t-3}^{sub,HD}$ and $r_t^{sub,HD}$ with the HD map m^{HD}. The geometric constrains $r_{t-3}^{sub,HD}$ and $r_t^{sub,HD}$ are constructed based on the NDT matching the sub-map m_{t-3}^{sub} and m_t^{sub} with the HD map m^{HD}. By optimizing higher level graph, we can obtain the globally optimized representative pose x_t and x_{t-3} with consideration on constraints of the

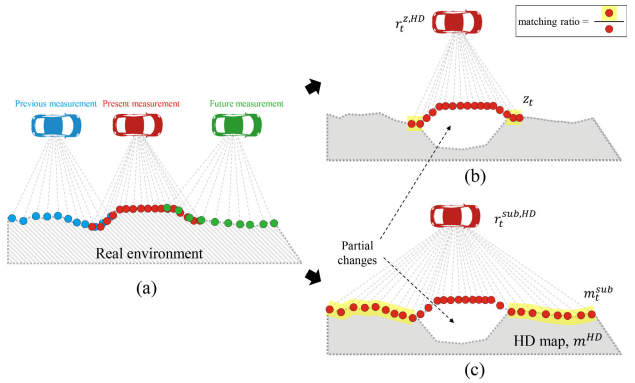

FIGURE 5. **(a) LiDAR measurements reflected from the road environment. (b) The single measurement constructs a matching constraint with the HD map, and (c) a sub-map with multiple measurements constructs a matching constraint with the sub-map in order to correct the pose of the sub-map nodes.**

HD map. The optimized representative poses x_t and x_{t-3} are propagated to the poses within the corresponding sub-graph.

3) ROBUST LOCALIZATION BASED ON NDT-BASED HIERARCHICAL GRAPH POSE OPTIMIZATION

Based on the two-step hierarchical graph optimization, we can obtain the optimized vehicle poses, which will be used for the map change detection. There are two reasons why the NDT registration-based hierarchical pose graph optimization is robust to the partial environment changes. First, the NDT algorithm is more robust under environment changes than other registration algorithms, as proved in the previous studies [57]–[60]. The NDT algorithm can reliably estimate the geometric relationship between source and target point clouds, even with partial changes in the target point cloud (HD map PCM layer). Second, a matching ratio between source and target point clouds is increased, as shown in Figure 5-(b) and (c). The matching ratio is the ratio of matched source points (red circles in the yellow thick line) in source points (red circles). As shown in Figure 5-(b), if the source points are formed from one or a few LiDAR scans, the matching ratio will be small in partial environment changes. The small matching ration can cause relatively inaccurate matching because the registration algorithm (NDT) does not have enough information to estimate the translation and rotation between the source and target point clouds. Conversely, if the sub-map grows, the matching ratio increases, as shown in Figure 5-(c). As the matching ratio increases, stable position estimation is possible; therefore, stable edges can be formed even with the partial environment change. The robust localization algorithm is summarized by a pseudo-code as shown in Algorithm 1.

Algorithm 1 Robust Localization

Input:
Motion information of the vehicle u_t
Point cloud from the vehicle z_t
Base HD map in cloud m^{HD}_{base}
Output:
Optimized vehicle pose $x_{opt,t}$

$m^{HD}_{veh} \leftarrow m^{HD}_{base}$
for all poses $x_t \in x_{window}$ **do**
  motion edge $\epsilon^T_{u,t} P^{-1}_{u,t} \epsilon_{u,t}$,
  where $\epsilon_{u,t} = x_t - h_u(x_{t-1}, u_t)$
  point cloud edge $\epsilon^T_{z,t} P^{-1}_{z,t} \epsilon_{z,t}$,
  where $\epsilon_{z,t} = x_t - h_z(x_{t-k}, z_{t-k}, z_t)$
  cost function J
  $\sum_{t \in W} \epsilon^T_{u,t} P^{-1}_{u,t} \epsilon_{u,t} + \sum_{t \in W} \epsilon^T_{z,t} P^{-1}_{z,t} \epsilon_{z,t}$
end for
optimize cost function J to find poses $x^*_{t-k:t}$
generate sub-map m^{sub}_t using $x^*_{t-k:t}$ and $z_{t-k:t}$
representative pose x_t
for all representative poses x^*_t **do**
  motion edge $\epsilon^{*\,T}_{u,t} P^{*\,-1}_{u,t} \epsilon^*_{u,t}$,
  where $\epsilon^*_{u,t} = x^*_t - h_u(x^*_{t-1}, u_t)$
  map matching edge $\epsilon^{*\,T}_{m,t} P^{*\,-1}_{m,t} \epsilon^*_{m,t}$,
  where $\epsilon^*_{m,t} = x^*_t - h_m(x^*_{t-1}, m^{sub}_t, z_t)$
  cost function J
  $\sum_t \epsilon^{*\,T}_{u,t} P^{*\,-1}_{u,t} \epsilon^*_{u,t} + \sum_t \epsilon^{*\,T}_{m,t} P^{*\,-1}_{m,t} \epsilon^*_{m,t}$
end for
optimize cost function J to find poses $x_{opt,t}$
return $x_{opt,t}$

V. MAP CHANGE DETECTION BASED ON PROBABILISTIC THEORY AND EVIDENTIAL THEORY

A. OVERVIEW OF MAP CHANGE DETECTION

When there are changes in the road environment, the PCM in the map cloud server must be updated to reflect the environment changes. To update the environment changes to the PCM, the changes must be detected in advance. Figure 6-(a) represents the road environment changes and PCM layer in the HD map. The road environment change is a situation in which an existing object is removed, and a new object is placed; however, the PCM in the HD map does not reflect the environment change. To reflect the changes, the proposed algorithm detects the map changes which are defined as the differences between the map and the environments. The map changes are composed of two parts: a deleted PCM part of the existing PCM and a newly PCM part added by the measured point cloud. The deleted PCM in the HD map and new PCM are uploaded to the cloud via a vehicle communication network, as shown in Figure 6-(b). There is not much communication traffic for uploading because only the changed parts are uploaded without all parts being uploaded.

B. MAP CHANGE DETECTION PROCESS BASED ON THE UPDATING OF PCM EXISTENCE STATE WITH LiDAR EXISTENCE FIELD

The process of the map change detection can be divided into two parts: the process when the PCM is in LiDAR field of view (FoV) and the process when the PCM is out of LiDAR FoV. Figure 7 shows the process of the PCM change detection based on the LiDAR installed on the vehicle.

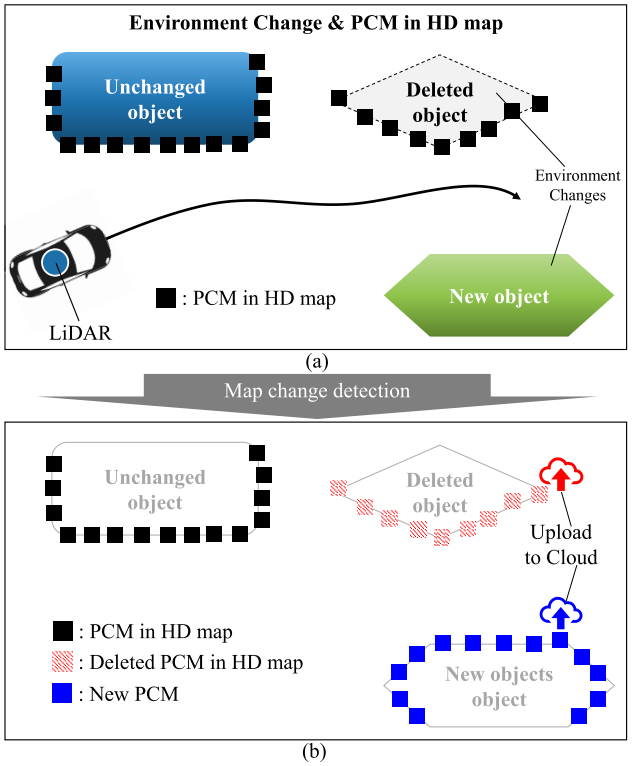

FIGURE 6. Overview of the map change detection. (a) road environment changes and PCM layer in an HD map, (b) map change detection, and uploading to the map cloud.

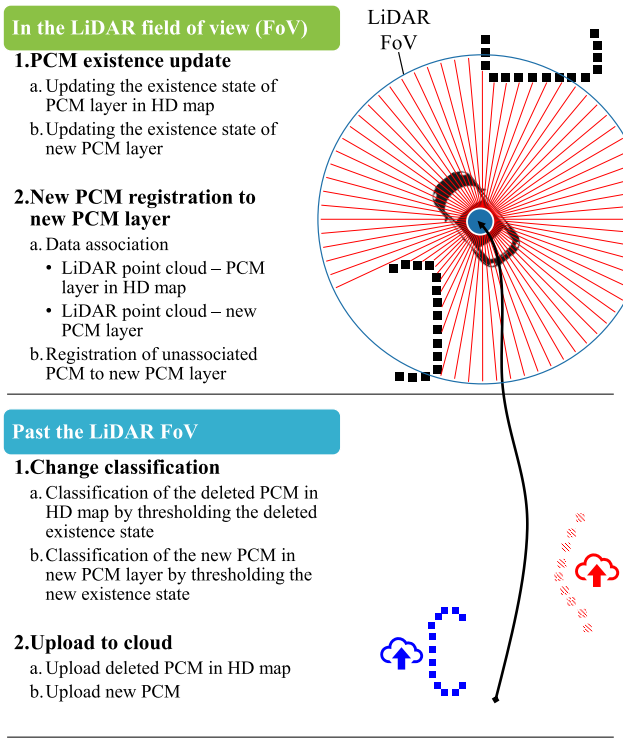

FIGURE 7. A two-step process of PCM change detection based on the LiDAR installed on the vehicle.

1) IN THE LiDAR FoV

The first step process in the LiDAR FoV is an existence update of points in the two PCM layers. As shown in Figure 6, there are two main PCM layers used to update map changes: HD map's PCM layer and new PCM layer to represent the new object. Each point in the PCM layers has a *state* that indicates the level of existence, which consists of *existed, deleted, unknown*. The elements of the *existence* state of each point have a *belief* between 0 and 1 by evidence theory. Theoretical descriptions of the existence state by evidence theory are described in the following subsection V-C.

The *existence* state of points in the PCM layer can be updated by the *existence field* generated based on the LiDAR measurement and the pose estimation by the previous robust localization. Figure 8 conceptually illustrates the formation of the existence field and the update of the existence state of the points in the PCM. The red dot represents 3D points of LiDAR measurement at time t, and it is represented by $z_{time}^{index,order}$. The black and blue squares represent the points in the PCM layers (HD map's PCM layer and new PCM layer), respectively, and the representation of each point of the PCM layers is p_{source}^{index}. A fan-shaped existence field can be formed using the laser beam characteristics (such as beam divergence and multi-echo) of LiDAR measurement centered on the LiDAR pose estimated from the robust localization. The green area in the existence field is the area where the laser has passed. When points (such as p_{HD}^{1} and p_{new}^{1} in the Figure 8) of PCM layers exists in this area, the *deleted belief* in the existence state increases and the *existed belief* and *unknown belief* decreases. Conversely, when points (e.g. p_{HD}^{3}, p_{HD}^{4} and p_{new}^{2}) of the PCM layers are in the red zone, the *existed belief* increases and the *deleted belief* and *unknown belief* decreases. If the points (e.g. p_{HD}^{2}) of the PCM layers are not in the existence field, it maintains the previous existence state. Theoretical details of existence state update based on the existence field will be described in the following subsections V-C.

