IEEE Transactions on Robotics (T-RO) paper presented at
2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
October 14-18, 2024. Abu Dhabi, UAE

Autonomous Vehicle Localization Without Prior High-Definition Map

Sangmin Lee, *Student Member, IEEE* and Jee-Hwan Ryu, *Senior Member, IEEE*

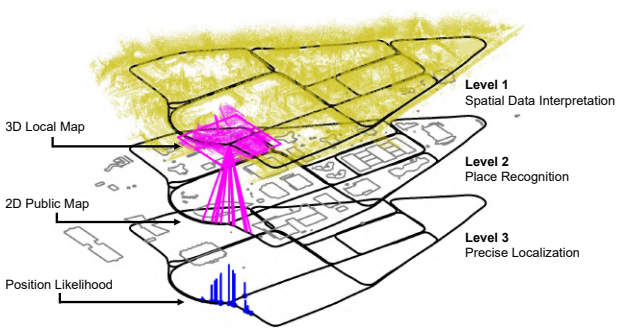
***Abstract*— Accurate localization by which vehicles can arrive at their destination while accurately following a given route is one of the most important factors for autonomous driving. In recent years, numerous studies have been conducted to achieve accurate localization using high-definition (HD) maps. Based on the HD map information (e.g., spatial data, lane, and traffic sign), autonomous vehicles can localize themselves by matching the surrounding spatial information obtained from onboard sensors to the HD maps. However, generating HD maps is a time-consuming and costly task. This study introduces a time-saving, effective, and accurate localization method inspired by humans, using only onboard sensors and publicly available two-dimensional (2D) map information. Similar to the multi-level localization process performed by humans, the proposed method interprets and matches the surrounding spatial data to the publicly available 2D maps using deep-learning-based place recognition and simultaneous localization and mapping (SLAM), thereby enabling autonomous vehicles to localize even without prior HD maps. Through the proposed method, our framework enables autonomous vehicles to perform maximally decimeter-level accurate localization without using HD maps. Evaluation of the proposed method using various datasets and publicly available map sources demonstrates that accurate global localization can be achieved without prior HD maps.**

I. Introduction

Localization is the key element of autonomous vehicles. Without localization, autonomous vehicles cannot estimate their position or reach their destination along a planned route. Over the past few decades, global positioning system (GPS)-based autonomous vehicle localization, particularly differential GPS (DGPS), has been studied for accurate localization. However, GPS-based localization, including DGPS, often fails when vehicles enter GPS shadow areas, such as tunnels, under bridges, and densely populated areas, which are common in urban environments. Hence, pre-built high-definition (HD) map-based localization methods [1]–[3] with their highly accurate and abundant information for autonomous driving (e.g., 3D spatial data, lane, traffic sign, and road types) have been used for localization.

However, autonomous vehicle localization using HD maps has limitations owing to their heavy information characteristics. Generating and updating HD maps is costly and time-consuming owing to the vast amount of information they contain [4], [5]. A mobile mapping system (MMS), which is designed to generate HD maps, needs to be re-mapped and updated every time there are changes in HD maps, such as lane changes, new buildings, or road changes. Additionally, the accessibility of HD maps is restricted. Autonomous vehicles cannot contain full-size HD maps owing to the massive amount of spatial and semantic information. Although some autonomous vehicles access HD maps using high-speed wireless communication, such as 5G, unreliable network connections can indeed lead to failures in the localization and navigation of autonomous vehicles [6].

In recent years, localization methods utilizing publicly available 2D vector graphic-based map sources, such as OpenStreetMap [7], national geographic information [8],



(a) Proposed multi-level localization framework

(b) Localization from publicly available map and local 3D map.

Fig. 1. Overall illustration of the proposed framework. In the first sequence, the proposed method acquires the surrounding environment information using onboard vehicle sensors. In the second sequence, the acquired 3D information is retrieved to a publicly available 2D map for place recognition using the deep-learning network. In the last sequence, the proposed method filters outliers and precisely localizes using semantic information of 2D digital maps and 3D maps. Hence, the proposed multi-level localization framework can work even in the absence of a high-definition map or high-precision global positioning system.

This research was supported in part by the Robot Industry Core Technology Development Program (20023294) funded By the Ministry of Trade, Industry & Energy(MOTIE, Korea) and in part by the National Research Foundation of Korea under Grant NRF-2020R1A2C200416915.

Sangmin Lee and Jee-Hwan Ryu are affiliated with the Department of Civil and Environmental Engineering, Korea Advanced Institute of Science and Technology (KAIST), Daejeon 34141, Korea (e-mail: iismn@kaist.ac.kr; jhryu@kaist.ac.kr)

and Google Maps, instead of pre-built HD maps have been proposed. Although prior research primarily addressed the metric localization problem, the achieved localization accuracy often exhibits deviations on the submeter scale [9]–[11]. In addition, approaches based on road networks [12], [13] may encounter challenges in the presence of fundamental ambiguities in the road layout, such as a Manhattan-shaped Road network. These challenges can result in difficulties for autonomous vehicles in accurately following the reference path or reaching the intended destination.

Furthermore, research endeavors have explored autonomous vehicle localization using aerial view image-based publicly available map sources. Aerial view-based map matching approaches exhibit superior accuracy compared to vector graphic-based localization methods owing to the wealth of information they provide. However, these approaches still face challenges in maintaining localization accuracy within sub-meter divergence during the process [14]–[16]. Furthermore, they heavily rely on a precise initial guess from GPS [14]–[17], introducing significant errors when vehicles traverse GPS-denied areas or encounter obstructing structures. Consequently, prior methods struggle to address the kidnapped problem and cannot effectively recover from localization errors associated with large initial guess errors. Notably, using high-resolution aerial image patches demands substantial computational resources, posing a challenge for vehicles in storing and processing complete aerial view patches across extensive geographical areas. In Section II, we will conduct an in-depth review of publicly available map-based localization methods.

This study presents a precise localization technique, surpassing other methods relying on public map sources. The proposed approach achieves accuracy up to the decimeter level using uncomplicated vector graphic-based publicly available maps, obviating the requirement for a prior HD map or DGPS. It is inspired by how humans localize themselves on navigation maps. Humans first localize their location on the map by comparing the rough shape of their surroundings with 2D top-view images of a publicly available map. Then, they precisely localize themselves by matching landmarks of nearby objects, such as buildings and roads [18]–[20]. Inspired by the human localization sequence, our proposed method also suggests a multi-step localization for autonomous vehicles following a similar sequence as shown in Fig. 1. First, it generates a local 3D map using a short-term SLAM algorithm to compare the surroundings to the top view of a publicly available map. Next, a Siamese-structured place recognition network simultaneously correlates and matches landmark information, such as buildings and roads, between distinct local 3D maps and vector-based 2D public maps, enabling global localization of the vehicle even in situations where DGPS or HD maps are unavailable. Hence, the proposed method can localize vehicles precisely at the decimeter level maximally, unlike previously publicly available map-based methods. The primary contribution of this paper lies in the development of a precise localization method for autonomous vehicles, achieving accuracy up to the decimeter level, even in locations where vehicles have not previously traversed. The key features of the proposed method include the following:

- It offers precise 3 degrees of freedom (3-DoF) vehicle localization utilizing only publicly accessible 2D maps.
- It does not depend on expensive differential GPS or the acquisition of prebuilt HD maps for localization.
- It addresses challenges in place recognition and metric localization within scenarios beyond the capabilities of GPS or conventional public map-based methods.
- It provides localization ability in various vehicles even if the LiDAR sensor configuration changes or has never been visited before.

II. Related Works

This section reviews previous HD-based localization methods and publicly available map-based localization methods.

A. HD map-based localization methods

Instead of GPS-based localization, which has inevitable failures owing to shadow areas, most autonomous vehicle developers use pre-built HD map-based localization methods that were built using an MMS comprising a precise inertial navigation system (INS), range-bearing sensor, and camera. Based on the HD map, they use spatial and visual information for accurate vehicle localization. Furthermore, they post-process LiDAR intensity, 3D structure, or visual features, and match HD map information for autonomous vehicle localization.

Levinson et al. [21] deployed an intensity-based HD map for precise vehicle localization, generated the HD map containing reflectance information using SLAM, and localized the vehicle using a particle filter. They were able to localize their vehicle by matching the reflectance information of the real-time onboard LiDAR scan with that of the generated HD map. Later, Levinson et al. [22] extended the reflectance map to a probabilistic map and achieved more accurate localization. However, both methods require prior HD maps, large data storage, and additional post-processing for localization. Therefore, considerably lighter lane-based localization has been proposed. Schreiber et al. [23] localized their vehicle by matching the detected lane to the lane information on the HD map using a camera. However, this can reduce localization accuracy when the vehicle travels in a straight-ahead solid lane. Therefore, Ma et al. [24] used traffic signals and lane information simultaneously for precise vehicle position estimation in the lateral and longitudinal directions. In addition, they used a considerably lighter HD map containing only the necessary lane and traffic sign information. Nevertheless, the aforementioned methods still require an HD map for localization and cannot localize in places the vehicle has never visited or non-HD map areas.

B. Publicly available map-based localization

Because of the difficulty of generating HD maps, researchers have recently attempted to localize autonomous

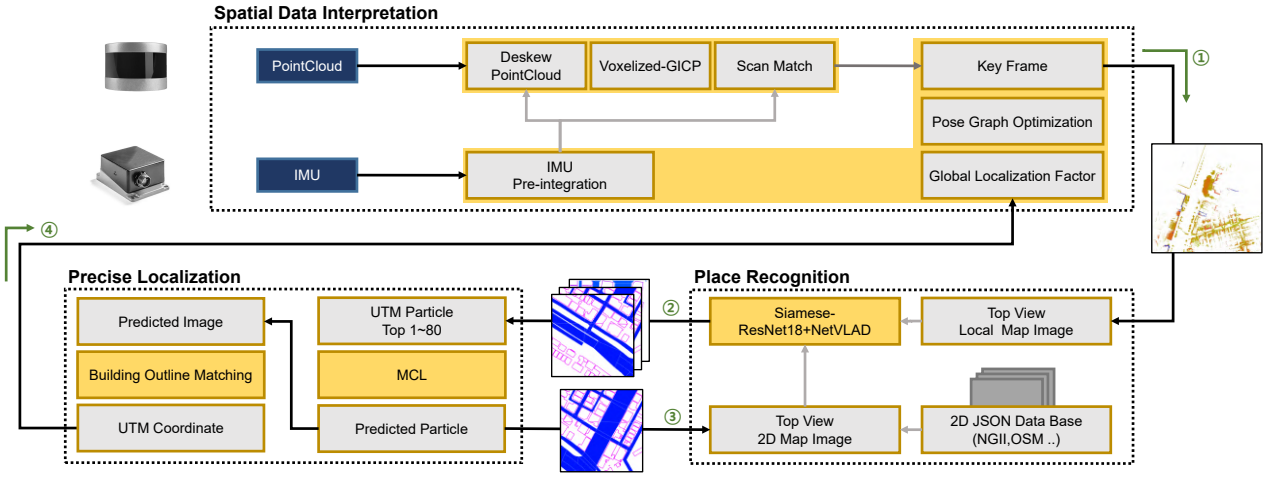


Fig. 2. Overall framework of the proposed method. The spatial data interpretation sequence generates a 3D local map from the point cloud and IMU data generated from a vehicle. In the place recognition sequence, a 3D local map is converted to an aerial-view image and retrieved to the corresponding cropped publicly available 2D map image. After retrieval, both retrieved 2D map images and 3D local map images feed into a precise localization sequence to estimate accurate position via particle filter and template matching. Finally, the proposed method can estimate accurate vehicle position from UTM coordinates of the 3D local map image and eliminates drift over time.

vehicles using publicly available maps (e.g., Google Maps, OpenStreetMap, national geographic information system, etc.) instead of an HD map.

Several studies have used the road network information from OpenStreetMap. Brubaker et al. [12] estimated the driving path using visual odometry and localized the vehicle by matching it with the road network on OpenStreetMap. Conversely, Ruchti et al. [13] classified roads using ground point cloud data from 3D LiDAR and matched it with the OpenStreetMap road network. Furthermore, they use the Monte-Carlo localization method with their sensor model, which calculates the weight using road cells and non-road cells from classified point cloud data. However, using only the road network for localization is ambiguous when a vehicle travels in a Manhattan-like road network. Researchers have used building information and road networks to solve this ambiguity. Yan et al. [9] proposed a global descriptor for place recognition using OpenStreetMap. A 4-bit descriptor encodes road intersections and building gaps by counting pixels in each quartered 360° sensor range and matching it with the encoded descriptor value from OpenStreetMap. Unlike Ruchti et al., Yan et al. used building gap information and adopted a particle filter-based weight update to converge the vehicle position. Cho et al. [11] proposed a more accurate version of the global descriptor based on a virtual building scan on OpenStreetMap. This method localized vehicle by matching each global descriptor from a virtual building scan on OpenStreetMap and an actual building scan from an onboard LiDAR. Unlike previous global descriptors, the proposed global descriptor was rotation invariant and did not need to drive a long distance for the particle filter convergence. However, localization accuracy exceeds the sub-meter level, which is not sufficiently high for autonomous driving.

Additionally, studies on localization based not on 2D vector-based publicly available maps but on publicly available aerial view images have also been conducted. Tang et al. [14]–[16] preprocessed the aerial view image to generate virtual radar or LiDAR scans from the corresponding location. Tang et al. attempted to compare virtual scan images with the actual radar or LiDAR aerial view images for place recognition and metric localization. However, their methods require additional large-scale deep learning networks or image preprocessing to extract valid information from the aerial image, and their metric localization accuracy exceeds the sub-meter level. This could impose limitations on the autonomous vehicle's reference path following or precise arrival at the destination. Fervers et al. [17] proposed methods that can solve high-precision localization at decimeter level maximum based on camera and LiDAR. Using the methods proposed by Fervers et al., state-of-the-art performance was demonstrated through a deep learning network without other preprocessing steps. However, they rely on an accurate DGPS to estimate the initial position, which can introduce significant errors when vehicles enter GPS-denied areas or navigate through areas surrounded by buildings. In addition, the use of high-resolution aerial image patches requires significant computational resources, even though aerial image patches require less memory than HD maps (i.e., 8 km^2 area, 2D public map 1.2 MB, aerial image 1 GB [8]). This presents a challenge for vehicles when it comes to storing and processing complete aerial view patches across vast geographic areas.

Although the aforementioned publicly available map-based localization methods can localize vehicles, the localization accuracy often diverges over the sub-meter level. This inaccuracy can be critical because it can cause self-driving cars to start or arrive at an incorrect location. Furthermore, previously proposed methods require more than a hundred

meters of initial driving to converge the vehicle's position or the large storage of aerial images, which require a much larger capacity than vector graphics-based public maps. To address these problems, this paper proposes a method for localization with higher precision using only publicly available maps. The accuracy of the proposed method can be compared with that of the HD map-based methods, and it can localize the vehicle without a lengthy initialization procedure for place recognition.

III. Proposed Method

This section proposes a publicly available map-based precise localization method. The proposed method comprises a multi-level localization sequence: spatial data interpretation, place recognition, and precise localization. First, *spatial data interpretation* sequence generates a top-view image from a local point cloud map. Next, *place recognition* sequence roughly interprets road and building segments from the local map. Then, it retrieves the best-matched places from the publicly available map using the top-view 3D local map image. Finally, *precise localization* sequence localizes the vehicle up to the decimeter level by fusing the 3D and 2D map images.

In the rest of this section, we will detail how the proposed method interprets spatial data and localizes vehicles with only publicly available 2D maps. The overall flow of the proposed method is described in Fig. 2.

A. Spatial Data Interpretation

As humans analyze the surrounding environment and localize themselves approximately, the first layer was constructed to interpret the surrounding spatial data from the onboard LiDAR and inertial measurement unit (IMU). However, the storage unit of an autonomous vehicle is not sufficient to contain complete spatial data during driving. Therefore, a spatial data-handling sequence is proposed to build a temporal 3D local map and compare it with a publicly available map. Consequently, it is unnecessary to save the entire spatial data for loop closing to eliminate drift over time, considering the proposed method continuously updates the vehicle position on the reference publicly available map.

To generate a temporal 3D local map from the surrounding spatial data, the proposed method uses the factor graph SLAM structure, which solves the Maximum a Posteriori (MAP) problem to optimize the local 3D map. Our factor graph comprised a pre-integrated IMU measurement, LiDAR-based motion estimation, initial heading compensation, and global localization.

1) Pre-integrated IMU measurement: The proposed method exploits a pre-integrated IMU measurement to estimate the initial guess of the LiDAR scan match and compensate for LiDAR distortion from the vehicle motion. First, we denote the vehicle state as

$$\mathbf{X}_I = [\mathbf{R}, \mathbf{p}, \mathbf{v}, \mathbf{b}_a, \mathbf{b}_g], \tag{1}$$

where $\mathbf{R} \in SO(3)$ comprises 9 elements. $\mathbf{p} \in \mathbb{R}^3$ is the position of the vehicle, $\mathbf{v}$ is the velocity, and $\mathbf{b}_a, \mathbf{b}_g$ are the biases of the IMU accelerometer and gyroscope. We denote the world and body frames as $\mathbf{W}$ and $\mathbf{B}$, respectively. From the IMU, we can obtain the acceleration $\mathbf{a}$ and angular velocity $\boldsymbol{\omega}$ in the body frame, expressed as

$$^{\mathbf{B}}\hat{\omega}_{\mathbf{W/B}}(t) = {}^{\mathbf{B}}\omega_{\mathbf{W/B}}(t) + \mathbf{b}^g(t) + \boldsymbol{\eta}^g(t), \tag{2}$$

$$\hat{\mathbf{a}}_{\mathbf{B}}(t) = {}^{\mathbf{W}}\mathbf{R}_{\mathbf{B}}({}^{\mathbf{W}}\mathbf{a}(t) - {}^{\mathbf{W}}\mathbf{g}) + \mathbf{b}^a(t) + \boldsymbol{\eta}^a(t), \tag{3}$$

where the measurements are affected by the slowly varying bias $\mathbf{b}^g, \mathbf{b}^a$ and white noise $\boldsymbol{\eta}^g, \boldsymbol{\eta}^a$. $^{\mathbf{W}}\mathbf{R}_{\mathbf{B}}$ is the rotation matrix of body frame $\mathbf{B}$ with relative to world frame $\mathbf{W}$. From the measurements, the vehicle rotation, velocity, and position can be computed consecutively at $t + \Delta t$.

$$\begin{aligned} {}^{\mathbf{W}}\mathbf{R}_{\mathbf{B}}(t+\Delta t) = {}& {}^{\mathbf{W}}\mathbf{R}_{\mathbf{B}}(t) \\ & \cdot \exp\left(({}^{\mathbf{B}}\hat{\omega}_{\mathbf{W/B}}(t) - \mathbf{b}^g(t) - \boldsymbol{\eta}^g(t))\Delta t\right), \end{aligned} \tag{4}$$

$$\begin{aligned} {}^{\mathbf{W}}\mathbf{v}(t+\Delta t) = {}& {}^{\mathbf{W}}\mathbf{v}(t) + {}^{\mathbf{W}}\mathbf{g}\Delta t \\ & + {}^{\mathbf{W}}\mathbf{R}_{\mathbf{B}}(t)(\hat{\mathbf{a}}_{\mathbf{B}}(t) - \mathbf{b}^a(t) - \boldsymbol{\eta}^a(t))\Delta t, \end{aligned} \tag{5}$$

$$\begin{aligned} {}^{\mathbf{W}}\mathbf{p}(t+\Delta t) = {}& {}^{\mathbf{W}}\mathbf{p}(t) + {}^{\mathbf{W}}\mathbf{v}(t)\Delta t \\ & + \frac{1}{2}{}^{\mathbf{W}}\mathbf{R}_{\mathbf{B}}(t)(\hat{\mathbf{a}}_{\mathbf{B}}(t) - \mathbf{b}^a(t) - \boldsymbol{\eta}^a(t))\Delta t^2. \end{aligned} \tag{6}$$

Subsequently, the proposed method adopts IMU pre-integration on the manifold proposed in [25], and can derive the vehicle state $\mathbf{X}_I$ between consecutive timestamps. Pre-integrated measurements between timestamps i and j can be obtained by

$$\Delta\mathbf{R}_{ij} = \mathbf{R}_i^{\mathsf{T}}\mathbf{R}_j \mathrm{Exp}(\delta\boldsymbol{\phi}_{ij}), \tag{7}$$

$$\Delta\mathbf{v}_{ij} = \mathbf{R}_i^{\mathsf{T}}(\mathbf{v}_j - \mathbf{v}_i - \mathbf{g}\Delta t_{ij}) + \delta\mathbf{v}_{ij}, \tag{8}$$

$$\Delta\mathbf{p}_{ij} = \mathbf{R}_i^{\mathsf{T}}(\mathbf{p}_j - \mathbf{p}_i - \mathbf{v}_i\Delta t_{ij} - \frac{1}{2}\mathbf{g}\Delta t_{ij}^2) + \delta\mathbf{p}_{ij}. \tag{9}$$

Consequently, the preintegrated IMU measurement $\Delta\mathbf{R}_{ij}, \Delta\mathbf{v}_{ij}, \Delta\mathbf{p}_{ij}$ is propagated to estimate the initial guess of the scan match between consecutive LiDAR frames. Combined with other factors, the IMU bias is optimized in the constructed factor graph proposed in [25].

2) LiDAR-based motion estimation: Based on the initial guess obtained from the results of IMU pre-integration, vehicle odometry can be accurately estimated through scan matching using the point cloud results from LiDAR scans. The method of point cloud scan matching has been proposed as a computationally efficient and robust LOAM-based method [26]–[28]. However, utilizing multiple LiDARs brings additional effort to the LOAM-based method, which is not suitable for autonomous vehicles that use multiple sensors to minimize blind spots. Hence, the proposed method utilizes multiple LiDAR concurrently, helping overcome the underestimation of ground and building point clouds, enabling accurate motion estimation between successive LiDAR frames.

However, multiple LiDARs are asynchronized and exhibit motion distortion, as illustrated in Fig.3, which can cause a large drift during the acquisition of LiDAR odometry. Firstly, to solve the motion distortion of rotating LiDAR, we performed point cloud deskew through the short-term IMU motion estimation proposed by Gentil et al. [29]. After

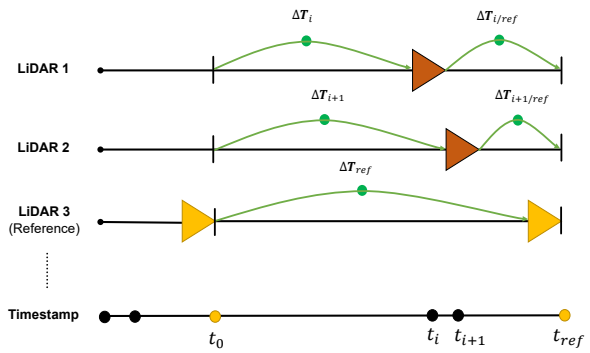


Fig. 3. Asynchronized multi-LiDAR system output point cloud at a different position. Each point cloud needs to be converted to reference LiDAR time t_i, t_j. The undistorted point cloud can be transformed to reference LiDAR position using pre-integrated IMU measurement between sub-LiDAR and reference LiDAR.

the deskew process, to merge the point cloud from each different LiDAR to the reference LiDAR frame, we derived the transformation from the reference LiDAR time t_{ref} to each LiDAR time t_i through imu-preintegration as we proposed in Section III-A.1. Each undistorted 3D point set of i-th LiDAR $\mathbf{C}_i = \{\mathbf{c}_0, \cdots, \mathbf{c}_n \mid \mathbf{c}_n \in \mathbb{R}^3\}$ is transformed to the point set generated from reference LiDAR $\mathbf{C}_{ref}$ using pre-integrated IMU measurements by

$$\begin{bmatrix} \mathbf{c}_{i_{ref}} \\ \mathbf{1} \end{bmatrix} = \mathbf{T}_{t_i/t_{ref}} \mathbf{T}_{i/ref} \begin{bmatrix} \mathbf{c}_i \\ \mathbf{1} \end{bmatrix}. \tag{10}$$

The transformation $\mathbf{T}_{t_i/t_{ref}} \in SE(3)$ is derived from odometry differences of pre-integrated IMU measurement between the reference LiDAR's time t_{ref} and i-th LiDAR's time t_i, whereas body transformation $\mathbf{T}_{i/ref}$ is the relative extrinsic transformation between the reference LiDAR and the i-th LiDAR.

Consequently, we could acquire point clouds with less motion-distorted points and rich spatial information fused with other LiDARs. Using a combined point cloud, we can accurately estimate the LiDAR odometry factors using a scan matching between consecutive reference LiDAR frames. For accurate LiDAR odometry estimation, we employ fast and precise voxelized-GICP [30]. Voxelized-GICP efficiently calculates voxel distributions using the distribution of each point within the voxel, allowing the proposed method to obtain odometry differences between consecutive reference LiDAR frames.

3) Initial heading compensation: Although using pre-integrated IMU measurements and undistorted point clouds enhances the odometry estimation, relying on them alone can result in drift accumulation. Therefore, this study proposes a method to localize a 3D local map on the geo-referenced, publicly available 2D map. Using a public 2D map, vehicles can localize in global coordinates and minimize odometry drift.

However, matching a 3D local map to a 2D map directly is highly ambiguous due to the difference in coordinate systems (i.e., local coordinates referenced by 3D local maps and UTM coordinates referenced by 2D maps). To address this, an initial alignment of coordinate systems is proposed using the IMU's magnetometer sensor to determine the north heading, similar to UTM coordinates. However, the alignment is imperfect due to the discrepancy between the magnetic north and the actual north pole (i.e., magnetic declination) and errors caused by Earth's ellipsoid shape during projection to a 2D coordinate (i.e., grid convergence).

To solve the heading ambiguity between both map coordinates, we initially aligned the 3D local map heading to the UTM coordinate heading before generating the 3D local map. We denote the magnetic north angle from the IMU as θ_{mag} and the true north angle as θ_n. The proposed method calculates the difference between the UTM north and magnetic angle using the World Magnetic Model (WMM) $\Theta(x, y)$ [31] at the vehicle's global position described by the UTM coordinates x, y, and the true grid north $\Gamma(x, y)$ [32] by

$$\theta_{md} = \theta_n - \theta_{mag} - \Theta(x, y), \tag{11}$$
$$\theta_{gc} = \theta_n - \Gamma(x, y), \tag{12}$$
$$\theta_{gm} = \theta_{md} - \theta_{gc}, \tag{13}$$

where θ_{md} is the magnetic declination, θ_{gc} is the grid convergence, and θ_{gm} is the grid magnetic angle difference. Consequently, an initial map can be generated using the initial vehicle heading θ_{gm}.

4) Global Localization: Through the short-term factor graph SLAM, the proposed method can form a local map aligned with the UTM coordinate system. Subsequently, through the alignment of the 3D local map and the publicly available map, the accumulated drift can be corrected in the global coordinates using the correspondence between the two images. Through these correction factors, the vehicle continuously updates the 3-DoF position information. In the following section, we explain how the proposed method can derive a global position from Section III-B to III-C.