The second step process in the LiDAR FoV is a new PCM registration to the new PCM Layer. When LiDAR detects points that are not registered in the two PCM layers, the points should be newly registered in the new PCM layer. The association between the measured LiDAR points and the points in both PCM layers is performed to find new points that are not registered in both PCM layers. The associations are formed when the nearest neighbor PCM points for the measured LiDAR points are within a certain threshold. When the measured LiDAR point is not associated with the points in both the PCM layer, the point is determined as a new point map and registered in the new PCM layer. In Figure 8, the LiDAR measurements z_t^1 and z_t^2 are not associated with points in both PCM layers. They are registered to the new PCM layer with p_{new}^{3} and p_{new}^{4}.

2) PAST THE LiDAR FoV

When the PCM layer points leave the LiDAR FoV, the existence states of those points are no longer updated and fixed. The *change (deleted or new)* of points in each PCM layer

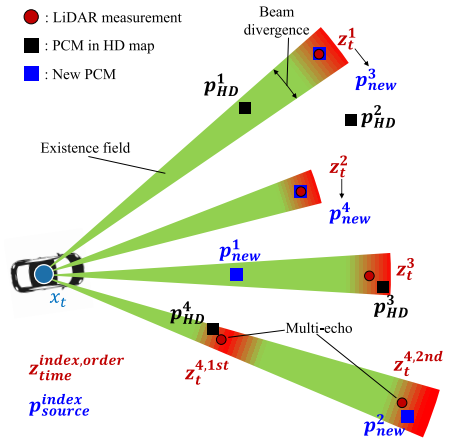

FIGURE 8. Formation of the existence field and the update of the *existence* state of the PCM points.

can be classified by the amount of the element value in each point's existence state. If the *deleted* value for the existence state of a point in the PCM layer of the HD map is higher than a certain existence threshold $th_{deleted}$, the point is classified as *deleted*. If the *new* element for the existence state of a point in the New PCM layer is higher than the threshold th_{new}, it is classified that the points have changed to *new*. The threshold value means the minimum confidence belief to judge the changes (deleted or new) and is determined by tuning according to localization and LiDAR characteristics. The HD map's points classified as deleted and the new PCM layer's points classified as new are uploaded to the map cloud server.

C. EXISTENCE FIELD MODELING AND EVIDENTIAL UPDATE OF PCM BASED ON PROBABILISTIC THEORY AND EVIDENTIAL THEORY

The existence state of each PCM point is updated through the existence field. This subsection explains how to model the existence field and how to update the existence state of each PCM point based on the combination of the probabilistic approach and the evidential approach.

1) LiDAR CHARACTERISTICS AND PCM LAYERS TO BE UPDATED

LiDAR can measure the distances and directions of nearby objects using a laser. A laser pulses are emitted at a certain angle and the time-of-flight (ToF) principle can be used to measure the distance to the object for that angle. The laser has a characteristic of *beam divergence*, which increases the cross-sectional area of the beam with distance. The *beam divergence* enables a *multi-echo* on an emitted laser pulse, simultaneously measuring multiple distances of objects of different distances. The final characteristic of LiDAR is a *uncertainty* of the distance measurement. The uncertainty of angular measurement is negligible because the LiDAR controls the laser pulse angle by an internal encoder. However, the uncertainty of distance measurement is determined by the measuring capability of the laser pulse's ToF and cannot be ignored. Many previous studies did not consider characteristics of LiDAR and treated a LiDAR measurement as single 3D point with no volume and constant 3D uncertainty. The proposed algorithm reflects these three characteristics of LiDAR (*beam divergence*, *multi-echo* and *distance uncertainty*) to update PCM point existence accurately and reliably.

The LiDAR measurement can be represented into two types of coordinate system: spherical coordinate and Cartesian coordinate. Representation of LiDAR measurement in spherical coordinates is $Z_{r\theta\phi,t} = \{z^{1,m}_{r\theta\phi,t}, \cdots, z^{i,m}_{r\theta\phi,t}, \cdots, z^{N,m}_{r\theta\phi,t}\}$. The one point $z^{i,m}_{r\theta\phi,t}$ is $\{r^{i,1}_t, \cdots, r^{i,m}_t, \theta^i_t, \phi^i_t\}$, where r is distance to a object, i is index of laser pulse, m is number of echos, θ is the vertical (polar) angle, ϕ is the horizontal (azimuthal) angle, and t is measurement time. The Cartesian coordinate representation is $Z_{xyz,t} = \{z^{1,m}_{xyz,t}, \cdots, z^{i,m}_{xyz,t}, \cdots, z^{N,m}_{xyz,t}\}$, where $z^{i,m}_{xyz,t} = \{x^{i,m}_t, y^{i,m}_t, z^{i,m}_t\}$. To avoid confusion, we will describe the LiDAR measurement as $Z_t = \{z^{1,m}_t, z^{2,m}_t, \cdots, z^{N,m}_t\}$ without separating the two coordinate systems.

There are two PCM layers to be updated by the proposed map change update algorithm: HD map's PCM layer and new PCM layer. The HD map's PCM layer is represented by P_{HD} and contains multiple points p^i_{HD}. The new PCM layer is represented by P_{new} and contains points p^i_{new}. In summary, both PCM layers are represented by $P_{HD} = \{p^1_{HD}, \cdots, p^D_{HD}\}$ and $P_{new} = \{p^1_{new}, \cdots, p^W_{new}\}$, where D and W are the number of points in the PCM layers of HD map and the new map, respectively. Since the proposed map change update process is applied equally to both P_{HD} and P_{new}, the two PCM layers are combined and represented as $P = \{P_{HD}, P_{new}\}$. The point elements of the combined PCM layer P is represented as $P = \{p^1_{HD}, \cdots, p^D_{HD}, p^{D+1}_{new}, \cdots, p^{D+W}_{new}\} = \{p^1, \cdots, p^i, \cdots, p^{D+W}\}$.

The *existence* of each point $\{p^1, \cdots, p^i, \cdots, p^{D+W}\}$ in the combined PCM P at time t can be represented as $E_t = \{e^1_t, \cdots, e^i_t, \cdots, e^{D+W}_t\}$. The e^j_t element has three state of PCM point existence $\{existed, deleted, unknown\}$. The first step process of the PCM existence update algorithm (in the LiDAR FoV) updates the E_t using two types of information: LiDAR measurement $Z_t = \{z^{1,m}_t, z^{2,m}_t, \cdots, z^{N,m}_t\}$ and LiDAR sensor pose x_t. The LiDAR sensor pose is estimated by robust localization, described in the previous chapter IV.

2) PROBABILISTIC MODELING FOR LiDAR POINT MOTION

Using the probabilistic approach, the existence E_t of the two *PCM* layers at time t can be modeled for the given measurements of the LiDAR sensor pose x_t and the LiDAR measurement Z_t. The probabilistic belief model of the existence $bel(E_t)$ at time t is represented by a conditional probability for the given conditions, LiDAR sensor pose x_t and LiDAR measurement Z_t, as described in equation (4).

$$bel(E_t) = p(E_t|P, Z_t, x_t) \tag{4}$$

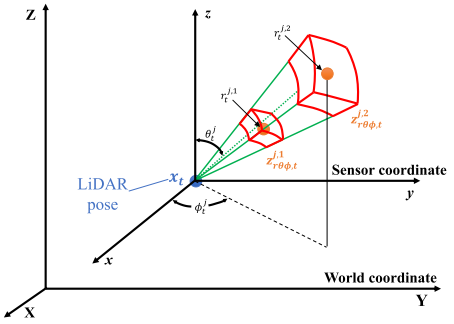

FIGURE 9. Existence field for the PCM point p^i is constructed by LiDAR point $z^{j,l}_{r\theta\phi,t}$ with consideration of the beam divergence and multi-echo.

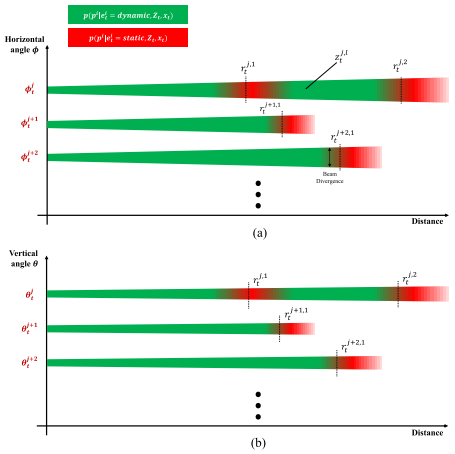

FIGURE 10. 2D existence field for the distance (r) and angle (ϕ and θ) of the LiDAR measurements $z^{j,1}_t$, $z^{j,2}_t$, $z^{j+1,1}_t$ and $z^{j+2,1}_t$.

Since the E_t is composed of the independent point existence $\{e^1_t, \cdots, e^j_t, \cdots, e^{D+W}_t\}$ for each point $\{p^1, \cdots, p^j, \cdots, p^{D+W}\}$ in the *PCM* layer, the $bel(E_t)$ can be represented by the set of conditional probabilities of each PCM point, as described in equation (5).

$$\begin{aligned} bel(E_t) &= \{bel(e^1_t), \cdots, bel(e^{D+W}_t)\} \\ &= \{p(e^1_t|p^1, Z_t, x_t), \cdots, p(e^{D+W}_t|p^{D+W}, Z_t, x_t)\} \end{aligned} \quad (5)$$

However, the probabilistic model only represents the existence state e^j_t as two states $\{existed, deleted\}$ because the probabilistic theory cannot handle the third state *unknown*. The third state *unknown* will be handled in the evidential update process. In the probabilistic process, the sum of $p(existed)$ and $p(deleted)$ is always one.