B. Place Recognition

Humans can recognize their location on a publicly available two-dimensional (2D) vector graph map by matching the surrounding building outlines and road shapes, even with a significant initial position error. Should this approach be successfully applied to autonomous vehicles, it could be lighter and more effective than comparing the whole spatial data of an HD map. Therefore, inspired by human place recognition in [18]–[20], this study proposes a place recognition method that uses building outlines and road shapes. To utilize semantic information like humans, the proposed method compresses the temporary 3D local map, built from the sequence in Section III-A, and retrieves it to public 2D map images using a deep-learning-based place recognition framework trained for building outline and road shape matching.

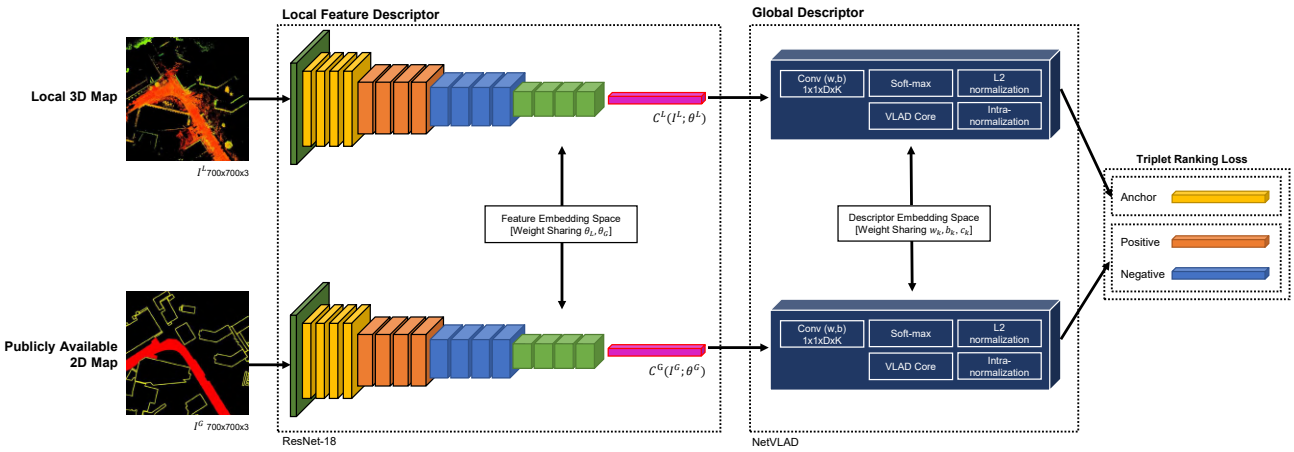


Fig. 4. The proposed Siamese-Network structure. Each network trains public 2D map images and 3D local map images separately and feeds them to the NetVLAD layer to generate a global descriptor. For training common features between 2D maps and 3D maps, the proposed network uses the same feature embedding space to share the same network weight of both networks.

1) Local map pre-processing: To align the 3D local map with the vector-based public map from a consistent viewpoint, the local map point cloud transformed an image, simulating an aerial perspective. The transformation process from a 3D local map to an image is initiated with a point cloud pre-processing step. Initially, a local 3D map, comprising an accumulated point cloud set denoted as $\mathbf{C}_{local} = \{\mathbf{c}_0, \cdots, \mathbf{c}_m \mid \mathbf{c}_m \in \mathbb{R}^3\}$, was cropped to a map size γ centered around the vehicle position $\mathbf{p} = [x, y]$. This conversion facilitated the transformation of the 3D local map into a top-view image.

$$R = \frac{\gamma \times 2}{W}, \tag{14}$$

$$I_u = \frac{\mathbf{c}_i(y) + \mathbf{p}(x)}{R}, \tag{15}$$

$$I_v = \frac{\mathbf{c}_i(x) - \mathbf{p}(y)}{R}, \tag{16}$$

$$I_i = H(\mathbf{c}_i(z)), \tag{17}$$

where W is image width and height corresponding public 2D map image, and γ is the metric size of the 3D local map. I_u, I_v are the image coordinates corresponding to the 3D local map, and I_i is the pixel intensity value from function H by

$$H(z) = \begin{Bmatrix} 0.5, z < 0 \\ 1, z \geq \delta \end{Bmatrix}, \tag{18}$$

δ signifies the height from the ground to vehicle height, which aids in distinguishing the ground point cloud.

2) Map retrieval network: The vehicle has precisely accumulated spatial information around the vehicle through the proposed spatial data interpretation. Leveraging this information, place recognition using a Siamese network is proposed, enabling accurate global localization with only onboard sensors and public map data without the need for additional sensors like GPS. This allows for rapid initial position estimation of the vehicle and enables precise vehicle localization without the use of expensive GPS sensors.

Although the 3D local map is converted to a top-view image that is similar to a publicly available 2D map, they cannot be compared, considering both images are generated using a different method that local 3D map images are generated using a point cloud while publicly available 2D maps are generated using vector graphics. However, humans can find and match common landmarks between real-world depth information and 2D vector maps, even if their intrinsic properties are different. Similar to humans, the proposed method structured a deep-learning network to find common landmarks between 3D local map images and public 2D vector map images and retrieve both images step-by-step. First, the Siamese network structure is proposed to learn public 2D map images and 3D local map images separately, which share a common feature space, thereby enabling the network to learn the same map characteristics, such as building outlines or road shapes. Then, the global descriptor layer extracts generalized global features from both images to retrieve the local map from the publicly available 2D map image database. Fig. 4 shows the overall network structure.

First, to identify features of the local map image and the public 2D vector map image, a deep learning network is structured to extract features of each image. Firstly, the 3D Local map image I^L and 2D public map image I^G are propagated through their respective feature extractors ξ^L, ξ^G, which then form descriptors composed of important local features such as roads or buildings. As a result, each feature extractor ξ^L, ξ^G can extract important information from both the 3D local map and the 2D public vector map, forming a local image feature descriptor. However, due to the different intrinsic properties of each image, each feature expression obtained from the two feature extractors is different even though their semantic information is the same.

The proposed method addresses this challenge by establishing a shared, common feature embedding space for both feature extractors. Previous camera image-based place recognition networks [33], [34] did not account for the differences between heterogeneous sensors, thus only enabling feature extraction from a single sensor using a single network. Moreover, Tang et al. [16] introduced map matching with different sensors using a deep learning network, but it relied on extensive preprocessing, including GAN-based transformations from camera images to LiDAR images before feature extraction using an additional CNN network. We structure the Siamese network with weight sharing to extract common features between heterogeneous image representations without the need for intensive preprocessing. This approach enables the finding and matching of common features between the semantic information (e.g., buildings and roads) of the public 2D map and the 3D local map by simply sharing the weight of each network. This facilitates information sharing between the networks, resulting in the formation of a shared local feature descriptor. The influence of the proposed common feature space will be analyzed in Section VII-B.

Consequently, the network can focus on overlapping regions of building outlines and road shapes from the public vector map images and the aerial view images generated from LIDAR data. So, proposed network can output common feature descriptor $F^L = \xi^L(I^L;\theta^L), F^G = \xi^G(I^G;\theta^G)$ with $H \times W \times D$ size feature map from each network, either where θ^L, θ^G are the network weight parameter. To formulate a common feature space (i.e., feature embedding space) for both networks, the same weight $\theta^L = \theta^G$ is updated during training both networks using the optimizer. As a result, our network can output a common feature map between the public 2D and local 3D map images from the local feature extractor's last convolution layer.

However, extracted feature maps contain countless feature descriptors in order less manner, which is not appropriate for comparing the overall correspondences between 3D local map images and public 2D map images directly. To train a common vocabulary capable of representing both 2D public map images and 3D local map images in the proposed network in an ordered and countable manner, an additional pooling layer was added to transform local feature descriptors into global descriptors. This layer enables the conversion of local information into a representation that aligns with the global context, facilitating the learning of a shared vocabulary for both types of map images. Furthermore, to train the common global features, a trainable global descriptor NetVLAD [35] inspired by the Vector of locally aggregated descriptors (VLAD) is attached at the last convolution layer. NetVLAD can learn better vocabulary to express global image features more flexibly than other handcrafted bag-of-visual words. To learn the proper vocabulary of both public 2D and local 3D map images each other, the NetVLAD layer is also structured as a Siamese structure to share a common vocabulary space (i.e., descriptor embedding space). First, the proposed method converts $H \times W \times D$ feature map to N-dimensional D-size descriptor. With a given set of local features from each network output $F^L = \left\{f_1^L, \cdots, f_i^L\right\}, F^G = \left\{f_1^G, \cdots, f_i^G\right\}$, the VLAD vector can be obtained as

$$V(j,k) = \sum_{i=1}^{N} \alpha_k(\mathbf{f}_i)(f_i(j) - \mu_k(j)), \tag{19}$$

where $f_i(j)$ is the j-th dimension of i-th descriptor , $\mathbf{f}_i$ is the group set of descriptor f, and $\mu_k(j)$ is j-th dimensions of the k-th cluster center work as vocabulary word. The hard-assignment function $\alpha_k(\mathbf{f}_i)$ denotes the correspondence of the descriptor to k-th cluster center. However, because $\alpha_k(\mathbf{f}_i)$ is not differentiable, the NetVLAD layer uses a soft-assignment function

$$V(j,k) = \sum_{i=1}^{N} \frac{e^{\mathbf{w}_k^T \mathbf{f}_i + b_k}}{\sum_{\hat{k}} e^{\mathbf{w}_{\hat{k}}^T \mathbf{f}_i + b_k}} (f_i(j) - \mu_k(j)), \tag{20}$$

where $\mathbf{w}_k, b_k, \boldsymbol{\mu}_k$ become trainable parameters. Using the trainable parameters, our network can be flexibly trained to learn better cluster centers for local map images and publicly available 2D map images. In addition, sharing the NetVLAD weight $\mathbf{w}_k, b_k, \boldsymbol{\mu}_k$ in the descriptor embedding space enables a global descriptor to represent images with the same vocabulary. The final VLAD descriptor can be derived by concatenating the 2D descriptor $V(j,k)$ to $[1, j \times k]$ size 1-D descriptor $V(I)$.

To train our network to perform place recognition tasks, the proposed network needs to retrieve the query 3D local map image I^L from publicly available 2D map images at the closest location. Using the distance relationship between the query and the retrieved image, the proposed network adapts the triple ranking loss to train our network. If the retrieved image is far from the query image location, the image is denoted as a negative image. Conversely, if the retrieved image is close to the query image, it is denoted as a positive image. Let us denote the query image as I^L, potential positive image I_p^G, and definite negative image I_n^G. To train our network to perform image retrieval tasks, the loss function attempted to minimize the Euclidean distance $d(I^L, I_p^G)$ from the query image location to the positive image pair and maximize the Euclidean distance of the negative image pair distance $d(I^L, I_n^G)$. Using the triplet image pair $\left\{I^L, I_p^G, I_n^G\right\}$, the triplet ranking loss can be calculated as

$$\begin{aligned}\mathcal{L}_{triple} = \sum_j \max(0, m + \left\|\xi^L(I^L;\theta^L) - \xi^G(I_p^L;\theta^G)\right\| \\ - \left\|\xi^L(I^L;\theta^L) - \xi^G(I_n^L;\theta^G)\right\|),\end{aligned} \tag{21}$$

where m is the constant margin parameter of the triplet ranking loss. The distance is calculated from the triplet tuple using the center of the map $\mathbf{p} = [x, y]$ in (15), (16) described by the UTM coordinates. We selected a potentially positive image close to the query image position distance d_N within 25m. The network was trained using the parameter θ of ξ^L, ξ^G from the ADAM optimizer [3] to ensure that the loss converges gently to the global minimum of our network.

C. Precise Localization

This subsection explains how the proposed method improves localization accuracy after the place recognition step. From the previous step, the proposed method can retrieve the query 3D local map image from publicly available 2D map images with the closest distance. However, the methods proposed thus far only provide sub-meter-level position estimation, which is still capable of following different roads or arriving at different destinations.

To remove outliers and improve the accuracy of enabling the autonomous vehicle to estimate its position precisely at the metric level, a particle filter combined with a learned place recognition network was designed to reject the outliers of the place recognition output. Particle filter combined with learned place recognition methods [9], [36], [37] has shown place recognition and odometry estimation ability; however, there are still limitations that hinder achieving accurate localization at the decimeter level. The combined particle filter needs to update weight only with the global feature descriptor distance between the prior map and query input point cloud, which does not contain structural information of the surrounding environment. Thus, in this work, we use the structure information to achieve accurate localization by using template matching to derive the relative position between the 3D local map image and the publicly available 2D map image.

1) Particle Filter: The proposed place-recognition method can recognize the position of the vehicle in a never-visited place. However, the place recognition result still exhibits a large localization error derived from the potential positive decision parameter $d_N = 25$m. Similar to the proposed network, humans cannot precisely localize their position by simply using the surrounding building outline and road shape. Thus, humans improve their localization accuracy by consistently matching their semantic information to a 2D map while walking along a path. Similarly, a precise localization sequence is implemented using a deep-learning network-weighted particle filter after place recognition to improve localization accuracy by constantly matching the surrounding environment. From our particle filter, the proposed method can estimate the vehicle position by considering vehicle odometry and rejecting outliers efficiently.

First, the proposed method initializes the particle filter using the UTM coordinates of the top-80 matched 2D map images from our network and matches local map images to public map images using top-1 weighted particles. In contrast to common particle filters that generate particles for the entire map road network for place recognition, the proposed particle filter only needs a small set of particles from our place-recognition network. Because the vehicle is assumed to move on a 2D planar surface, the position of the particle P_i^t at time t can be predicted by using a 2D motion model and LiDAR odometry input u in 2D space as

$$\mathrm{P}_i^t = head2tail(\mathrm{P}_i^{t-1}, u), u = [x, y, \phi], \tag{22}$$

where $head2tail$ is defined by

$$\mathrm{P}_i^t(x) = x \cdot cos(\phi) - y \cdot sin(\phi) + \mathrm{P}_i^{t-1}(x), \tag{23}$$

$$\mathrm{P}_i^t(y) = x \cdot sin(\phi) + y \cdot cos(\phi) + \mathrm{P}_i^{t-1}(y), \tag{24}$$

$$\mathrm{P}_i^t(\phi) = \mathrm{P}_i^{t-1}(\phi) + \phi. \tag{25}$$

After the prediction step, the public 2D map image candidates are rearranged from the predicted particle's UTM coordinates to reject outliers from the network output and predict potential candidates by considering the vehicle odometry. After the prediction step, the particle weight is updated as

$$w_i^t = \frac{w_i^{t-1} \cdot net(I^L, I^{\mathrm{P}_i})}{\sum_{i=1}^{N} w_i^{t-1} \cdot net(I^L, I^{\mathrm{P}_i})}, \tag{26}$$

where the weight function $net(I^L, I^{\mathrm{P}_i})$ calculates the L2 distance between the query image and the 2D map images from the output of our place recognition network. The L2 distance of the VLAD descriptor can be calculated as

$$net(I^L, I^{\mathrm{P}_i}) = \frac{\max(D(I^L, I^{\mathrm{P}_i})) - D(I^L, I^{\mathrm{P}_i})}{\max(D(I^L, I^{\mathrm{P}_i})) - \min(D(I^L, I^{\mathrm{P}_i}))}, \tag{27}$$

$$D(I^L, I^{\mathrm{P}_i}) = \left\| V(I^{\mathrm{P}_i}) - V(I^L) \right\|. \tag{28}$$

To avoid particle insufficiency, particle resampling is conducted using the stochastic resampling method if the effective particles are below N_{eff} by

$$N_{eff} = \frac{1}{\sum_{i=1}^{N} (w_i)^2}. \tag{29}$$

As a result, our deep-learning network-weighted particle filter enables our place recognition network to focus on a high-potential public 2D map image that considers the vehicle trajectory.

2) Building Outline Matching: Although the proposed particle filter can improve the place recognition accuracy, the proposed deep learning network only knows the mutual distance of the image descriptors and not the relative position between the images. Therefore, a direct template-matching method is proposed for higher localization accuracy by using a building outline and road shape. Using the proposed matching method, vehicles can localize the decimeter-level accuracy to a maximum.

Because normal handcrafted feature-based matches [38]–[40] are sensitive to image exposure, noise, and camera type, these limitations cause a mismatch of features between the 3D local map generated using the point cloud and the public 2D map generated using vector graphics. Therefore, To determine the relative position between the local 3D map image and the public 2D map image using building and road information, an intensity-based template match, and voxelized-GICP algorithm are used. First, both images are approximately matched using the normalized cross-correlation (NCC)-based template match [41] method. The proposed method extracts the building point cloud from the 3D local map above the vehicle using extrinsic LiDAR and regenerates the 3D local map image $I^L(u, v)$ using

TABLE I

TRAINING DATASET CONFIGURATION FOR PLACE RECOGNITION SEQUENCE.

Dataset	Sequence	Length	3D Map Image	2D Map Image	Area	Publicly Available Map Source
Complex Urban Dataset	Urban 08	1.56 km	902	1804	Residential	NGII
	Urban 15	5.43 km	2640	9200	Urban, Residential	
	Urban 09	15.7 km	12914	14278	Urban, Residential	OpenStreetMap
Our In-house Dataset	City 01	2.73 km	2968	5936	Urban, Residential	NGII
	Campus 01	2.03 km	2521	2760	Campus	
	Campus 03	1.78 km	1315	2630	Campus	
	Residential 01	1.95 km	1510	3962	Residential	
	Residential 02	1.12 km	1981	2132	Residential	
	Residential 03	1.71 km	1066	3020	Residential	
Total		38.15 km	30211	50764		

the extracted building point cloud. Then, the relative pixel differences between the local 3D map images and public 2D map images can be acquired by

$$B(u,v) = \frac{\sum_{u',v'}(I^L(u',v') \cdot I^{P_i}(u+u', v+v'))^2}{\sqrt{\sum_{u',v'}(I^L(u',v')^2 \cdot I^{P_i}(u+u', v+v'))^2}}, \tag{30}$$

where I^{P_i} denotes the retrieved 2D map image. From the NCC derivation, corresponding template-matching coordinates u_d, v_d can be derived from the maximum output value $B(u,v)$. Additionally, zero padding to the 2D map image is added, which enables NCC-based template matching to conduct sliding-window-based searching. We selected a zero-padding size from our network potential positive distance threshold $d_N = 25$ m in (21). As a result, the proposed method can initially align both images at the smallest building pixel intensity difference $B(u,v)$.

However, NCC-based template matching can only align both images in the horizontal and vertical directions. To acquire accurate position differences considering heading differences, voxelized GICP-based pixel-to-pixel matching is additionally conducted to estimate the relative positions between both images. First, the building point clouds $\mathbf{C}^L, \mathbf{C}^{P_i}$ are extracted from both images. Because the point cloud is extracted from the image coordinates, the point cloud is converted to a metric scale by multiplying R annotated in (14) to $\mathbf{C}^L, \mathbf{C}^{P_i}$. Then, the transformation $\mathbf{T} \in SE(2)$ can be estimated using the voxelized-GICP in Section III-C. As a result, the UTM coordinates of the 3D local map image can be acquired using

$$\mathbf{U}^L = \mathbf{T} \cdot \mathbf{U}^{P_i}, \tag{31}$$

where $\mathbf{U}^{P_i}$ is the UTM coordinate of a public 2D map image with homogeneous coordinates (x,y,1). Finally, the estimated UTM coordinates $\mathbf{U}^L$ of the local 3D map image can be propagated to the spatial data interpretation sequence as a global localization measurement and accurately perform localization. Global localization measurement only propagated when odometry covariance becomes larger than map covariance shown in [8], [42], [43].

D. Implementation Detail

To implement the proposed method, experiments were conducted based on the parameters listed in Table III. For the network training, the same parameters (d_n, k, N, lr, m) in the NetVLAD [35] were adopted. To train the model according to LiDAR and 2D vector graphic representation, all layers from conv1 to conv5 of the ResNet18 were retrained. The consequence of crop map size parameters in the proposed method is discussed in Section VII-B.

IV. DATASETS

To evaluate the proposed method, residential areas, and complex urban areas are selected that have low priority for generating HD maps owing to their complexity, difficulty in updating in real-time, or existence in GPS-shadowed areas. Selected sequences are recorded in various regions and MMS setup included in publicly available datasets, *KITTI odometry* [44], *Complex Urban Dataset* [45], *MulRan* [46], and our in-house dataset. We leverage a comprehensive dataset spanning 81.7 km, encompassing diverse geographic regions, environmental conditions, and various publicly available maps. Further details about the dataset and adaptation for training and testing are elaborated in the subsequent sections.

A. Publicly Available Datasets

1) Complex Urban Dataset: The complex Urban dataset provides sensor data for large-scale urban areas for place recognition and odometry estimation accuracy. The MMS of a complex Urban dataset uses tilted two 16-ray LiDARs and IMU information, which is appropriate to test our proposed method with various sensor setups. For publicly available map sources, a map from the National Geographic Information Institute (NGII) is used.

2) KITTI Odometry: The KITTI odometry dataset is widely used to evaluate odometry accuracy and place recognition in autonomous vehicles. MMS of the KITTI dataset uses horizontally mounted 64-ray LiDAR and INS information for evaluation. Because the KITTI dataset area did not nationally provide a publicly available 2D map, OpenStreetMap was used as a publicly available map source.

TABLE II

TEST DATASET CONFIGURATION.

Dataset	Sequence	Length	Query Set (3D Map)	Database (2D Map)	Area	Publicly Available Map Source
MulRan	DCC 02 / Ours	4.68 km	1673 / 2730	5898	Urban	NGII
	KAIST 02	6.10 km	2359	4061	Campus	
Complex Urban Dataset	Urban 07	2.55 km	2074	4431	Residential	
	Urban 02	4.20 km	2243	7721		
	Urban 03	3.06 km	1882	7829		
KITTI	05	2.21 km	11074	2565	Residential	OpenStreetMap
In-house	Campus 03	1.12 km	1809	1104	Campus	NGII

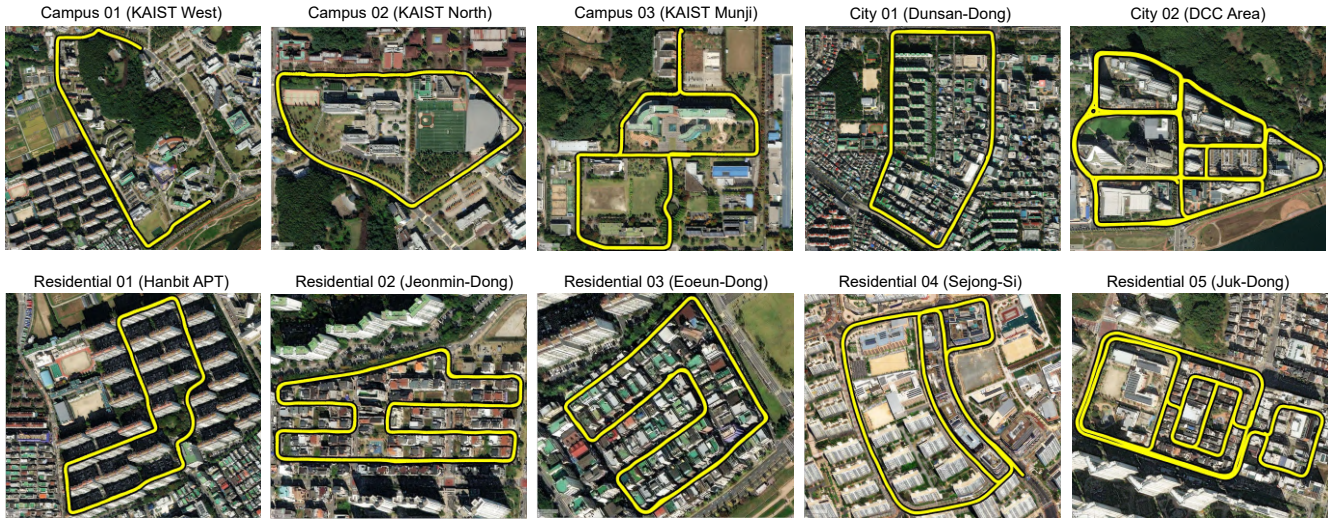


Fig. 5. Aerial image depicting the trajectory of an in-house dataset. We try to collect datasets that are hard to generate and update HD maps. Selected areas have characteristics with narrow roads, and high-rise buildings densely located far from the main street.

TABLE III

IMPLEMENTATION DETAILS

Implemented parameter details	
Crop map size (γ)	50 m (Residential area) / 80 m (Complex urban area)
Image size ($W \times H$)	700×700
Decision distance (d_n)	25 m
Cluster (k)	64
VLAD dimension (N)	512-D
Learning rate (lr)	0.0001
Epoch	40
Constant margin (m)	0.1

3) MulRan Dataset: The MulRan dataset provides small urban data for place recognition by radar and LiDAR. The MulRan dataset MMS uses horizontally mounted 64-ray LiDAR and IMU information. Unlike the KITTI and Complex Urban datasets, the `KAIST01`, `KAIST02` sequences contain less building information, and `DCC01`, `DCC02` sequences have abundant building information but contain atypical obstacles (e.g., parking cars, plants, trees, pedestrians, etc.). For publicly available map sources, a map from NGII is used.

4) Our In-house Dataset: In addition, we captured LiDAR and IMU datasets from our MMS to generate small-scale residential areas that were not contained in other datasets. Most of the captured area comprised apartment complexes or small-scale complex urban environments, which are relatively neglected when generating HD maps. For the dataset, triple 16-ray LiDAR and IMU information are used. To generate ground truth, an odometry-INS-integrated RTK-GPS system is used. The characteristics of the captured areas are shown in Fig. 5 as an aerial view, and the MMS setup used is described in Fig. 6. Additionally, a map from NGII is used as a publicly available map source.

B. Training Dataset

To generate aerial view images for training, *Complex Urban Dataset* and our in-house datasets were used. `Urban08`, `Urban15`, `Urban09` in *Complex Urban Dataset* and `City01`, `Campus01`, `Campus02`, `Residential01`, `Residential02`, `Residential03` were selected in our dataset, as shown in Fig. 5. The training sets comprised image pairs containing query 3D local map images and publicly available 2D map images. A public 2D map image was generated at 5 m intervals from the ground truth trajectory at the UTM location of the dataset. The training dataset comprised 30,211 3D local map images and 50,764 2D public map images split by 80% and 20% into training and validation

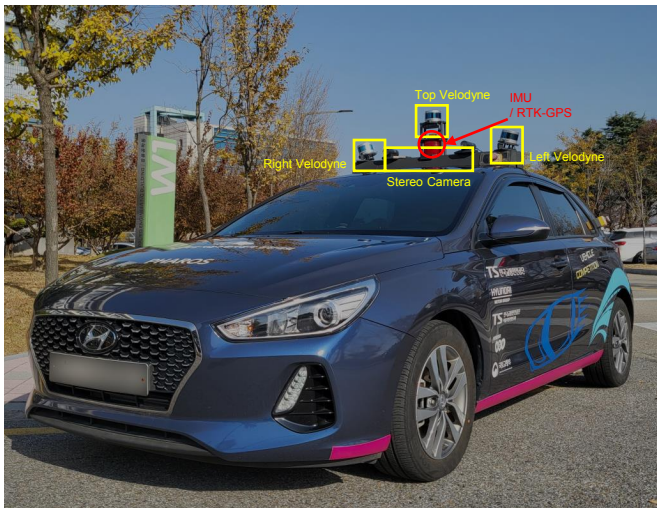


Fig. 6. In-house dataset MMS platform. Our platform contains triple 16-ray LiDAR (Velodyne VLP-16), MEMS-IMU (Xsens MTI-30G), stereo camera (FLIR BlackFly S) and RTK-Level INS integrated GPS (Ublox ZED-F9K).

sets, respectively. The training dataset configurations are summarized in Table I.