The conditional probability of one PCM point existence can be rearranged by the Bayes rule, as described in equation (6).

$$bel(e^i_t) = p(e^i_t|p^i, Z_t, x_t) = \frac{p(p^i|e^i_t, Z_t, x_t)p(e^i_t|Z_t, x_t)}{p(p^i|Z_t, x_t)}, \quad (6)$$

where $p(p^i|e^i_t, Z_t, x_t)$ is a likelihood of the PCM point for the given existence e^i_t, LiDAR measurement Z_t and LiDAR pose x_t. $p(e^i_t|Z_t, x_t)$ is a predicted probability of the PCM point existence e^i_t for the given Z_t and x_t. The $p(e^i_t|Z_t, x_t)$ is a uniform distribution ($p(e^i_t = existed|Z_t, x_t) = 0.5$ and $p(e^i_t = deleted|Z_t, x_t) = 0.5$) because the existence itself is independent to the Z_t and x_t. $p(p^i|Z_t, x_t)$ is represented as a normalization factor by applying total probability theorem, as described by equation (7).

$$p(p^i|Z_t, x_t) = \sum_{existence} p(p^i|e^i_t, Z_t, x_t)p(e^i_t|Z_t, x_t) = \eta \quad (7)$$

Since the $p(e^i_t|Z_t, x_t)$ is a uniform distribution and $p(p^i|Z_t, x_t)$ is a normalization factor η, the probability of the PCM point existence is represented by equation (8).

$$bel(e^i_t) = p(e^i_t|p^i, Z_t, x_t) = \eta\, p(p^i|e^i_t, Z_t, x_t) \quad (8)$$

From equation (8), we can figure out that the conditional probability of one PCM point existence can be expressed by the likelihood of the PCM point existence with a normalization factor. Therefore, the problem of PCM point existence probabilistic modeling $bel(e^i_t)$ is changed to a problem of PCM point likelihood estimation $p(p^i|e^i_t, Z_t, x_t)$ for the given existence e^i_t, LiDAR measurement Z_t and LiDAR pose x_t.

3) EXISTENCE FIELD CONSTRUCTION FROM THE LIKELIHOOD OF LiDAR MEASUREMENT

The existence state of PCM points are updated by an existence field as shown in the Figure 8. The existence field of a PCM point p^i can be constructed from the likelihood $p(p^i|e^i_t, Z_t, x_t)$. Figure 9 shows intuitively an existence field (likelihood field) of $p(p^i|e^i_t, z^{j,l}_t, x_t)$, where $z^{j,l}_t$ $(= z^{j,l}_{r\theta\phi,t})$ is LiDAR points in point cloud Z_t measured at pose x_t at the time t. The existence field for the PCM point p^i is constructed in the form of a triangular-pyramid by each LiDAR point $z^{j,l}_{r\theta\phi,t}$ of spherical coordinates with taking in to account the LiDAR characteristics of the beam divergence and multi-echo. The point p^i in the green region of the existence field is likely to be *deleted*, and the point p^i in the red region is likely to be *existed*.

The 3D existence field for the likelihood $p(p^i|e^i_t, z^{j,l}_t, x_t)$ can be divided into two 2D existence fields for the angular axis: distance-horizontal angle $(r - \phi)$ plane and distance-vertical angle $(r - \theta)$ plane. Figure 10 is the two 2D existence field of the $(r - \phi)$ and $(r - \theta)$ planes for the LiDAR measurements $z^{j,1}_t$, $z^{j,2}_t$, $z^{j+1,1}_t$ and $z^{j+2,1}_t$. The color intensity means the size of the likelihood. The existence field considers the LiDAR characteristics of the beam divergence and the multi-echo.

One laser beam $z^{j,l}_t$ in Figure 10 can be represented in the distance-likelihood plane, as shown in Figure 11. Figure 11-(a) shows the likelihood $p(p^i|e^j_t = existed, z^{j,l}_t, x_t)$ when a PCM point p^i is *existed* for given $z^{j,l}_t$ and x_t. The PCM point p^i and the LiDAR measurement $z^{j,l}_t$ can be represented in spherical coordinates as $\{r^i, \phi^i\}$ and $\{r^{j,1}_t, r^{j,2}_t, \phi^j_t\}$, respectively. Since the LiDAR measures the object distance based on ToF principle, the uncertainty of the distance measurement $r^{j,m}_t$

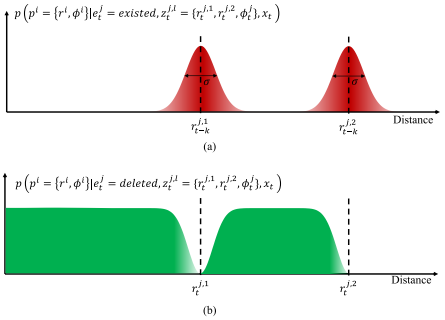

FIGURE 11. Likelihood of *existed* (red) and *deleted* (green) for one laser beam.

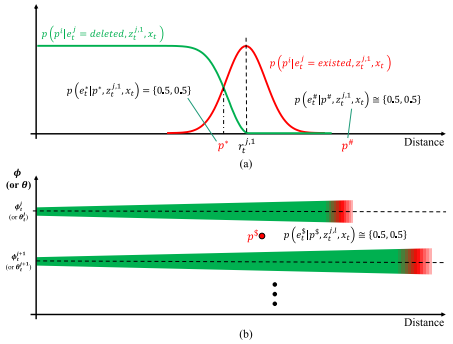

FIGURE 12. Problems of probabilistic motion segmentation.

depends on the performance of the ToF system. By taking into account the LiDAR distance uncertainty, the likelihood of *existed* existence is represented as a Gaussian normal distribution, as shown in Figure 11-(a). The Gaussian normal distribution of the distance measurement is represented by equation (9)

$$\begin{aligned} &p(p^i|e_t^j = existed, z_t^{j,l}, x_t) \\ &\quad = p(r^i|e_t^j = existed, r_t^{j,l}, x_t) = \frac{1}{\sigma\sqrt{2\pi}} e^{-(r-r^i)^2/2\sigma^2}, \end{aligned} \tag{9}$$

where σ is the standard deviation of the LiDAR distance measurement.

Figure 11-(b) shows the likelihood $p(p^i|e_t^j = deleted, z_t^{j,l}, x_t)$ when a PCM point p^i is *deleted* for given $z_t^{j,l}$ and x_t. Since the area where the LiDAR beam $z_t^{j,l}$ passed is likely to be free, the PCM point p^i located in the LiDAR beam passed region is more likely to be deleted. The likelihood $p(p^i|e_t^j = deleted, z_t^{j,l}, x_t)$ can be calculated based on its relationship with the likelihood $p(p^i|e_t^j = existed, z_t^{j,l}, x_t)$, as described in equation (10).

$$\begin{aligned} &p(p^i|e_t^j = deleted, z_t^{j,l}, x_t) \\ &= p(r^i|e_t^j = deleted, r_t^{j,l}, x_t) \\ &= \begin{cases} \dfrac{1}{\sigma\sqrt{2\pi}} - p(r^i|e_t^j = existed, z_t^{j,l}, x_t) & r <= r_t^{j,max(m)} \\ 0 & else \end{cases} \end{aligned} \tag{10}$$

$\frac{1}{\sigma\sqrt{2\pi}}$ denotes the maximum likelihood value of $p(r^i|e_t^j = existed, r_t^{j,l}, x_t)$.

4) EVIDENTIAL UPDATE OF PCM POINT EXISTENCE STATE

If the PCM point p^i is located in the existence field, the existence probability of the p^i can be obtained through the likelihood equations (9, 10) and the posterior probability equation (8). However, this approach has limitations that the probability theory cannot explicitly handle the *unknown* state. Figure 12 shows limitations of the probabilistic approach for updating the PCM point existence. A PCM point p^* is located inside the existence field and the point existence probability $p(e_t^*|p^*, z_t^j, x_t)$ for *existed* and *deleted* are the same. The probability of $\{existed, deleted\} = \{0.5, 0.5\}$ cannot explicitly represents *unknown* state. $p^\#$ in Figure 12-(a) and $p^\$$ in (b) are not located inside the existence field. Therefore, the likelihoods probability for each point is close to {0.5, 0.5} because both likelihoods for *existed* and *deleted* are close to zero. To represents the *unknown* state of the p_t^*, $p_t^\#$, and $p_t^\$$, an evidential approach is applied to update the existence of PCM points.

The probabilistic existence update process handles the existence in the two states $\{existed, deleted\}$. The evidence theory defines the two states as a frame of discernment $\Omega = \{existed, deleted\}$ and manages more states by extending the Ω to the power set $2^\Omega = \{existed, deleted, \Omega, \phi\}$. The state Ω represents an *unknown* state because the PCM point existence cannot be *existed* and *deleted* in the same time. The state ϕ is an empty set, and it is a *conflict* state because the point existence must belong to one state that *existed* or *deleted*. A mass m quantifies the belief of the power set $2^\Omega = \{existed, deleted, unknown, conflict\}$. The mass of $m^i(static)$, $m^i(dynamic)$, $m^i(unknown)$ and $m^i(conflict)$ represent the belief of PCM point p^i being *existed*, *deleted*, *unknown* and *conflict*, respectively. The sum of mass for each state must be *one*.

The *mass* of PCM point p^i existence for the given $z_t^{j,l}$ and x_t is represented by $m_t^{j,l\to i}(state)$ for $state = \{existed, deleted, unknown, conflict\}$. $m_t^{j,l\to i}(state)$ is obtained based on the existence probability $p(e_t^i|p^i, z_t^{j,l}, x_t)$ and the localization confidence λ_{loc} by using the following equations:

$$\begin{aligned} m_t^{j,l\to i}(existed) &= \lambda_{loc} p(e_t^i = existed|p^i, z_t^{j,l}, x_t) \\ m_t^{j,l\to i}(deleted) &= \lambda_{loc} p(e_t^i = deleted|p^i, z_t^{j,l}, x_t) \\ m_t^{j,l\to i}(unknown) &= 1 - m_t^{j,l\to i}(existed) - m_t^{j,l\to i}(deleted) \\ m_t^{j,l\to i}(conflict) &= 0 \end{aligned} \tag{11}$$

The λ_{loc} is tuning factor determined based on the performance of the robust localization. If the localization is very accurate and stable, the value is close to one; however, if it is not good, it is close to zero.

For the given LiDAR measurement $z_t^{j,l}$ and the given poses x_t, the existence of PCM point of p^i can be described by mass function $m_t^{j,l\to i}$. For all given scan points $Z_t = \{z_t^{1,l}, \cdots, z_t^{j,l}, \cdots, z_t^{N,l}\}$ and the given poses x_t, multiple mass functions $m_t^{1,l\to i} \cdots m_t^{N,l\to i}$ are calculated. The mass functions must be merged into a mass function m_t^i at time t. Then, the mass function $m_{1,\cdots,t}^i$ at the various time must be integrated into one mass function m^i to represent the existence of one point p^i. The integration of multiple mass functions can be described by the following equation (12).

$$\begin{aligned} m_t^i &= m_t^{1,l\to i} \oplus m_t^{2,l\to i} \oplus \cdots \oplus m_t^{N,l\to i} \\ m^i &= m_1^i \oplus m_2^i \oplus \cdots \oplus m_t^i \end{aligned} \tag{12}$$

Here, Dempster's combination rule (13), which is based on the conjunctive combination rule described by equation (14), is applied to integrate two different mass.

$$\begin{aligned} &m_1 \oplus m_2 \Longrightarrow \\ &m_{1\oplus 2}(A) = \frac{m_{1\cap 2}(A)}{1 - m_{1\cap 2}(\phi)}, m_{1\oplus 2}(\phi) = 0, \\ &\forall A \subseteq \Omega, A \neq \phi \end{aligned} \tag{13}$$

$$m_{1\cap 2}(A) = \sum_{B\cap C = A | B, C \subseteq \Omega} m_1(B) \cdot m_2(B) \tag{14}$$

Existence of one PCM point p^i can be described by he integrated mass function m^i. Since the updated PCM layer $P = \{p^1, \cdots, p^i, \cdots, p^{D+W}\}$ is composed of two PCM layers $P = \{P_{HD}, P_{new}\} = \{p_{HD}^1, \cdots, p_{HD}^D, p_{new}^{D+1}, \cdots, p_{new}^{D+W}\}$, the mass function of the PCM can be represented by $M = \{M_{HD}, M_{new}\} = \{m_{HD}^1, \cdots, m_{HD}^D, m_{new}^{D+1}, \cdots, m_{new}^{D+W}\}$. The proposed map change detection process is summarized as shown in Algorithm 2.