C. Test Dataset

Before estimating vehicle odometry in global coordinates, the vehicle must initially localize at global coordinates. Therefore, we conduct the place recognition accuracy evaluation of the proposed method through an additional test dataset. In our test dataset, we aimed to assess the generalization of the proposed place recognition method. Dataset included regions that did not overlap with the training dataset and encompassed the KITTI dataset, which differs in nationalities and publicly available map sources. The test dataset is summarized in Table II.

V. Evaluation Metric and Criteria

This section introduces the evaluation metrics for comparing the proposed method with other global localization methodologies using odometry estimation. In addition, an evaluation metric for place recognition accuracy was also provided to evaluate the global localization performance. The evaluation metrics and corresponding comparison targets are described below.

A. Odometry Estimation Evaluation Metric

To evaluate global odometry estimation during autonomous vehicle driving, the absolute trajectory error (ATE) was evaluated between the estimated odometry and ground truth odometry. Through evaluation, experiments are conducted to determine whether the proposed method can achieve similar performance to HD map-based odometry estimation using only vehicle onboard sensors and publicly available maps. The ATE provides an intuitive comparison result, that is, position, rotation, and velocity. The ATE compares the absolute distance between the subsequent odometry. The odometry consistency between the ground truth and measurement can be aligned by the Horn method [47], which derives a rigid relative body transformation between the ground truth and comparison results. As the specific time i is denoted for estimating the position error matrices,

$$E_i = Q_i^{-1} S P_i, \tag{32}$$

where S is the rigid body transformation, Q_t is the ground-truth position, and P_t is the estimated odometry result from our full framework. To evaluate the mean of the entire trajectory error between the ground truth and measurements,

$$\text{ATE}_{\text{rot}} = \frac{1}{n}\sum_{i=1}^{n} \angle(\mathbf{R}_i), \tag{33}$$

$$\text{ATE}_{\text{pos}} = \frac{1}{n}\sum_{i=1}^{n} p_i, \tag{34}$$

where $\angle\mathbf{R}_i$ is the rotation part $SO(3)$ in E_i which is converted into an angle-axis representation, and p_i is the position vector $\mathbb{R}^3$ in E_i.

B. Place Recognition Evaluation Metric

Before estimating vehicle odometry in global coordinates, the vehicle must initially localize at global coordinates. To evaluate the localization performance in global coordinates, Recall was evaluated for the place recognition evaluation metric used in [11], [35], [48]. Suppose at least one of the top N database images is located less than d_n from the ground-truth position of the query image. In that case, the image is considered to have been appropriately localized. The percentage of successfully identified queries was plotted for the various N values. The N value is selected as 1 to 80 for evaluation. Furthermore, the precision-recall curve and maximum F1 score are provided for Top-1 match cases to evaluate our proposed network.

C. Comparison Objective

To evaluate the global odometry estimation performance, the proposed method was compared with prior map-based and DGPS-based localization methods.

1) HD Map: As a reference, we used the common HD map-based localization method used in [50]. Using a reference HD map, the vehicle consistently matched the point cloud from the onboard LiDAR sensor to the reference map. For the scan match between the HD map and LiDAR scan, voxelized-GICP, the same scan-match algorithm used in Section III-A is adapted. Reference HD maps of KITTI were generated using a point cloud corresponding to the given ground-truth position. In MulRan and ComplexUrabanDataset, we evaluated accuracy using HD Map data in the LAS format provided as a ground truth by each dataset.

2) LIO-SAM: LIO-SAM [27] represents a methodology for estimating vehicle odometry utilizing LiDAR data and inertial information; this is achieved through the application of a LOAM-based scan-match algorithm and the integration of preintegrated inertial measurements, contributing to accurate odometry estimation. In the evaluation of LIO-SAM within global coordinates, supplemental DGPS (Differential

TABLE IV

ATEs (TRANSLATION AND ROTATION) OF GLOBAL ODOMETRY ESTIMATION RESULT.

Dataset	Sequence	Length	HD Map		LIO-SAM [27]		RangeMCL [49]		Ours	
			ATE_{pos}(m) (mean / median / std)	ATE_{rot}(deg) (mean / median / std)	ATE_{pos}(m) (mean / median / std)	ATE_{rot}(deg) (mean / median / std)	ATE_{pos}(m) (mean / median / std)	ATE_{rot}(deg) (mean / median / std)	ATE_{pos}(m) (mean / median / std)	ATE_{rot}(deg) (mean / median / std)
MulRan	DCC 02[1]	4.09 km	0.6480 / 0.6721 / 0.218	1.5336 / 1.5347 / 0.003	**1.5361 / 1.5260 / 0.708**	3.1320 / 3.1335 / 0.007	2.5923 / 2.3365 / 1.399	0.0578 / 0.0292 /0.097	1.7703 / 1.7211 / 0.774	**0.9958 / 1.0626 / 0.009**
	KAIST 02[1]	6.10 km	0.8541 / 0.7512 / 0.492	0.6367 / 0.4529 / 0.011	1.3839 / 1.2647 / 0.698	**0.0325 / 0.0293 / 0.014**	1.9436 / 1.7589 / 0.940	3.1127 / 3.1227 /0.037	**1.3823 / 1.2470 / 0.948**	**0.0162 / 0.0128 / 0.0133**
Complex Urban Dataset	Urban 07[1]	2.55 km	0.4936 / 0.4409 / 0.280	0.0143 / 0.0120 / 0.008	1.2371 / 0.9743 / 0.643	0.0197 / 0.0175 / 0.011	-	-	**1.1327 / 0.9183 / 0.660**	**0.0180 / 0.0171 / 0.007**
	Urban 02	4.20 km	0.2746 / 0.2572 / 0.202	0.5537 / 0.4713 / 0.007	3.2992 / 2.9517 / 4.443	**3.3380 / 3.0204 / 0.039**	-	-	**1.4626 / 1.3569 / 0.951**	3.6788 / 3.5927 / 0.020
	Urban 03	3.06 km	0.3054 / 0.2527 / 0.245	0.7703 / 0.6237 / 0.010	13.5201 / 8.5198 / 12.805	**0.1304 / 0.1115 / 0.064**	-	-	**1.6906 / 1.5083 / 1.961**	2.3875 / 2.1847 / 0.020
KITTI	00	2.05 km	0.1109 / 0.0847 / 0.091	0.3942 / 0.3029 / 0.033	-	-	-	-	**0.5796 / 0.5110 / 0.515**	**0.6007 / 0.3693 / 0.585**
	02	2.52 km	0.3623 / 0.2407 / 0.469	0.5901 / 0.4022 / 0.009	-	-	-	-	**5.9826 / 4.8641 / 5.102**	**9.4600 / 9.7348 / 0.046**
	04	0.01 km	0.2526 / 0.1258 / 0.556	0.1032 / 0.0647 / 0.001	-	-	0.9505 / 0.7004 / 1.055	2.0490 / 1.7654 / 1.196	**0.2837 / 0.1610 / 0.605**	**3.5654 / 3.5654 / 0.257**
	05	2.21 km	0.0777 / 0.0467 / 0.216	0.1475 / 0.0897 / 0.006	-	-	5.3390 / 4.6384 / 3.065	59.8047 / 59.2617 / 0.055	**0.7450 / 0.7058 / 0.416**	**0.6279 / 0.5586 / 0.007**
	06	1.23 km	0.1135 / 0.0873 / 0.139	0.1815 / 0.1580 / 0.002	-	-	12.0529 / 11.7213 / 6.555	56.7885 / 58.3031 / 0.118	**0.7481 / 0.7909 / 0.383**	**0.8137 / 0.6665 / 0.008**
	07	0.39 km	0.1784 / 0.1608 / 0.107	0.3603 / 0.3252 / 0.003	-	-	0.3539 / 0.2835 / 0.398	2.4488 / 2.4496 / 0.022	**0.1789 / 0.1624 / 0.135**	**2.1739 / 2.4857 / 0.017**
	08	2.05 km	0.1151 / 0.0979 / 0.089	2.7386 / 2.9339 / 0.004	-	-	-	-	**1.5473 / 1.3849 / 0.746**	**1.1436 / 0.9601 / 0.014**
	09	1.01 km	0.1586 / 0.1519 / 0.089	3.7079 / 3.6873 / 0.026	-	-	-	-	**0.9876 / 1.0259 / 0.526**	**1.4755 / 0.8966 / 0.030**
	10	0.79km	0.2738 / 0.2419 / 0.145	5.1048 / 5.1124 / 0.015	-	-	23.7221 / 17.0965 / 20.305	2.5005 / 2.4299 / 0.276	**0.6663 / 0.6531 / 0.319**	**5.4632 / 5.8072 / 0.019**
In-house	Campus 03	1.12 km	0.2931 / 0.2345 / 0.273	0.4308 / 0.2862 / 0.007	**0.5304 / 0.4601 / 0.325**	**0.0136 / 0.0115 / 0.009**	-	-	0.8562 / 0.7131 / 0.549	1.5369 / 1.2919 / 0.014
	Residential 04	2.52 km	0.2320 / 0.1406 / 0.388	0.5979 / 0.4791 / 0.009	0.7557 / 0.6695 / 0.461	**0.0233 / 0.0242 / 0.009**	-	-	**0.6399 / 0.5444 / 0.464**	1.3803 / 1.3045 / 0.011
	Residential 05	2.40 km	0.2269 / 0.1138 / 0.315	0.9713 / 0.7358 / 0.014	**0.6046 / 0.5548 / 0.338**	**0.0260 / 0.0160 / 0.021**	-	-	0.7942 / 0.7270 / 0.624	1.9863 / 1.5066 / 0.025

[1] Only these sequences use a prior map generated at the same trajectory with different times.
The bold values are the ATE for the position and rotation with the best performance in each sequence.

Global Positioning System) information is incorporated as a GPS prior factor. DGPS, widely employed in commercial vehicle navigation, offers accuracy at the decimeter-to-meter level. The implementation involved utilizing the provided code and parameter settings for LIO-SAM with enabled GPS factor propagation[1]. Furthermore, the comparison included loop closing, a form of place recognition aimed at global drift minimization.

3) RangeMCL: RangeMCL [49] exploits range images from LiDAR for the global localization and odometry estimation of vehicles. They used a prior mesh map as a reference and a single LiDAR scan to localize a vehicle. The same parameters were used provided[2] except for MulRan. In the *MulRan* dataset, the field-of-view parameter was changed specified in the ouster LiDAR sensor. During the evaluation, we only used place recognition via a particle filter for the `DCC 02`, `KAIST 02` dataset, and `Urban 07` sequence, which has prior data captured at the same place but at different times.

VI. Experimental Results and Evaluation

This section shows experiment results of odometry accuracy and place recognition accuracy conducted on publicly available datasets. First, we evaluate the odometry estimation accuracy compared with ground truth data to test whether the proposed method can estimate vehicle trajectory in global coordinates. Additionally, experiments on place recognition accuracy versus prior map-based place recognition methods are conducted to test the global localization accuracy of the proposed method.

A. Global Odometry Estimation Evaluation

The proposed method's odometry accuracy is evaluated with various sensor configurations and diverse environments. At `KAIST02`, `DCC02`, `Urban07` sequences, the evaluation is conducted on a prior map created at a different time than the test dataset. Other sequences are tested using a prior map created at the same time as the test dataset because datasets with the same location at different times are scarce. The proposed method was evaluated using ATE mean, median, and standard deviation for position and rotation; furthermore, it compared position error and pose uncertainty. Table IV shows the overall comparison result.

1) MulRan Dataset: At the `DCC 02` sequence, the proposed method can accurately localize the vehicle compared with other DGPS-based methods or prior map-based methods. Our accuracy can be established by consistently matching building information to the geo-referenced publicly available map. Further, even though publicly available map information can not be used if there is a lack of building, the proposed method can still estimate vehicle odometry using a combined LiDAR-based odometry estimation framework. At `KAIST02` sequence, our method can localize vehicles even more accurately, because the campus area has large openings with fewer obstructions. Contrarily, other methods have larger estimation errors than the proposed method even though they have prior information or DGPS as shown in Fig. 7. LIO-SAM aided by DGPS has a large error at a GPS-shadow area when high-rise buildings surround the vehicle but loop-closing can compensate drift over time. However, LIO-SAM must revisit the same place to reduce drift using loop-closing and needs to save the whole point cloud for the loop-closing, which can demand spacious computation space. RangeMCL also suffers from drift over time even if they use prior information. However, as shown in Fig. 8, the proposed method can accurately localize vehicles using the building outlines even though there are obstructions like surrounding plants or existence in high-rise buildings.

2) Complex Urban Dataset: In `Urban07` sequence, the proposed method can estimate vehicle position accurately even if there are differences in sensor setup or characteristics of the tested area. The proposed method has demonstrated its generalization ability to localize in complex urban datasets with a tilted-mounted sensor setup, showing a small difference compared to datasets with a horizontally mounted sensor setup, such as the MulRan dataset. Not even `Urban07`, our method can successfully localize vehicles more accurately than other methods by using the publicly

[1] https://github.com/TixiaoShan/LIO-SAM
[2] https://github.com/PRBonn/range-mcl

available map in `Urban02` and `Urban03` sequence. LIO-SAM with the DGPS method exhibits high accuracy when a vehicle drives into a large opening area. However, when entering complex urban areas, it suffers from the multi-path problem and tends to diverge. GPS challenging situation can be seen in Fig. 9 that suffers from the multi-path problem, but the proposed method demonstrates that can estimate vehicle trajectory similar to reference HD map trajectory. The RangeMCL method, based on depth map comparisons, faces challenges when employed with tilted LiDAR setups. This is largely owing to the LiDAR configuration in the Complex Urban dataset being primarily designed to detect the ground, offering limited coverage for depth information on the surrounding environment.

3) KITTI Dataset: In contrast to other datasets, in the *KITTI* dataset, OpenStreetMap was used for a publicly available map. Additionally, KITTI does not provide commercial-level GPS data and synchronized IMU data; therefore, the DGPS-aided LIO-SAM method has not been evaluated, and the proposed method did not use IMU-preintegration for deskew point cloud or initial guess estimation for scan match. Despite using different publicly available map formats and outages of IMU information for point cloud deskew and scan match initial guess, the proposed method can accurately localize vehicles in never-visited places or different nations. Most of the *KITTI* dataset sequences are captured in residential areas that have unique building outline patterns and less noise (e.g., plants, moving vehicles, and pedestrians); hence,

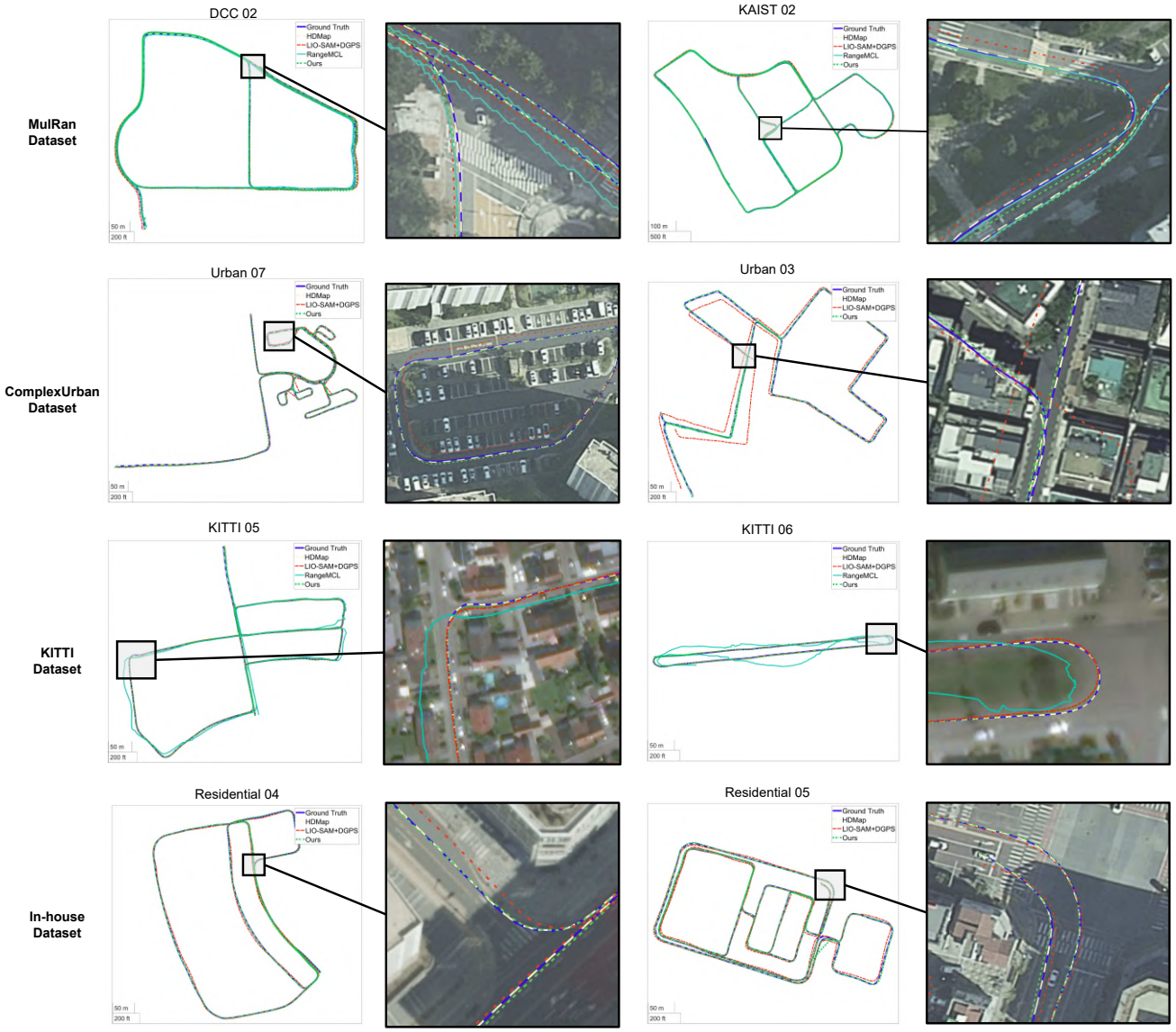


Fig. 7. 3D local map image and the corresponding publicly available map. Place recognition network retrieves 3D local map to public map image which has closest image descriptor. Using template match, the proposed method can derive the relative position between the public map image coordinate and the 3D local map image.

TABLE V

ATE(TRANSLATION) COMPARISON WITH OTHER PUBLICLY AVAILABLE SOURCE-BASED METHODS.

Method	KITTI Odometry Sequence									
	00	01	02	04	05	06	07	08	09	10
RangeMCL [49]	×	×	×	1.17 m	5.34 m	12.06 m	0.35 m	×	×	23.72 m
Lost! [12]	1.8 m	**2.5 m**	2.2 m	-	2.7 m	-	1.5 m	2.0 m	3.8 m	2.5 m
OpenStreetSLAM [10]	> 10 m	-	> 20 m	-	-	-	-	-	-	-
AGCV-LOAM [36]		-	-	-	-	-	-	4.45 m	-	-
Yan et al. [9]	> 10 m	-	-	-	> 10 m	> 10 m	> 10 m	-	> 10 m	> 10 m
Miller et al. [51]	2.0 m	-	9.1 m	-	-	-	-	-	7.2 m	-
Tang et al. [15]	-	-	3.7 m	-	-	-	-	-	-	-
Fervers et al. [17]	×	2.53 m	**1.42 m**	0.66 m	0.77 m	**0.57 m**	0.85 m	2.51 m	×	0.96 m
Ours	**0.58 m**	×	5.98 m	**0.28 m**	**0.75 m**	0.75 m	**0.18 m**	**1.55 m**	**1.00 m**	**0.67 m**

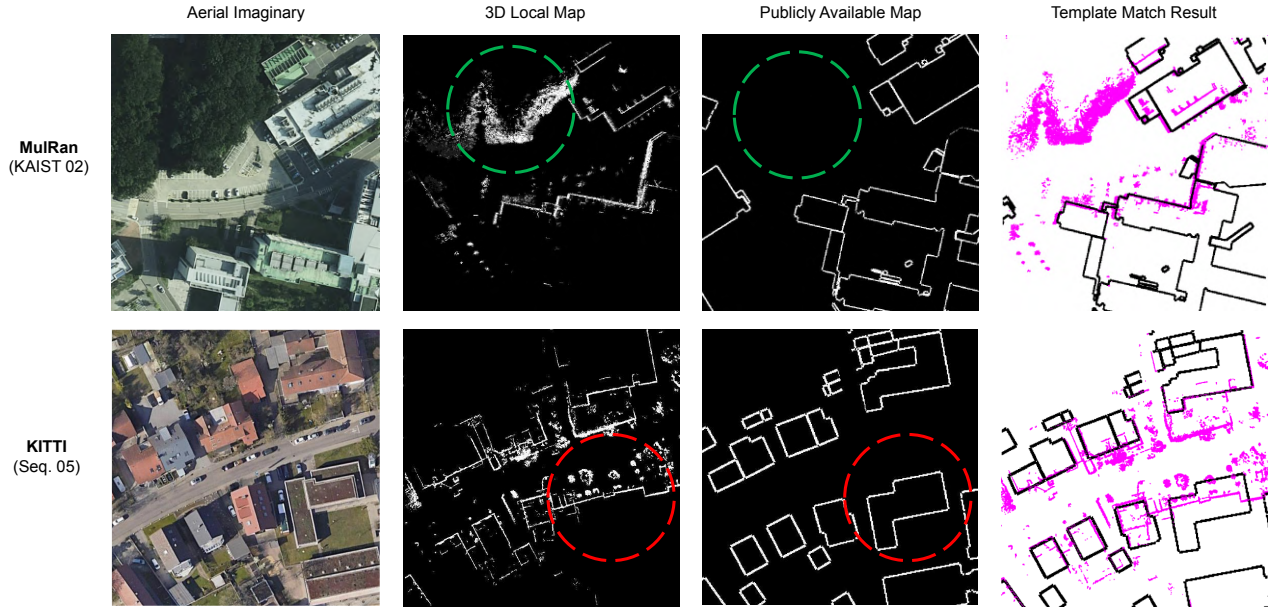


Fig. 8. Proposed method template match result. Top: In `KAIST 02` sequence, the proposed method can retrieve a 3D local map image even if there are large plant areas (Green circled area); Bottom: In the KITTI sequence `05` dataset, the proposed method can successfully match with another building outline although the newly updated publicly available map is inconsistent with the 3D local map (Red circled area).

the proposed method can achieve more accurate odometry estimation results than other methods listed in Table IV. However, RangeMCL has large localization errors despite having a prior map.

Also, based on the all odometry sequences within the KITTI dataset, which includes various road conditions and complex terrains, evaluations are conducted along with other public map sources-based metric localization methods. Methods are evaluated using the mean ATE of position and summarized results in Table V. Cells for methods that did not provide results for other sequences were left blank. Sections marked with '×' represent sequences where odometry estimation failed. The proposed method achieved state-of-the-art performance in a total of 7 out of 10 sequences. Unlike vision-based methods [12] with errors surpassing sub-meter levels, and resource-intensive aerial view-based approaches or methods relying on precise initial DGPS guesses [15], [17], [36], the proposed approach attains accurate decimeter-level localization using only publicly available vector maps and onboard sensors. However, in sequences like `KITTI 01` where buildings were entirely absent, or in `KITTI 02` sequence where building facades have low visibility, global localization was not possible. The limitation of the proposed method will be analyzed in Section VII-A.4.

4) Our In-house Dataset: In our in-house dataset, the proposed method achieved precise odometry estimation for all the sequences. Unlike other datasets, the in-house dataset was mainly recorded in downtown areas, where buildings were clearly visible. In addition, the LiDAR configuration of the dataset can capture abundant information about surrounding vehicles. As a result, the proposed method can achieve decimeter-level localization accuracy at a maximum

Fig. 9. Comparison result at `Urban 03` between HD map, proposed method, and virtual reference station (VRS)-GPS. VRS-GPS has higher accuracy (up to cm-level) than DGPS; however, it still suffers from signal divergence owing to surrounding buildings. Nonetheless, the proposed method can localize vehicles more accurately similar to HD map-based methods.

comparable to the HD map-based approach. As shown in Fig. 7, our trajectory performs similarly to the HD map-based approach, which can localize vehicles more accurately than other methods. LIO-SAM could provide relatively accurate odometry estimation from loop-closing. Still, LIO-SAM requires revisiting the same place to minimize drift through loop-closing, necessitating the storage of the entire point cloud for this process. RangeMCL diverges over sub-meter on in-house datasets.

B. Place Recognition Evaluation

We showed that the proposed method can estimate the vehicle position accurately in the full framework. However, our place recognition sequences should proceed before the precise localization sequence. Therefore, a further evaluation is conducted wherein the proposed place recognition network can successfully retrieve 3D local map images to 2D map images. Our place recognition network is compared with other place recognition methods, ScanContext, PointNetVLAD, Y.Cho et al., and original NetVLAD. We select ScanContext with 50 candidates in KD-Tree[3] and train PointNetVLAD[4] and original NetVLAD with the same parameter author use and trained with the same training dataset in Table I as proposed method.

1) Same Sensor Configuration: First, the proposed network is evaluated using the same sensor configuration captured at different times. As shown in Fig. 10-(a), the proposed method recognizes a place with higher accuracy than prior map-based place recognition. The proposed method demonstrates accurate global localization using abundant buildings and road geometry features similar to those of humans. Consequently, the proposed method can successfully perform place recognition tasks with a public 2D map image database, even for vehicles that have never visited before. Conversely, our previous approach [11] shows less accurate results, considering it requires a perfectly segmented building point cloud and a horizontally mounted LiDAR configuration. ScanContext requires preprocessed point clouds, and vehicles must be visited at least once to utilize prior information even if their method demonstrates better F1-score in `KAIST 02` and `DCC 02` sequences, as shown in Table VI.

2) Different Sensor Configuration: The place recognition accuracy at the same place is evaluated but with different sensor configurations captured at different times because not all autonomous vehicles have the same sensor specifications or mounting positions. To acquire a dataset with different sensor setups, the spatial data from the same MMS is recorded in Fig. 6. From our MMS, LiDAR, IMU, and ground truth positions are collected in the overlapping area corresponding to the *MulRan* dataset `DCC 02` sequence.