VI. INTEGRATION OF CROWD-SOURCED PCM CHANGE

The masses of all points, $M = \{M_{HD}, M_{new}\} = \{m_{HD}^1, \cdots, m_{HD}^D, m_{new}^{D+1}, \cdots, m_{new}^{D+W}\}$, are updated by the evidential updating process based on the probabilistic existence field. Changed points in HD map can be classified by thresholding the updated mass value. If the mass of deleted $m_{HD}^i(deleted)$ for the PCM point in HD map is larger than $th_{deleted}$, the PCM point p_{HD}^i is classified into *deleted*. If the mass of *existed* for the PCM point in new map, $m_{new}^j(existed)$, is larger than th_{new}, the PCM point p_{new}^j is classified into *new*. The PCM change classification is performed in multiple crowd-sourced cars, and the points classified as *deleted* or *new* are uploaded to the map cloud. Only changed points are uploaded so that the data transferring in the map update system is efficient.

At the cloud side, the changed points are uploaded from crowd-sourced vehicles. The changed point cloud is composed of the *deleted* point cloud $P_{cs,deleted} = \{p_{cs,deleted}^1, \cdots, p_{cs,deleted}^i, \cdots, p_{cs,deleted}^{CD}\}$ and the *new* point cloud $P_{cs,new} = \{p_{cs,new}^1, \cdots, p_{cs,new}^i, \cdots, p_{cs,new}^{CN}\}$. These updated point clouds are merged back to the PCM layer of the HD map based on crowd-sourced map integration process.

Algorithm 2 Map Change Detection

Input:
Optimized vehicle pose $x_{opt,t}$
Point cloud from the vehicle z_t
Base HD map in cloud m_{base}^{HD}
Output:
Changed point cloud $P_{cs,delete}, P_{cs,new}$

points in the based HD map $P_{HD} \leftarrow m_{base}^{HD}$ around x_t^*
for all poses $x_{opt,t}$ **do**
  for all points $z_t^i \in z_t$ **do**
    register_flag_hd_map $\leftarrow$ *TRUE*
    ray casting based on z_t^i in P_{HD} domain
    for all map points $p_{HD}^k \in P_{HD}$ **do**
      if p_{HD}^k is near z_t^i **then**
        existence evidence update of p_{HD}^k
        register_flag_hd_map $\leftarrow$ *FALSE*
      else
        if exist point in P_{HD} in the ray **then**
          deleted evidence update of p_{HD}^k
        end if
      end if
    end for
    register_flag_new_map $\leftarrow$ *TRUE*
    ray casting based on z_t^i in P_{new} domain
    for all map points $p_{new}^k \in P_{new}$ **do**
      if p_{new}^k is near z_t^i **then**
        existence evidence update of p_{new}^k
        register_flag_new_map $\leftarrow$ *FALSE*
      else
        if exist point in P_{new} in the ray **then**
          deleted evidence update of p_{new}^k
        end if
      end if
    end for
    if *register_flag_hd_map* == *TRUE* **and** *register_flag_new_map* == *TRUE* **then**
      register point z_t^i into P_{new}
    end if
  end for
end for
thresholding *deleted* evidence in P_{HD}
thresholding *existence* evidence in P_{new}
$P_{cs,delete} \leftarrow P_{HD,threshold}$
$P_{cs,new} \leftarrow P_{new,threshold}$
return $P_{cs,delete}, P_{cs,new}$

The integration process consists of three steps: 1) voxelization, 2) existence update, and 3) confirmation.

The first step is voxelization of the point cloud in the HD map P_{HD}. The points of P_{HD} are placed in evenly divided 3D voxels by the voxelization. If each voxel contains more than one point, the points are merged into one representative point by averaging the 3D position of the points (VoxelGrid filter). The same voxelization process is applied to

the crowd-sourced deleted points $P_{cs,deleted}$ and the crowd-sourced new points $P_{cs,new}$, which are uploaded from one vehicle. Figure 13 shows the voxelization of P_{HD} (blue), $P_{cs,new}$ (orange) and $P_{cs,deleted}$ (green).

The second step is the existence update of the voxels. Each voxel has each existence mass $\{m^1_{voxel}, \cdots, m^i_{voxel}, \cdots, m^N_{voxel}\}$, where state of mass represents $\{existed, deleted, unknown, conflict\}$. The initial value of each voxel mass m^i_{voxel} is determined by whether the voxel contains the PCM points of P_{HD}. If the voxel contains the point of P_{HD}, the existence mass is initialized as $\{1, 0, 0, 0\}$ which means the point exists in the voxel. If the voxel does not contain the point of P_{HD}, the existence mass is initialized as $\{0, 1, 0, 0\}$ which means the voxel is empty. The $P_{cs,deleted}$ and $P_{cs,new}$ are used to update the existence of voxel. The existence mass of voxel contained the $P_{cs,deleted}$ is updated by the following equation

$$m^i_{voxel} = m^i_{voxel} \oplus \{0, \lambda^{cs}_{deleted}, 1 - \lambda^{cs}_{deleted}, 0\}, \quad (15)$$

where $\lambda^{cs}_{deleted}$ represents the confidence of the crowd-source change classification and $\oplus$ is an operation of Dempster's combination (13). In a similar way, the existence mass of voxel contained the $P_{cs,new}$ is updated by

$$m^i_{voxel} = m^i_{voxel} \oplus \{\lambda^{cs}_{new}, 0, 1 - \lambda^{cs}_{new}, 0\}, \quad (16)$$

where λ^{cs}_{new} represents the confidence of the crowd-source new mapping. When the voxel is updated for the first time, the points are placed in the voxel as representative points. Otherwise, the representative points in the voxels are averaged with the new points and updated.

The final step is a confirmation of reflecting the updated voxel existence to the PCL layer of HD map. If the mass of *deleted* for the voxel, $m^i_{voxel}(deleted)$, is larger than $th_{cs,deleted}$, the PCM point in that voxel is deleted from the HD map PCL layer. If the mass of *existed* for the voxel, $m^i_{voxel}(existed)$, is larger than $th_{cs,new}$, the representative PCM point in the voxel is added to the HD map. The integration step is summarized in Algorithm 3.

Algorithm 3 Integration of Crowd-Sourced PCM

Input:
Changed point cloud $P_{cs,delete}$, $P_{cs,new}$
Base HD map in cloud m^{HD}_{base}
Output:
Updated HD map in cloud $m^{HD}_{updated}$
merged point cloud P_{merge}
$\{P^{HD}_{base} \in m^{HD}_{base}, P_{cs,delete}, P_{cs,new}\}$
voxelization
for all voxels **do**
  Dempster-Shafer rule-based evidence update
  thresholding *deleted*, *existence* evidence
end for
delete and add point in P_{merge}
$m^{HD}_{updated} \leftarrow P_{merge}$
return $m^{HD}_{updated}$

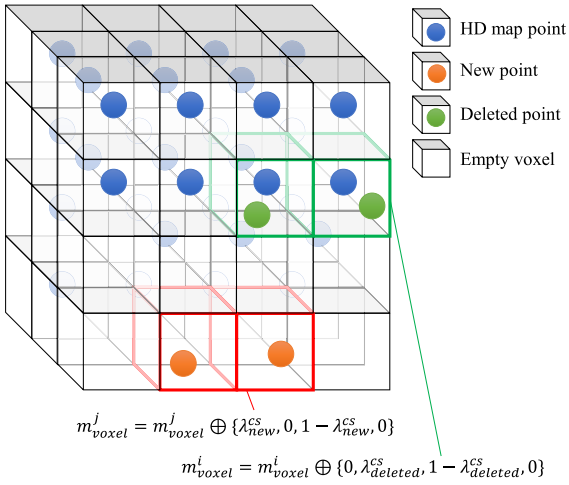

FIGURE 13. Voxelization of the point cloud in the HD map P_{HD}, the crowd-sourced deleted points $P_{cs,deleted}$, and the crowd-sourced new points $P_{cs,new}$. Each existence mass of each voxel is updated by $P_{cs,deleted}$ and $P_{cs,new}$.

VII. SIMULATIONS

A. SIMULATION ENVIRONMENTS

Although the map update framework should be evaluated under various environments, it is difficult to find large map changing environments like construction sites in practice. Accordingly, the proposed framework was first evaluated in the simulation process to generate map-changing environments. The simulation used a test vehicle equipped with several sensors such as an U-blox low-cost GNSS (2.5 meters position accuracy and 4Hz sampling rate), on-board motion sensors (automotive-level yaw rate and wheel speed sensors), and a high-precision OXTS RT3002 GNSS/INS (0.01-meter position accuracy, 0.05 km/h speed accuracy, 0.1-degree heading accuracy, 0.03-degree roll and pitch accuracy, and 250Hz sampling rate). The GNSS and motion information measured by the low-cost GNSS and on-board sensors was used as the inputs of the implementation of the proposed algorithms. The low-cost GNSS was used for the robust localization, to obtain the initial position and establish the fault boundary. An in-house LiDAR simulator constructed the point cloud from the vehicle pose to the modeled environments based on the LiDAR ray-casting considering the sensor specification (horizontal and vertical resolutions and laser beam divergences) and the predefined extrinsic calibration parameters of LiDAR. Since the vehicle poses for the LiDAR simulator were measured by the high-precision GNSS/INS, the vehicle poses measured by the high-precision GNSS/INS were considered as reference vehicle pose data. Accumulating the simulated point clouds based on the reference pose data, we constructed the base PCM layer. To simulate the changing environments, we modified the modeled environments by adding/removing/moving static obstacles. As a result, we constructed the PCM layer after map changes. The crowd-sourcing information from

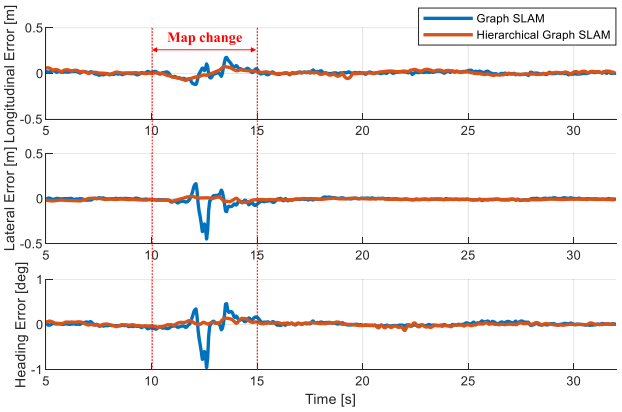

FIGURE 14. The comparison of the longitudinal, lateral, heading errors between the general Graph SLAM and the proposed hierarchical Graph SLAM.

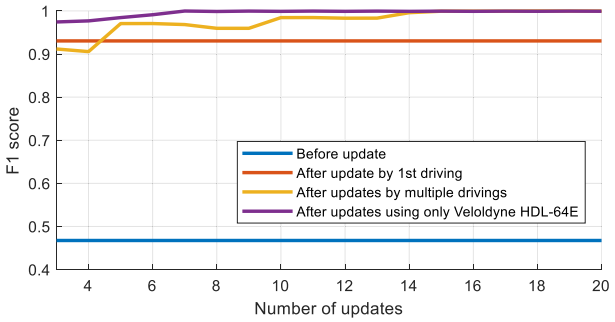

FIGURE 15. The F1-score by the number of drivings.

intelligent vehicles was simulated based on 20 driving data in modified environments with moving obstacles using and various extrinsic calibrations and sensor specifications such as Velodyne VLP-16, Velodyne HDL-32E, Velodyne HDL-64E, Robosense RS-LiDAR-16, and Valeo Scala. All processes in the simulation was performed in a computer with Intel Core i7-8750H CPU without cloud computing for evaluation.

B. ROBUST LOCALIZATION IN MAP-CHANGING ENVIRONMENTS

To detect the changes in the HD map, the exact pose (position and orientation) of LiDAR sensors must be required. Since the LiDAR calibration parameters were predefined, we derived the LiDAR pose from the vehicle pose. The accurate vehicle pose can be estimated through the proposed hierarchical Graph SLAM algorithm. In order to evaluate the proposed algorithm, a comparison of localization performance (longitudinal, lateral, and heading errors) for the general Graph SLAM and proposed hierarchical Graph SLAM is shown in Figure 14. The errors were evaluated by the reference pose data measured by the high-precision GNSS/INS. The differences between the environments and the base map occured from 10 to 15 seconds. As shown in Figure 14, the general Graph SLAM algorithm has poor performances in the environment change regions because the small matching ratio caused a mismatching between LiDAR point cloud and the PCM layer of the HD map. The proposed hierarchical Graph SLAM increased the matching ratio by generating the sub-map windows, the errors of the localization were reduced.

C. MAP CHANGE DETECTION AND UPDATE BASED ON CROWD-SOURCING VEHICLES

The results of map change detection based on a single driving and map update based on multiple drivings were represented by three confusion matrices, as shown in Table 1. The rows (*unchanged*, *new*, *deleted*, and *empty*) means the true classes of the voxels in the voxelization process of Figure 13. The true class of the voxel can be derived by the comparison between the base map constructed by previous environments and the modified map constructed by the modified environments. The first column of the Table 1 represents the class of the voxels in the base map. Since the base map is not updated, there is no change information (*new* and *deleted*) in the first column. The second column means the voxel class of the map updated by change detection of only one driving. The not-excluded moving vehicles increased the inaccurate estimate from actual *empty* voxels to *new* voxels in the change detection process. The misclassification from actual *deleted* voxels to *unchanged* voxels was caused due to the occlusion that the LiDAR cannot measure objects behind obstacles by LiDAR specifications. The final column represents the voxel class of the map updated by crowd-sourced PCM changes. We adopted the F1-score (17) to evaluate the map update performances because the number of voxels in each class is not imbalanced (*empty* $\gg$ *unchanged* $\gg$ *new*, *deleted*). As a result, F1-scores of the base map, the map updated by single driving, and the map updated by crowd-sourcing changes are 46.73%, 93.03%, and 99.92%, respectively. The performance of the map change detection due to the misclassification based on the LiDAR characteristics can be overcome by the crowd-sourcing changes detected by multiple vehicles.

$$F_1 = 2 \times \frac{precision \times recall}{precision + recall} \tag{17}$$

In order to analyze the effect of the number of drivings to the map update performance, the F1-scores based on the number of drivings are represented in Figure 15. In Figure 15, the blue line means the performance of the base map, which is represented by the first column of Table 1. The red line means the performance of the map updated by single driving, in which the map changes were detected by the Velodyne HDL-64E. The blue line means the performances of the map updates determined by the number of drivings, when the various sensors were used in the proposed framework. The purple line represents the performance of the map updates using the Velodyne HDL-64E. Since the Velodyne HDL-64E had high resolution with 64 layers, it provided fast approximation. The results of the map update based on various

TABLE 1. Confusion matrices about non-update, change detection based on single driving, and map update based on multiple drivings.

		Base map				Recall [%]	Map updated by change detection of one driving				Recall [%]	Map updated by crowd-sourcing changes				Recall [%]
		Unchanged	*New*	*Deleted*	*Empty*		*Unchanged*	*New*	*Deleted*	*Empty*		*Unchanged*	*New*	*Deleted*	*Empty*	
True	*Unchanged*	94312	0	0	0	100	94312	0	0	0	100	94312	0	0	0	100
	New	0	0	0	19782	0	0	19781	0	1	99.99	0	19667	0	115	99.42
	Deleted	30576	0	0	0	0	2825	0	27751	0	90.76	5	0	30571	0	99.98
	Empty	0	0	0	34198940	100	0	14214	0	34184726	99.96	0	0	0	34198940	100
Precision [%]		75.51	0	0	99.99	**F1 score 46.73**	97.09	58.19	100	100	**F1 score 93.03**	99.99	100	100	100	**F1 score 99.92**

sensors provided poor performances than the map change detection by Velodyne HDL-64L in a low number of updates because the various sensors were low resolutions than the Velodyne HDL-64L. Though the initial poor performances, various sensors provided a similar performance with the performance of the map updates using the Velodyne HDL-64E after 15 drivings.

VIII. EXPERIMENTS

A. EXPERIMENTAL ENVIRONMENTS

The experiments for verifying the proposed framework were performed based on the test vehicle of the simulation. Two LIDARs (Velodyne VLP-16), which have 100 meters detection range, ±3cm accuracy, ±15° horizontal field of view (FoV), 360° vertical (FoV), and 10Hz scanning rate, were additionally installed on the test vehicle. The LiDARs were used to generate the HD map and detect the map change based on the proposed framework. The high-precision GNSS/INS was used for the construction of the HD map and for a reference system to evaluate the performance of the proposed algorithm. A CPU specification of the vehicle computer was Intel® Core i7-8750H CPU @2.21 GHz CPU, and the cloud server computer was Intel ® Core i5-4670 @3.40 GHz CPU. The File Transfer Protocol (FTP) was applied for uploading and downloading of information between the crowd-sourcing vehicles and the cloud through the 4G wireless network communication.

The experiments were performed on Wangsimni street in South Korea, as shown in Figure 16-(a). To evaluate the crowd-sourcing map update framework, we need a test site where the actual environment had changed, but the changes were not reflected in the cloud HD map's PCM layer. As shown in Figure 16-(b) and (c), the test site was under construction from August 1st, 2019 to October 12th, 2019. The HD map's PCM layer was generated on August 1st, 2019, by using two LiDARs and RTK-GNSS/INS (RTK narrow int state). However, the map, created on August 1st, 2019, did not reflect the road environment changed on October 12th, 2019. Multiple test vehicles equipped with sensors were required to evaluate the crowd-sourcing framework. Since we only had one test vehicle, we operated the one test vehicle 38 times on October 12th, 2019, to emulate crowd-sourcing information

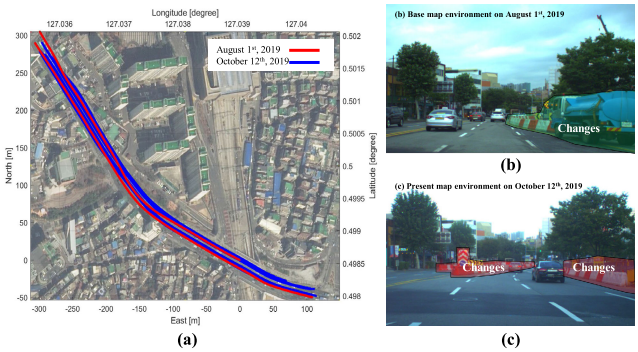

FIGURE 16. (a) Test site for experiment: Wangsimni-ro in South Korea. The test site under construction from (b) August 1st 2019 to (c) October 12th 2019.

from multiple vehicles. The feasibility of the replacement of various sensors to one sensor-based multiple drivings was shown in section VII. The vehicles detected the map changes and uploaded them to the cloud server. The cloud server received the map changes from the multiple vehicles and reflected the PCM layer changes. After the map update by crowd-sourcing information, the point cloud in the green polygon of Figure 16-(b) should be removed from the cloud PCM, the point cloud in red polygons of Figure 16-(c) should be added to the cloud.

B. ROBUST LOCALIZATION PERFORMANCE IN MAP-CHANGING ENVIRONMENT

Similar to the evaluation process in the simulation, the localization performance was evaluated in real environments. The localization performance was derived by the longitudinal, lateral, and heading errors compared with the reference poses measured by the high-precision GNSS/INS, as shown in Figure 17. The blue and orange lines in Figure 17 represented results of the general Graph SLAM and the proposed hierarchical Graph SLAM, respectively. Actual environment changes of Figure 16 occurred from the 20 seconds to 30 seconds. In the map-changing environments, maximum longitudinal errors of two approaches were 0.59 m and 0.3313 m. The lateral errors were 0.6663 m and 0.2015 m, respectively. Finally, the heading errors of the two approaches

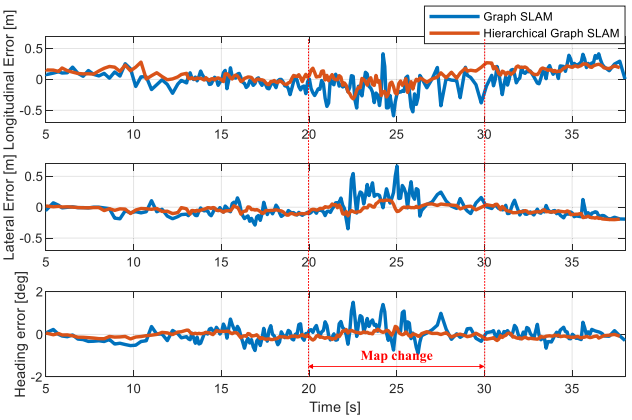

FIGURE 17. The comparison of RMS error for the longitudinal, lateral, heading between the normal Graph SLAM and proposed hierarchical Graph SLAM.

were 1.5071 ◦ and 0.3762 ◦. The proposed hierarchical Graph SLAM, which increased the matching ratio by accumulating the point cloud using sub-map windows, can provide better performance than the general Graph SLAM in overall map-changing environments. Applying the proposed algorithm showed similar tendencies in other logging data.