From the different datasets, experiments were conducted on the `DCC 02` sequence. The accuracy results in Fig. 10-(d) demonstrate that the proposed methods can achieve robust place recognition performance compared to other prior map-based methods, even with different LiDAR resolutions and mounted angles. In addition, the proposed method can perform place recognition tasks successfully, even in the absence of precise prior 3D information or different sensor configurations. As a result, the proposed method can solve the place recognition problem in large-scale urban or residential areas by using only publicly available 2D maps, building outlines, and road shapes from onboard sensors. In contrast, other prior 3D information-based methods have inconsistent accuracies and large degradation in F1-Score when the sensor configuration differs, as shown in Table VI and

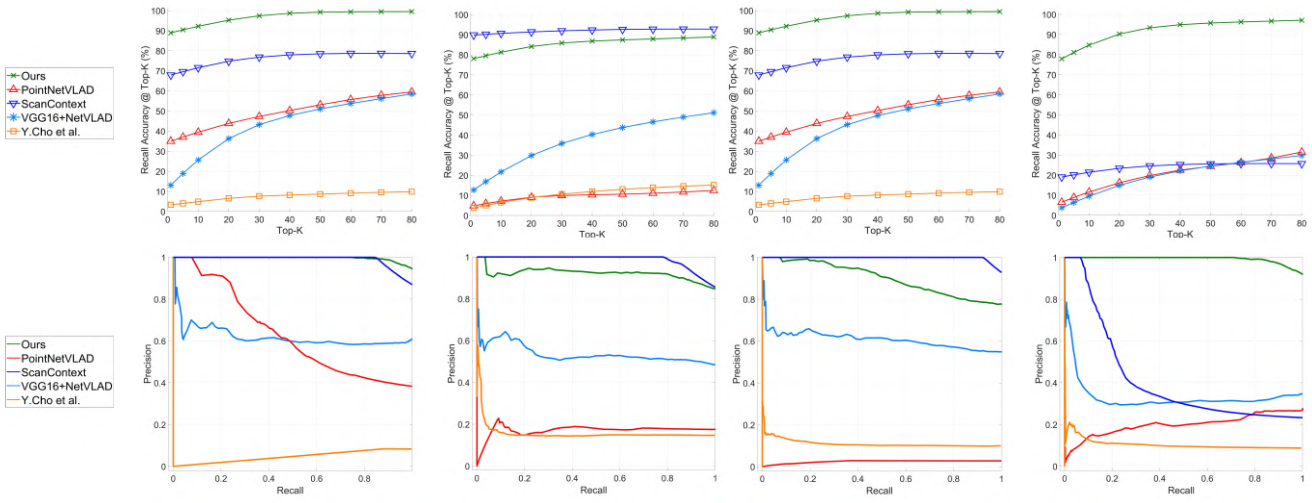
TABLE VI
MAX F1-SCORE OF EACH SEQUENCE.

Max F1-Score				
Methods	Datasets			
	Urban 07	KAIST 02	DCC 02	DCC 02 (In-house)
PointNetVLAD [48]	0.5542	0.0539	0.3000	0.4363
ScanContext [52]	0.9299	**0.9624**	**0.9220**	0.3776
VGG16+NetVLAD [35]	0.7594	0.7094	0.6522	0.5159
Y.Cho et al. [11]	0.1532	0.1810	0.2563	0.1607
Ours	**0.9717**	0.8745	0.9178	**0.9592**

TABLE VII
COMPUTATIONAL TIME EVALUATION. (`KAIST 02`)

Processing time (ms)	
Local map generation	77.65 ms
Local map image conversion	50.32 ms
Network-based place recognition	14.15 ms
Local map-public map template matching	2.13 ms
Total processing time	144.25 ms

[3]https://github.com/irapkaist/scancontext
[4]https://github.com/mikacuy/pointnetvlad



(a) Urban 07 (b) KAIST 02 (c) DCC 02 (d) DCC 02 (In-house MMS)

Fig. 10. Recall@N and Precision-Recall curve comparison result. Both (a) and (b) are evaluated with the same sensor configuration as the prior map captured at different times. (c) and (d) are evaluated with different sensor configurations, and the results demonstrate that the proposed method can recognize a place even when the sensor configuration is changed. Y.Cho et al. method is neglected on DCC 02 area because of low accuracy under 5% recall accuracy.

Fig.10. LiDAR descriptor-based methods exhibit weaknesses when their LiDAR configurations are mutually different. Furthermore, they must visit at least once and generate precise spatial information to utilize high-resolution prior information, which requires a large computing space and cannot be updated in real-time owing to its accessibility.

VII. Discussion

This section discusses the limitations of the proposed method and conducts an ablation study for further analysis of our network.

A. Limitation

1) Computational Cost: Since the computing resources for autonomous vehicles are limited, computing time for the proposed method has been evaluated for further analysis. We conducted a computational time evaluation on a PC equipped with an Intel i7-9700K CPU, 64GB of RAM, and a GeForce RTX 2070 GPU for our deep-learning network. The average computational time for each level of the proposed method is listed in Table VII. Considering the scan interval of LiDAR is 10Hz, the proposed method has limitations in localizing vehicles in real-time, where the process interval is 7Hz; this is due to the significant time required to correspond numerous point cloud data to individual image pixels during the local map-to-image conversion process, resulting in a bottleneck. Therefore, further code optimization for efficient image processing of numerous point clouds should be carried out as future work.

2) Operational Design Domain: The proposed method aimed to address the localization problem of autonomous vehicles using 2D vector graphics-based publicly available maps. The proposed method achieved precise vehicle localization in diverse urban settings with various buildings and complex road scenarios. However, due to the limitations mentioned above, it was unable to perform localization in areas with no surrounding buildings or in environments with repetitive patterns, such as tunnels or forested paths. Therefore, we specify the Operational Design Domain (ODD) under which the proposed method can be used, delineating the environments in which the proposed method is applicable in VIII. Since global localization is not applicable in areas with no building information, the proposed method can enable the localization of autonomous vehicles in urban or rural areas where buildings are densely located. In addition, the proposed method can be applied to urban driving speeds up to 50 km/h [53], and since LiDAR is robust to illumination changes, localization can be performed in all but foggy conditions.

TABLE VIII
Operation Design Domain.

Operational Design Domain (ODD)	
Physical infrastructure	Motorized roadways
Operation constraint	0∼50km/h (City driving speed)
Environmental Condition	Low / High illumination, Rain, Snow
Zone	Urban and rural areas (Densely populated areas)
Connectiviey	Online / Offline

3) Publicly Available Map Accuracy: The proposed method enables accurate vehicle localization through place recognition and template matching based on publicly available maps (e.g., national public GIS data, OpenStreetMap). The errors related to the public map used for map matching were further discussed since the error can vary depending on the reference map. First, the publicly provided GIS map, with a maximum scale of 1:1000, was created with a covariance of 0.36m [8]. Hence, the proposed method allows for vehicle positioning at decimeter-level accuracy maximum. However,

TABLE IX

MEAN ATE(TRANSLATION) OF DEGRADATION EVALUATION

Sequences	G-ICP (w/ HD Map)	Ours (w/ HD Map)	Ours (w/ Public Map)
KITTI 00	0.111 m	0.377 m	0.580 m
KITTI 02	0.362 m	0.594 m	5.983 m

crowd-sourced OpenStreetMap exhibits variations in accuracy concerning the presence of buildings, road shapes, and building outlines across different regions. The accuracy assessment of OpenStreetMap has been discussed in recent studies [42], [43], with reported variations ranging from a minimum of 0.3 m to a maximum of 2 m. Additionally, the non-map error of the proposed method can reach decimeter-level precision, less than the errors associated with OSM sources. However, if the publicly available map closely aligns with the actual environment, publicly available maps have minimal errors within 0.1 m [54]. This showcases that our proposed approach can achieve decimeter-level localization accuracy at maximum even without the need for an HD map in cases where the publicly available map closely represents the real-world environment. Therefore, ensuring the accuracy of public map sources is important for achieving higher accuracy.

4) Degradation Cases: Furthermore, additional evaluations were conducted to analyze the errors caused by the proposed method, apart from the errors caused by the maps mentioned in the previous section. For further evaluation, we compared the 3D HD map-based matching method combined with Voxelized-GICP and our proposed method that substitutes HD maps for 2D public maps. As shown in Table IX, the localization error of the proposed method itself demonstrated within the decimeter level without noticeable degradation when using the HD Map as the reference. Moreover, even considering the error derived from the 2D public map, the proposed method can achieve decimeter-level localization accuracy at maximum in non-obstructed areas (e.g., KITTI00). Therefore, as the accuracy of the 2D public map increases, the proposed method can achieve even more accurate localization. However, as shown in Fig. 11, the localization accuracy of the proposed method degrades in areas where the building outline is elusive due to obstacles (e.g., KITTI02).

B. Ablation Study

We conducted an additional ablation study to validate the structural performance of the proposed place recognition network. The proposed network employs a Siamese structure to extract common local and global descriptors from different representations of two images. Therefore, for the ablation study of the proposed network, we conducted the study in four phases: firstly, a comparison between the Siamese structure and a single network structure; secondly, a comparison between ResNet and the VGG-16 used in the existing NetVLAD and the impact of NetVLAD global descriptors versus standard max-pooling descriptors; and finally, an examination of the retrieval performance based on the crop map size in the proposed method.

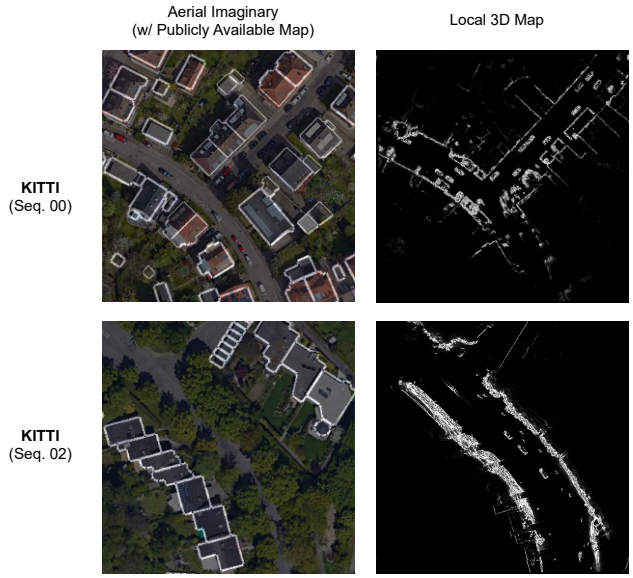


Fig. 11. Comparing the merged image with aerial imagery and public maps to the 3D local map image. In KITTI sequence 00, where building information was abundant, can localize vehicle at decimeter level. But in KITTI sequence 02, despite nearby building data, obstacles hindered the extraction of building facade point clouds. As a result, localization accuracy diverges over the sub-meter.

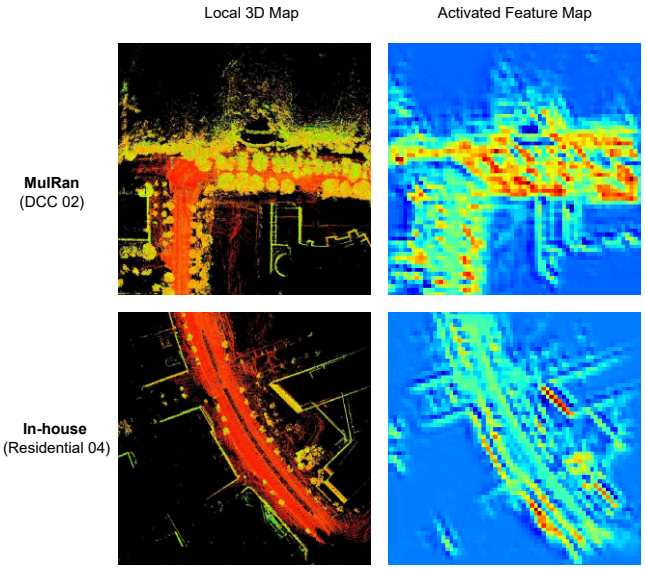


Fig. 12. Highly activated feature map extracted from the conv-10 layer of the proposed place recognition network. The proposed network focuses on building and road shapes like humans when localizing using publicly available maps.

1) Single Network vs Siamese Network: The proposed method employs a Siamese network architecture to find

common features between a vector graphics-based 2D public map and a local 3D map. To investigate the effect of the Siamese network, we conducted training and evaluation using the same configuration, training, and recall@N but with a single network. The experiments were conducted on four datasets with different sensor configurations. As shown in the experimental results, Fig.13, the proposed method employs a Siamese network architecture, facilitating precise place recognition by sharing the feature space across distinct network structures, aiding in identifying shared features between two images. As a result, our Siamese network can attend to building outline and road shape information shown in Fig.12. However, when a single network structure is used, accuracy notably diminishes due to differing representations of the two images.

2) Pooling layer Methods: Moreover, we conducted an investigation into the global pooling technique applied to the feature maps generated by the Siamese network. Illustrated in Fig.13, the performance of the proposed NetVLAD layer, functioning as a pooling layer, surpassed that of pooling utilizing the max-pooling layer. Notably, our experimentation involved VGG16+NetVLAD and VGG16+Max, wherein the Siamese network's ResNet18 backbone was replaced with VGG-16, and both were trained under identical configurations. Additionally, employing ResNet18 as the local feature extractor showcased improved Recall@N outcomes compared to utilizing VGG-16. This improvement can be attributed to the effectiveness of ResNet18, leveraging residual networks for efficient learning without encountering degradation.

3) Crop Map Size Evaluation: Before performing place recognition using the proposed network, images need to be converted to the local 3D map and public 2D map using a crop map size γ. The choice of crop map size has an impact on place recognition performance at large opening areas. As shown in Fig.14, in densely populated, complex urban areas, the performance difference based on crop map size is not substantial. However, in areas with large open spaces like campuses, where there are a few detectable buildings with a small map crop size, the accuracy decreases. Therefore, the proposed method may have limitations when used in areas with large open spaces. In such regions, there is relatively limited building and road information available.

4) Accuracy Improvement of Each Sequence: Our experimental evaluation confirms that the proposed method, using a multi-level localization sequence, significantly improves autonomous vehicle accuracy to the decimeter level. We conducted an ablation study at `Urban 07` sequence to assess the individual contributions of each level of the proposed method to further enhance accuracy during localization. As depicted in Fig.15, place recognition (PR) fails to achieve metric localization and is susceptible to false positive matches. With the addition of a particle filter, place recognition (PR+PF) achieves a median error of 1.962 meters, enabling it to reject false positives and attain metric-level localization. Further, when fused with precise localization (PR+PF+PL), a median error of 0.918 meters is achieved, ensuring decimeter-level accuracy.