C. MAP CHANGE DETECTION AND UPDATE

Figure 18-(a) shows the ground truth about the point cloud changes (*new* and *deleted*) in the test area of Figure 16. The *new* and *deleted* states of the PCM are indicated by the red and green points, respectively, and the blue points mean the *unchanged* state. The ground truth of the change was constructed by comparing the MMS-based PCM of Figure 16-(b) and (c) for the performance evaluation of the map change detection algorithm. The point cloud height was limited from 0 to 1.5 meters because there are too many leaves of the street trees to interfere with the analysis.

Figure 18-(b) shows the change detection results classified by the proposed change detection algorithm with a single driving. There were two tuning factors for the map change detection algorithm. For the PCM existence update at the equation (11), the λ_{loc} is a tuning factor that can be determined by the performance of the robust localization. In this experience, we set the parameter λ_{loc} as 0.9 based on the experiment results of the robust localization. Another tuning factors are classification thresholds, which mean the minimum confidence belief to judge the changes (deleted or new). These factors were determined by localization performance and LiDAR characteristics. In this experiment, $th_{deleted}$ was set as 0.2 and, and th_{new} was set as 0.9.

Figure 18-(c) shows the map change merging results on the map cloud server updated by the crowd-sourcing vehicles. The multiple vehicles performed the change detection and uploaded the change results into the cloud server. In these experiments, the crowd-sourcing information of the multiple vehicles was emulated by 38 times driving with one test vehicle. The uploaded point cloud changes were merged into

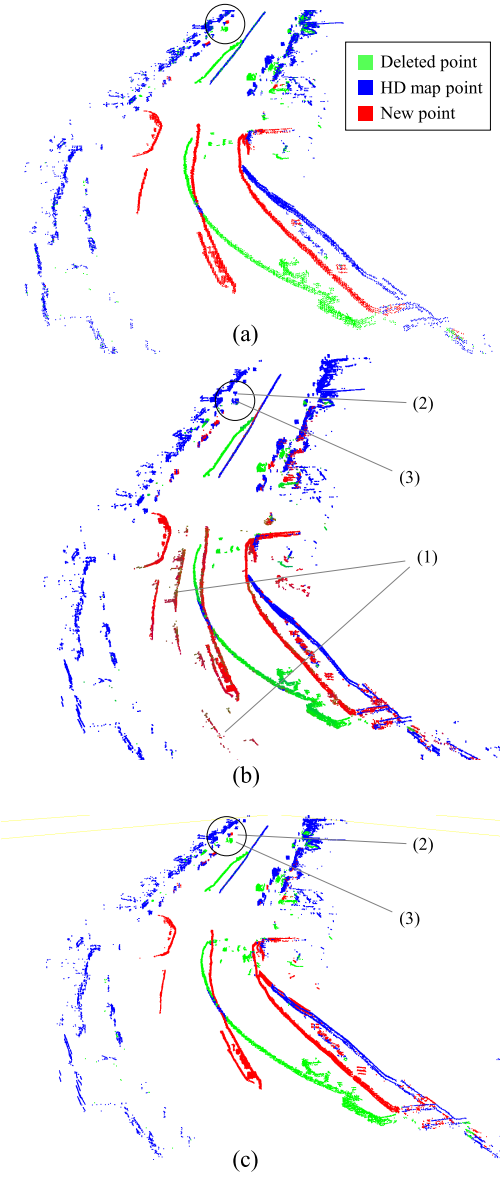

FIGURE 18. (a) ground truth (b) map change detection in the single vehicle (c) map change update results in the cloud using crowd-sourced map update.

the base PCM layer using an evidential merging algorithm. The $\lambda^{cs}_{deleted}$ and λ^{cs}_{new} for the algorithm was set as 0.2 and 0.9, respectively. The voxelization size of the merging algorithm is 10 cm.

Table 2 represented three confusion matrices about the base map constructed on August 1st, 2019, the map updated by the change detection algorithm, and the map updated by crowd-sourcing changes. Figure 19 shows the degree of performance improvement for each update. Point classes of Figure 18-(a) were represented as voxel classes corresponding with rows of confusion matrices. It means that the rows represent true classes of voxels. The point classes of Figure 18-(b) and (c) were represented as voxel classes in the second and third confusion matrices of the Table 2. The confusion matrices

TABLE 2. Confusion matrices about non-update, change detection based on single driving, and map update based on multiple drivings.

		Base map				Recall [%]	Map updated by change detection of one driving				Recall [%]	Map updated by crowd-sourcing changes				Recall [%]
		Unchanged	*New*	*Deleted*	*Empty*		*Unchanged*	*New*	*Deleted*	*Empty*		*Unchanged*	*New*	*Deleted*	*Empty*	
True	*Unchanged*	41775	0	0	0	100	40854	0	921	0	97.79	41239	0	536	0	98.71
	New	0	0	0	14402	0	0	11368	0	3034	78.93	0	13053	0	1349	90.63
	Deleted	9866	0	0	0	0	3215	0	6651	0	67.41	498	0	9368	0	94.95
	Empty	0	0	0	59042347	100	0	10349	0	59031998	99.98	0	390	0	59041957	99.99
Precision [%]		80.89	0	0	99.97	F1 score 47.48	92.70	52.34	87.83	99.99	F1 score 84.60	98.80	97.09	94.58	99.99	F1 score 96.84

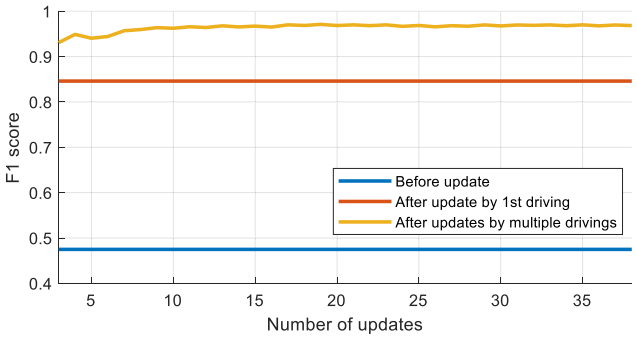

FIGURE 19. Improvement of map similarity with crowd-sourcing map update.

had similar tendencies with the confusion matrices of the simulation. Since the base map was not changed, there were no change voxels (*new* and *deleted*) in the first confusion matrix. The F1-score of the base map was represented as 47.48 %. The point cloud measured from the moving objects (such as car and pedestrian) indicated by (1) of Figure 18-(b) was misclassified as the *new* class. Accordingly, the *new*-estimated and actually *deleted* voxels in the second confusion matrix of Table 2 were 10349. Since the moving objects were not always existed, the misclassification by the change detection was overcome by the crowd-sourcing information; therefore, the *new*-estimated and actually *deleted* voxels were reduced to 390. The limited measurement range causes two problems: non-detection of *new* and non-removal of *unchanged*, which were represented as (2) and (3) in Figure 18. Although the vehicle cannot classify the *new* and *deleted* voxels far away from the vehicle, other vehicles, which were driven on other roads, can classify the changed information correctly. Accordingly, the *empty*-estimated and actually *new* voxels were reduced from 3034 to 1349 and the *unchanged*-estimated and actually *deleted* voxels were reduced from 3215 to 498, respectively. As a result, the F1-score was improved from 84.6 % to 96.84 % based on crowd-sourcing information.

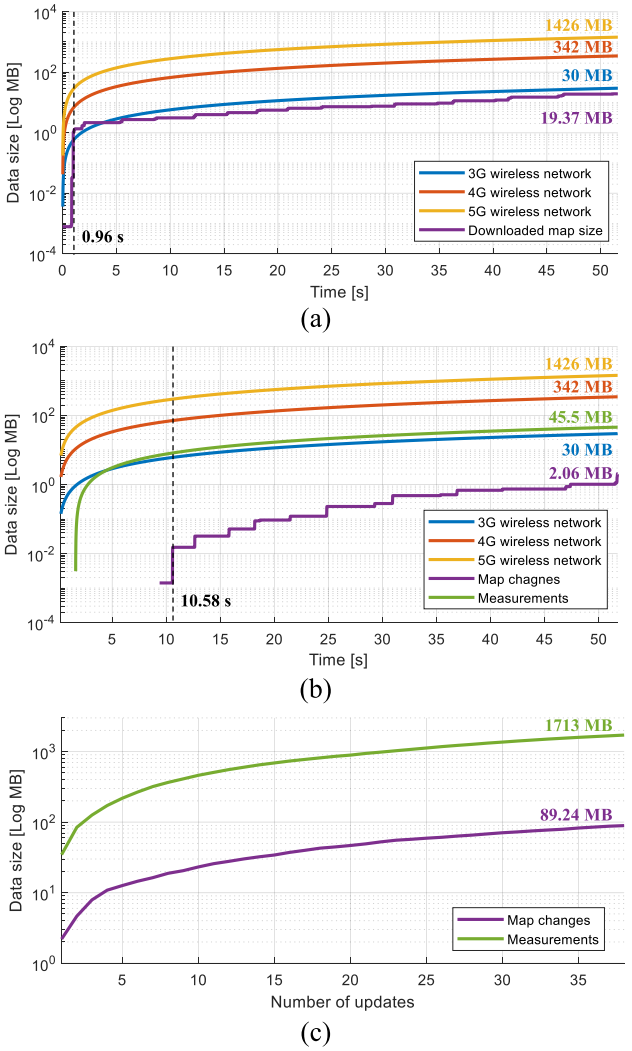

FIGURE 20. Traffic of wireless network communications about (a) downloading, (b) uploading from a single vehicle, and (c) uploading from crowd-sourcing vehicles.

D. TRAFFIC OF WIRELESS NETWORK COMMUNICATIONS

Figure 20 shows the traffic of the wireless network communications between the vehicles and the server. The blue, orange, and yellow lines in Figure 20 mean the data sizes, which can be downloaded through average 3G, 4G, and 5G wireless network communications in Korea. The purple line of Figure 20-(a) represents the PCM sizes downloaded from

the server to the vehicle. During the startup period until 0.96 seconds, all directional map data around the vehicle are downloaded, different from the general driving period where the vehicle downloads only the front-directional map data. If the vehicle is driven after the startup period, the 3G, 4G, and 5G wireless network communications can be used for the map downloading. Figure 20-(b) represents uploading data sizes to update the HD map's environment changes. The purple line represents the results processed by the proposed map change detection algorithm. Map changes in the regions passed by the vehicle are uploaded from the 10.58 seconds. On the other hand, the green line represents all measurement data acquired from in-vehicle sensors, the low-cost GNSS, and the LiDAR, as the approaches similar to the crowd-sourcing update of the landmark-level map. Since the proposed algorithm uploads the map changes, the uploaded data size can be reduced from 45.5 MB to 2.06 MB. A similar tendency can be shown in Figure 20-(c), representing the data uploaded from crowd-sourcing vehicles. During the 38 drivings, 1713 MB of all measurements are transmitted, and 89.24 MB of map changes proposed by the framework are uploaded to the server. The transmitted data can be dramatically reduced based on the proposed framework.

IX. CONCLUSION

This paper proposes a framework for updating point cloud layers in the HD maps through crowd-sourcing of LiDAR data from multiple vehicles. The proposed framework consists of five steps. In the first step, the crowd-sourcing vehicle downloads the point cloud layer from the HD map cloud server via internet communication. In the second step, a robust localization algorithm estimates the centimeter-level vehicle pose using the LiDAR and downloaded point cloud layer. The third step is the detection of changes to the downloaded point cloud layer, taking into account the laser properties of LiDAR. The detected point changes are updated to the map cloud server in the fourth step, and the updated point cloud changes are merged into the point cloud layer of the HD map in the final step. The main contributions of this paper are summarized as follows:

1) This paper proposed a map update framework based on crowd-sourcing vehicles. Based on this framework, a lot of vehicles can participate in making the HD map keep up-to-date state via the internet.
2) The proposed robust localization provides the centimeter-level vehicle pose even under the map change environments. The sub-map explicitly models the environmental change, which makes the proposed localization algorithm more robust in challenging situations.
3) The proposed change detection algorithm figures out the environmental changes based on the probabilistic and evidential approaches. These approaches can handle LiDAR measurement characteristics, such as beam divergence and multiple echoes. Based on the change detection algorithm, only the changed point cloud is uploaded to the cloud server, which reduces the network burden between the cloud and the vehicles.
4) The proposed merge algorithm updates the HD map based on the detected change information using evidential theory. The evidential theory represents more states than the probabilistic approach. In addition, the merged point cloud map reflects both the deleted point cloud and the newly added point cloud.

The proposed framework was verified and evaluated via simulations and experiments in a road environment changing condition. In the experiments, the crowd-sourcing results showed that the proposed algorithm can be kept up to date with 96.84 % F1-score (49.36 % improvement over the base map) through the proposed point cloud map update algorithm. In future research, we will extend the proposed framework for updating other map types as well as the point cloud map.

REFERENCES

[1] R. Liu, J. Wang, and B. Zhang, "High definition map for automated driving: Overview and analysis," *J. Navigat.*, vol. 73, no. 2, pp. 324–341, 2019.
[2] J. Jeong, Y. Cho, and A. Kim, "Road-SLAM : Road marking based SLAM with lane-level accuracy," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Jun. 2017, pp. 1736–1743.
[3] J. Levinson and S. Thrun, "Robust vehicle localization in urban environments using probabilistic maps," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2010, pp. 4372–4378.
[4] O. Pink, "Visual map matching and localization using a global feature map," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. Workshops, CVPR Workshops*, Jun. 2008, pp. 1–7.
[5] R. W. Wolcott and R. M. Eustice, "Robust LIDAR localization using multiresolution Gaussian mixture maps for autonomous driving," *Int. J. Robot. Res.*, vol. 36, no. 3, pp. 292–319, Mar. 2017. [Online]. Available: http://journals.sagepub.com/doi/10.1177/0278364917696568
[6] C. Jang, S. Cho, S. Jeong, J. K. Suhr, H. G. Jung, and M. Sunwoo, "Traffic light recognition exploiting map and localization at every stage," *Expert Syst. Appl.*, vol. 88, pp. 290–304, Dec. 2017. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0957417417304724
[7] J. Ziegler *et al.*, "Making bertha drive—An autonomous journey on a historic route," *IEEE Intell. Transp. Syst. Mag.*, vol. 6, no. 2, pp. 8–20, Apr. 2014.
[8] L. Kent. (2015). *HERE Introduces HD Maps for Highly Automated Vehicle Testing*. [Online]. Available: https://360.here.com/2015/07/20/here-introduces-hd-maps-for-highly-automated-vehicle-testing/
[9] TomTom. *TomTom HD Map With RoadDNA*. Accessed: Oct. 29, 2019. [Online]. Available: https://www.tomtom.com/automotive/automotive-solutions/automated-driving/hd-map-roaddna/
[10] Carmera. (2018). *Our Next Voyage: Mapping 750 Miles to Power Autonomous Taxis at The Villages, Florida*. [Online]. Available: https://medium.com/field-of-view/our-next-voyage-mapping-750-miles-to-power-autonomous-taxis-at-the-villages-florida-afd12dfa3907
[11] K. Jo, M. Lee, C. Kim, and M. Sunwoo, "Construction process of a three-dimensional roadway geometry map for autonomous driving," *Proc. Inst. Mech. Eng., D, J. Automobile Eng.*, vol. 231, no. 10, pp. 1414–1434, Sep. 2017.
[12] A. Schindler, P. Tobias, H. Zweitgutachter, and T. Sauer, "Vehicle self-localization with high-precision digital maps," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Jun. 2013, pp. 134–139. [Online]. Available: http://ieeexplore.ieee.org/document/6615239/
[13] A. Ranganathan, D. Ilstrup, and T. Wu, "Light-weight localization for vehicles using road markings," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Nov. 2013, pp. 921–927.
[14] X. Qu, B. Soheilian, and N. Paparoditis, "Vehicle localization using mono-camera and GEO-referenced traffic signs," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Jun. 2015, pp. 605–610.

[15] Z. Liu, H. Chen, H. Di, Y. Tao, J. Gong, G. Xiong, and J. Qi, "Real-time 6d LIDAR SLAM in large scale natural terrains for UGV," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Jun. 2018, pp. 662–667.

[16] W. Maddern, G. Pascoe, and P. Newman, "Leveraging experience for large-scale LIDAR localisation in changing cities," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2015, pp. 1684–1691.

[17] A. D. Stewart and P. Newman, "LAPS—localisation using appearance of prior structure: 6-DoF monocular camera localisation using prior pointclouds," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2012, pp. 2625–2632.

[18] Y. Xu, V. John, S. Mita, H. Tehrani, K. Ishimaru, and S. Nishino, "3D point cloud map based vehicle localization using stereo camera," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Jun. 2017, pp. 487–492.

[19] G. Grisetti, C. Stachniss, W. Burgard, and Others, "Improved techniques for grid mapping with rao-blackwellized particle filters," *IEEE Trans. Robot.*, vol. 23, no. 1, p. 34, 2007.

[20] S. Thrun, W. Burgard, and D. Fox, "A real-time algorithm for mobile robot mapping with applications to multi-robot and 3D mapping," in *Proc. Millennium Conf., IEEE Int. Conf. Robot. Automat., Symposia ICRA*, Apr. 2000, pp. 321–328.

[21] A. Hornung, K. M. Wurm, M. Bennewitz, C. Stachniss, and W. Burgard, "OctoMap: An efficient probabilistic 3D mapping framework based on octrees," *Auto. Robots*, vol. 34, no. 3, pp. 189–206, Apr. 2013. [Online]. Available: http://link.springer.com/10.1007/s10514-012-9321-0

[22] K. M. Wurm *et al.*, "OctoMap: A probabilistic, flexible, and compact 3D map representation for robotic systems," in *Proc. ICRA Workshop Best Pract. 3D Perception Modeling Mobile Manipulation*, vol. 2, 2010.

[23] J. Choi, "Hybrid map-based SLAM using a velodyne laser scanner," in *Proc. 17th Int. IEEE Conf. Intell. Transp. Syst. (ITSC)*, Oct. 2014, pp. 3082–3087.

[24] J. Levinson, M. Montemerlo, and S. Thrun, "Map-based precision vehicle localization in urban environments," in *Robotics: Science and Systems*, vol. 4. Citeseer, 2007.

[25] X. Wang, Y. Mizukami, M. Tada, and F. Matsuno, "Navigation of a mobile robot in a dynamic environment using a point cloud map," *Artif. Life Robot.*, Jul. 2020, Art. no. 0123456789. [Online]. Available: https://link.springer.com/article/10.1007/s10015-020-00617-3#Ack1, doi: 10.1007/s10015-020-00617-3.

[26] O. Dabeer, W. Ding, R. Gowaiker, S. K. Grzechnik, M. J. Lakshman, S. Lee, G. Reitmayr, A. Sharma, K. Somasundaram, R. T. Sukhavasi, and X. Wu, "An end-to-end system for crowdsourced 3D maps for autonomous vehicles: The mapping component," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Sep. 2017, pp. 634–641.

[27] M. Zhou, Y. Liu, Y. Wang, and Z. Tian, "Anonymous crowdsourcing-based WLAN indoor localization," *Digit. Commun. Netw.*, vol. 5, no. 4, pp. 226–236, Nov. 2019, doi: 10.1016/j.dcan.2019.09.001.

[28] F. Jomrich, J. Schmid, S. Knapp, A. Hoss, R. Steinmetz, and B. Schuller, "Analysing communication requirements for crowd sourced backend generation of HD maps used in automated driving," in *Proc. IEEE Veh. Netw. Conf. (VNC)*, Dec. 2018, pp. 1–8.

[29] D. Liang, Y. Guo, S. Zhang, S.-H. Zhang, P. Hall, M. Zhang, and S. Hu, "LineNet: A zoomable CNN for crowdsourced high definition maps modeling in urban environments," 2018, *arXiv:1807.05696*. [Online]. Available: http://arxiv.org/abs/1807.05696

[30] J. Lu, Y. Xin, Z. Zhang, S. Tang, C. Tang, and S. Wan, "Extortion and cooperation in rating protocol design for competitive crowdsourcing," *IEEE Trans. Comput. Social Syst.*, early access, Jan. 28, 2021, doi: 10.1109/TCSS.2020.2964284.

[31] C. Kim, S. Cho, M. Sunwoo, and K. Jo, "Crowd-sourced mapping of new feature layer for high-definition map," *Sensors*, vol. 18, no. 12, p. 4172, Nov. 2018. [Online]. Available: https://www.mdpi.com/1424-8220/18/12/4172

[32] C. Kim, K. Jo, B. Bradai, and M. Sunwoo, "Multiple vehicles based new landmark feature mapping for highly autonomous driving map," in *Proc. 14th Workshop Positioning, Navigat. Commun. (WPNC)*, Oct. 2017, pp. 1–6.

[33] 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, Sep. 2018. [Online]. Available: http://www.mdpi.com/1424-8220/18/9/3145

[34] D. Pannen, M. Liebner, W. Hempel, and W. Burgard, "How to keep HD maps for automated driving up to date," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2020, pp. 2288–2294.

[35] Daimler. *Daimler and HERE to Bring HD Live Map to Future Mercedes-Benz Models*. Accessed: Jun. 12, 2019. [Online]. Available: https://media.daimler.com/marsMediaSite/en/instance/ko/Daimler-and-HERE-to-bring-HD-Live-Map-to-future-Mercedes-Benz-models.xhtml?oid=33472763

[36] Justyna Zander. (2019). *DRIVE Mapping Introduces HD Map Update for Safe Driving*. [Online]. Available: https://blogs.nvidia.com/blog/2019/03/20/drive-mapping-hd-map-update/

[37] Bosch. *Bosch Road Signature*. Accessed: Jun. 12, 2019. [Online]. Available: https://www.bosch-mobility-solutions.com/en/products-and-services/passenger-cars-and-light-commercial-vehicles/automated-driving/bosch-road-signature/

[38] TomTom. *Qualcomm and TomTom to Crowdsource High-Definition Map Data*. Accessed: Oct. 29, 2019. [Online]. Available: https://www.roadtraffic-technology.com/news/newsqualcomm-and-tomtom-to-crowd-source-high-definition-map-data-5752324/

[39] S. Sanjay, "Self-healing maps for autonomous driving," in *Proc. NVIDIA GPU Technol. Conf.*, 2017. [Online]. Available: https://on-demand-gtc.gputechconf.com/gtcnew/sessionview.php?sessionName=s7665-the+self-healing+map+for+automated+driving

[40] D. Girardeau-Montaut, M. Roux, R. Marc, and G. Thibault, "Change detection on points cloud data acquired with a ground laser scanner," *Int. Arch. Photogramm., Remote Sens. Spatial Inf. Sci.*, vol. 36, no. 3, p. W19, 2005. [Online]. Available: http://www.isprs.org/proceedings/XXXVI/3-W19/papers/030.pdf

[41] L. Wellhausen, R. Dube, A. Gawel, R. Siegwart, and C. Cadena, "Reliable real-time change detection and mapping for 3D LiDARs," in *Proc. IEEE Int. Symp. Saf., Secur. Rescue Robot. (SSRR)*, Oct. 2017, pp. 81–87. [Online]. Available: http://ieeexplore.ieee.org/document/8088144/

[42] F. Ferri, M. Gianni, M. Menna, and F. Pirri, "Dynamic obstacles detection and 3D map updating," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Sep. 2015, pp. 5694–5699.

[43] J. P. Underwood, D. Gillsjo, T. Bailey, and V. Vlaskine, "Explicit 3D change detection using ray-tracing in spherical coordinates," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2013, pp. 4735–4741.

[44] W. Xiao, B. Vallet, M. Brédif, and N. Paparoditis, "Street environment change detection from mobile laser scanning point clouds," *ISPRS J. Photogramm. Remote Sens.*, vol. 107, pp. 38–49, Sep. 2015.

[45] W. Xiao, B. Vallet, and N. Paparoditis, "Change detection in 3D point clouds acquired by a mobile mapping system," *ISPRS Ann. Photogramm., Remote Sens. Spatial Inf. Sci.*, vols. II–5/W2, pp. 331–336, Oct. 2013. [Online]. Available: http://www.isprs-ann-photogramm-remote-sens-spatial-inf-sci.net/II-5-W2/331/2013/

[46] W. Burgard, C. Stachniss, and D. Hähnel, "Mobile robot map learning from range data in dynamic environments," in *Autonomous Navigation in Dynamic Environments*. Berlin, Germany: Springer, 2007, pp. 3–28. [Online]. Available: http://link.springer.com/10.1007/978-3-540-73422-2_1

[47] 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 *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2016, pp. 4558–4563.

[48] 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 Trans. Robot.*, vol. 33, no. 4, pp. 964–977, Aug. 2017.

[49] G. Trehard, Z. Alsayed, E. Pollard, B. Bradai, and F. Nashashibi, "Credibilist simultaneous localization and mapping with a LIDAR," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Sep. 2014, pp. 2699–2706.

[50] G. Trehard, E. Pollard, B. Bradai, and F. Nashashibi, "On line mapping and global positioning for autonomous driving in urban environment based on evidential SLAM," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Jun. 2015, pp. 814–819.

[51] F. Abrate, B. Bona, M. Indri, S. Rosa, and F. Tibaldi, "Multi-robot map updating in dynamic environments," in *Distributed Autonomous Robotic Systems* (Springer Tracts in Advanced Robotics), vol. 83. Mumbai, India: STAR, 2012, pp. 147–160.

[52] N. Shaik, T. Liebig, C. Kirsch, and H. Müller, "Dynamic map update of non-static facility logistics environment with a multi-robot system," in *KI 2017: Advances in Artificial Intelligence* (Lecture Notes in Computer Science), vol. 10505. Cham, Switzerland: Springer, doi: 10.1007/978-3-319-67190-1_19.

[53] M. Labbé and F. Michaud, "Long-term online multi-session graph-based SPLAM with memory management," *Auto. Robots*, vol. 42, no. 6, pp. 1133–1150, Aug. 2018, doi: 10.1007/s10514-017-9682-5.

[54] K. Jo, S. Cho, C. Kim, P. Resende, B. Bradai, F. Nashashibi, and M. Sunwoo, "Cloud update of tiled evidential occupancy grid maps for the multi-vehicle mapping," *Sensors*, vol. 18, no. 12, p. 4119, Nov. 2018. [Online]. Available: http://www.mdpi.com/1424-8220/18/12/4119
[55] 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, Sep. 2019. [Online]. Available: https://www.mdpi.com/1424-8220/19/19/4116
[56] R. Kummerle, G. Grisetti, H. Strasdat, K. Konolige, and W. Burgard, "g2o: A general framework for graph optimization," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2011, pp. 3607–3613. [Online]. Available: http://ieeexplore.ieee.org/document/5979949/
[57] M. Magnusson, A. Nuchter, C. Lorken, A. J. Lilienthal, and J. Hertzberg, "Evaluation of 3D registration reliability and speed—A comparison of ICP and NDT," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2009, pp. 3907–3912.
[58] S. Pang, D. Kent, X. Cai, H. Al-Qassab, D. Morris, and H. Radha, "3D scan registration based localization for autonomous vehicles—A comparison of NDT and ICP under realistic conditions," in *Proc. IEEE Veh. Technol. Conf.*, Aug. 2018, pp. 5–9.
[59] M. Aguilar-Moreno and M. Gra na, "A comparison of registration methods for SLAM with the M8 Quanergy LiDAR," in *Proc. Int. Workshop Soft Comput. Models Ind. Environ. Appl.*, Springer, Sep. 2020, pp. 824–834.
[60] H. Lim, S. Hwang, S. Shin, and H. Myung, "Normal distributions transform is enough: Real-time 3D scan matching for pose correction of mobile robot under large odometry uncertainties," in *Proc. 20th Int. Conf. Control, Autom. Syst. (ICCAS)*, Oct. 2020, pp. 1155–1161.

CHANSOO KIM (Member, IEEE) received the B.S. degree in mechanical engineering and the Ph.D. degree in automotive engineering from Hanyang University, Seoul, South Korea, in 2020. Since 2020, he has been working as a Postdoctoral Researcher with the Automotive Control and Electronics Laboratory (ACE Lab), Department of Automotive Engineering, Hanyang University, researching system design and implementation of autonomous driving and autonomous valet parking. His main research interests include object detection and prediction, deep learning, localization, mapping, simultaneous localization and mapping, map updating, and real-time systems for autonomous cars.

SUNGJIN CHO (Student Member, IEEE) received the B.S. degree in electronic engineering from Hanyang University, Seoul, South Korea, in 2014, where he is currently pursuing the Ph.D. degree with the Automotive Control and Electronics Laboratory. His main research interests include precise positioning and localization, information fusion theories, real-time systems for an autonomous car, and design of parallel computing architecture for vehicle localization.

MYOUNGHO SUNWOO (Member, IEEE) received the B.S. degree in electrical engineering from Hanyang University, in 1979, the M.S. degree in electrical engineering from the University of Texas at Austin, in 1983, and the Ph.D. degree in system engineering from Oakland University, in 1990. He joined General Motors Research (GMR) Laboratories, Warren, MI, USA, in 1985, and has worked in the area of automotive electronics and control for 30 years. During his nine-year tenure at GMR, he worked on the design and development of various electronic control systems for powertrains and chassis. Since 1993, he has been leading research activities as a Professor with the Department of Automotive Engineering, Hanyang University. His work has focused on automotive electronics and controls (such as modeling and control of internal combustion engines, design of automotive distributed real-time control systems, intelligent autonomous vehicles, and automotive education programs).

PAULO RESENDE (Member, IEEE) received the degree in electrical and computer engineering from the University of Coimbra, Portugal, in 2006. In 2007, he worked for ESRIN, the European Space Agency Center for Earth Observation, Frascati, Italy. In early 2008, he worked with the Earth Observation and Command and Control Engineering Area of Critical Software, with Taveiro, and with Lisbon, Portugal. In mid 2008, he integrated the Project Team IMARA with the INRIA Research Center, Rocquencourt, France, where he worked on trajectory planning and control of driverless and cooperative vehicles (cybercars). Since 2014, he has been working for the Valeo Driving Assistance Research, Bobigny, France, as an Autonomous Driving System Team Leader in the development of a Driving Assistance Research Traversable (DART) Platform to support the development of Advanced Driving Assistance Systems (ADAS) technologies for highly and fully automated vehicles, and in particular in the development of Drive4U Locate precise localization systems.

BENAZOUZ BRADAÏ (Member, IEEE) received the Ph.D. degree in multisensor fusion from Haute Alsace University, France, in 2007. From 2007 to 2011, he was an Algorithm Engineer and an Expert of ADAS functions, including Lighting Automation, Traffic Signs/lights recognition, and Eco-Driving for Hybrid Vehicles Using Cameras and Multi Sensor Fusion with navigation maps. From 2011 to 2018, he has been working as a Project Manager and a Senior Expert of ADAS and Automated Driving, including Valeo Urban Automated Driving Drive4U Project. Since 2019, he has been the Innovation Platform Manager of Automated Driving developments. His role covers the management of transversal/generic Automated Driving development and several Valeo Automated Driving projects (Cruise4U, Drive4, and eDeliver4U). His main research interests include automated driving with several scientific contributions and patents in multi-sensors fusion, precise localization, and mapping and automated driving systems. He is a member of various professional associations, including ADASIS Forum, SAE, and SIA in France.

KICHUN JO (Member, IEEE) received the B.S. degree in mechanical engineering and the Ph.D. degree in automotive engineering from Hanyang University, Seoul, South Korea, in 2008 and 2014, respectively. From 2014 to 2015, he worked with the Automotive Control and Electronics Laboratory (ACE Lab), Department of Automotive Engineering, Hanyang University, doing research on system design and implementation of autonomous cars. Since 2015, he has working with the Valeo Driving Assistance Research, Bobigny, France, working on highly automated driving. His main research interests include localization and mapping, objects tracking, information fusion, vehicle state estimation, behavior planning, vehicle motion control for highly automated vehicles, and hardware and software platform design for autonomous cars based on distributed real-time embedded systems and in-vehicle networks.