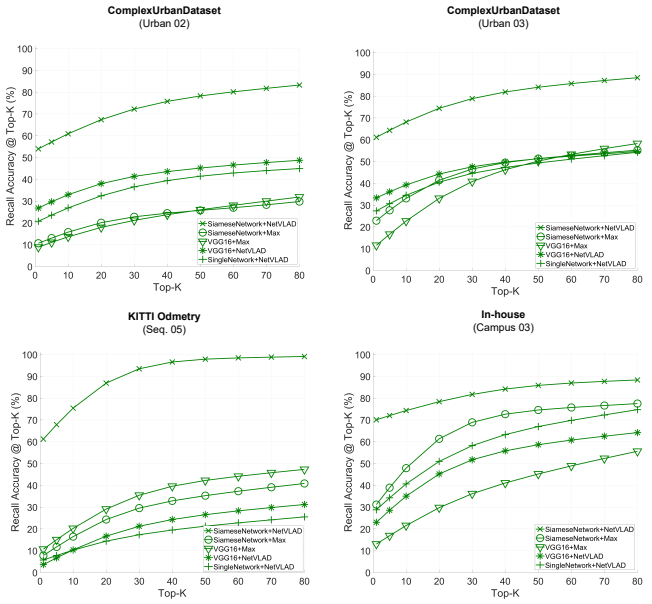


Fig. 13. Recall accuracy comparison result. The proposed method's network outperforms than existing image-based place recognition network. Our Siamese network architecture can retrieve 3D local map images to public 2D map images with high accuracy, even if their expression is dissimilar.

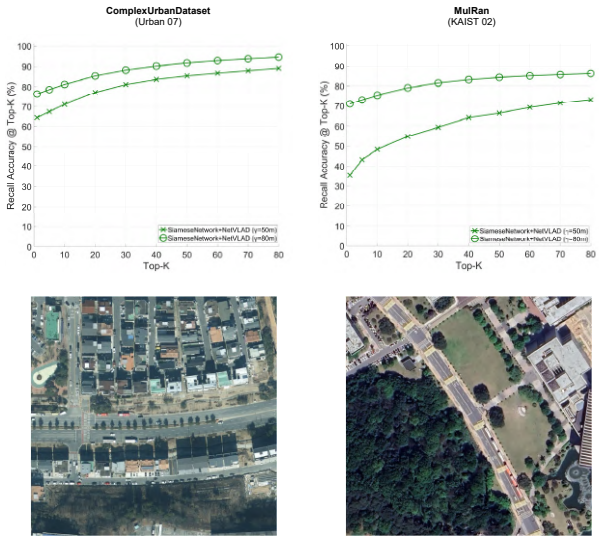


Fig. 14. Recall@N comparison of place recognition using the proposed method with different map crop sizes. Both aerial images are generated at the same scale in 1:1000. `Urban 07` sequence contains building information densely, but `KAIST 02` sequence generated from large opening and lack of building information.

VIII. Conclusion

In this study, a vehicle-localization framework is proposed that functions without prior HD maps and uses only publicly available maps and vehicle onboard sensors. The proposed

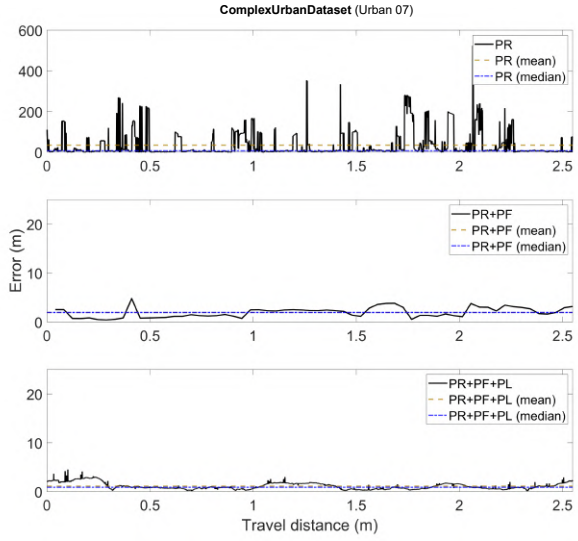


Fig. 15. Comparing accuracy improvements in the `Urban 07` sequence. In place recognition (PR), errors are notably high due to false positives. The addition of the particle filter (PR+PF) significantly enhances accuracy. Moreover, when precise localization (PR+PF+PL) is integrated, decimeter-level localization accuracy is achieved.

multi-level localization framework could accurately estimate vehicle position using spatial data interpretation, place recognition, and precise localization sequence, devised from a human-like localization framework. Similar to humans, the proposed method can localize vehicles by consistently comparing building outlines and road shapes on a public 2D map and enables vehicles to localize accurately, even vehicles that do not need to communicate online or prior visits to use a previously created HD map. To estimate the accuracy of the proposed methods, experiments are conducted using several mobile mapping systems in various countries and publicly available maps. Experimental results demonstrate that the proposed method can estimate the odometry of vehicles maximally at the decimeter level even without an HD map or additional positioning sensor. As a result, our method does not require any large storage for an HD map database or expensive RTK-GPS and an INS system for autonomous localization.

In future studies, real-world adaptation can be conducted to localize vehicles in non-HD map areas. Using the proposed methods, autonomous vehicles can localize in areas where it is relatively difficult to generate and update HD maps (e.g., residential areas and urban areas). Additionally, because prior map accuracy is a key element for accurate localization, improving public map accuracy without a time-consuming process (e.g., using MMS, human annotating) can be conducted in future works.

REFERENCES

[1] E. Guizzo, "How google's self-driving car works," *IEEE Spectrum Online*, vol. 18, no. 7, pp. 1132–1141, 2011.

[2] L. Kent, "HERE introduce HD maps for highly automated vehicle testing," *HERE, Amsterdam, The Netherlands. Available online: http://360. here. com/2015/07/20/here-introduces-hd-maps-for-highlyautomated-vehicle-testing/(accessed on 16 April 2018)*, 2015.

[3] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[4] R. Liu, J. Wang, and B. Zhang, "High definition map for automated driving: Overview and analysis," *The Journal of Navigation*, vol. 73, no. 2, p. 324–341, 2020.

[5] M. Elhousni, Y. Lyu, Z. Zhang, and X. Huang, "Automatic building and labeling of hd maps with deep learning," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, pp. 13255–13260, Apr. 2020.

[6] Q. Luo, Y. Cao, J. Liu, and A. Benslimane, "Localization and navigation in autonomous driving: Threats and countermeasures," *IEEE Wireless Communications*, vol. 26, no. 4, pp. 38–45, 2019.

[7] OpenStreetMap contributors, "Planet dump retrieved from https://planet.osm.org ." https://www.openstreetmap.org, 2017.

[8] "Mapping: Topographic map, Large-scale Digital Topographic Map," *National Geographic Information Institute, Ministry of Land, Infrastructure and Transport, Republic of Korea. Available online: https://www.ngii.go.kr/kor/content.do?sq=207(accessed on 12 April 2022)*, 2022.

[9] F. Yan, O. Vysotska, and C. Stachniss, "Global localization on openstreetmap using 4-bit semantic descriptors," in *European Conference on Mobile Robots (ECMR)*, pp. 1–7, 2019.

[10] G. Floros, B. Van Der Zander, and B. Leibe, "OpenStreetSLAM: Global vehicle localization using openstreetmaps," in *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1054–1059, 2013.

[11] Y. Cho, G. Kim, S. Lee, and J.-H. Ryu, "OpenStreetMap-Based LiDAR Global Localization in Urban Environment Without a Prior LiDAR Map," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 4999–5006, 2022.

[12] M. A. Brubaker, A. Geiger, and R. Urtasun, "Lost! leveraging the crowd for probabilistic visual self-localization," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3057–3064, 2013.

[13] P. Ruchti, B. Steder, M. Ruhnke, and W. Burgard, "Localization on OpenStreetMap data using a 3D laser scanner," in *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 5260–5265, 2015.

[14] T. Y. Tang, D. De Martini, D. Barnes, and P. Newman, "Rsl-net: Localising in satellite images from a radar on the ground," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 1087–1094, 2020.

[15] T. Y. Tang, D. De Martini, S. Wu, and P. Newman, "Self-supervised learning for using overhead imagery as maps in outdoor range sensor localization," *International Journal of Robotics Research*, vol. 40, no. 12-14, pp. 1488–1509, 2021.

[16] T. Y. Tang, D. De Martini, and P. Newman, "Get to the point: Learning lidar place recognition and metric localisation using overhead imagery," *Robotics: Science and Systems*, 2021.

[17] F. Fervers, S. Bullinger, C. Bodensteiner, M. Arens, and R. Stiefelhagen, "Continuous self-localization on aerial images using visual and lidar sensors," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 7028–7035, 2022.

[18] J. C. Xia, C. Arrowsmith, M. Jackson, and W. Cartwright, "The wayfinding process relationships between decision-making and landmark utility," *Tourism Management*, vol. 29, no. 3, pp. 445–457, 2008.

[19] E. H. Cornell, A. Sorenson, and T. Mio, "Human sense of direction and wayfinding," *Annals of the Association of American Geographers*, vol. 93, no. 2, pp. 399–425, 2003.

[20] H. Iftikhar and Y. Luximon, "Wayfinding information syntheses: A study of wayfinding efficiency and behavior in complex outdoor institutional environment," *HERD: Health Environments Research & Design Journal*, vol. 16, no. 2, pp. 250–267, 2023.

[21] J. Levinson, M. Montemerlo, and S. Thrun, "Map-based precision vehicle localization in urban environments.," in *Robotics: science and systems*, vol. 4, p. 1, 2007.

[22] J. Levinson and S. Thrun, "Robust vehicle localization in urban environments using probabilistic maps," in *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 4372–4378, 2010.

[23] M. Schreiber, C. Knöppel, and U. Franke, "LaneLoc: Lane marking based localization using highly accurate maps," in *IEEE Intelligent Vehicles Symposium (IV)*, pp. 449–454, 2013.

[24] W.-C. Ma, I. Tartavull, I. A. Bârsan, S. Wang, M. Bai, G. Mattyus, N. Homayounfar, S. K. Lakshmikanth, A. Pokrovsky, and R. Urtasun, "Exploiting Sparse Semantic HD Maps for Self-Driving Vehicle Localization," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 5304–5311, 2019.

[25] C. Forster, L. Carlone, F. Dellaert, and D. Scaramuzza, "IMU Preintegration on Manifold for Efficient Visual-Inertial Maximum-a-Posteriori Estimation.," in *Robotics: Science and Systems*, 2015.

[26] T. Shan and B. Englot, "LeGO-LOAM: Lightweight and Ground-Optimized Lidar Odometry and Mapping on Variable Terrain," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 4758–4765, 2018.

[27] T. Shan, B. Englot, D. Meyers, W. Wang, C. Ratti, and D. Rus, "LIO-SAM: Tightly-coupled Lidar Inertial Odometry via Smoothing and Mapping," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 5135–5142, 2020.

[28] H. Ye, Y. Chen, and M. Liu, "Tightly Coupled 3D Lidar Inertial Odometry and Mapping," in *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 3144–3150, 2019.

[29] C. Le Gentil, T. Vidal-Calleja, and S. Huang, "IN2LAAMA: Inertial lidar localization autocalibration and mapping," *IEEE Transactions on Robotics*, vol. 37, no. 1, pp. 275–290, 2020.

[30] K. Koide, M. Yokozuka, S. Oishi, and A. Banno, "Voxelized gicp for fast and accurate 3d point cloud registration," in *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 11054–11059, 2021.

[31] P. Alken, E. Thébault, C. D. Beggan, H. Amit, J. Aubert, J. Baerenzung, T. N. Bondar, W. J. Brown, S. Califf, A. Chambodut, A. Chulliat, G. A. Cox, C. C. Finlay, A. Fournier, N. Gillet, A. Grayver, M. D. Hammer, M. Holschneider, L. Huder, G. Hulot, T. Jager, C. Kloss, M. Korte, W. Kuang, A. Kuvshinov, B. Langlais, J.-M. Léger, V. Lesur, P. W. Livermore, F. J. Lowes, S. Macmillan, W. Magnes, M. Mandea, S. Marsal, J. Matzka, M. C. Metman, T. Minami, A. Morschhauser, J. E. Mound, M. Nair, S. Nakano, N. Olsen, F. J. Pavón-Carrasco, V. G. Petrov, G. Ropp, M. Rother, T. J. Sabaka, S. Sanchez, D. Saturnino, N. R. Schnepf, X. Shen, C. Stolle, A. Tangborn, L. Tøffner-Clausen, H. Toh, J. M. Torta, J. Varner, F. Vervelidou, P. Vigneron, I. Wardinski, J. Wicht, A. Woods, Y. Yang, Z. Zeren, and B. Zhou, "International geomagnetic reference field: the thirteenth generation," *Earth, Planets and Space*, vol. 73, p. 49, Feb 2021.

[32] S. V. Estopinal, *A guide to understanding land surveys*. John Wiley & Sons, 1993.

[33] H. Noh, A. Araujo, J. Sim, T. Weyand, and B. Han, "Large-scale image retrieval with attentive deep local features," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pp. 3456–3465, 2017.

[34] J. Revaud, C. De Souza, M. Humenberger, and P. Weinzaepfel, "R2d2: Reliable and repeatable detector and descriptor," *Advances in neural information processing systems*, vol. 32, 2019.

[35] R. Arandjelovic, P. Gronat, A. Torii, T. Pajdla, and J. Sivic, "NetVLAD: CNN Architecture for Weakly Supervised Place Recognition," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5297–5307, 2016.

[36] M. Zhu, Y. Yang, W. Song, M. Wang, and M. Fu, "Agcv-loam: Air-ground cross-view based lidar odometry and mapping," in *Chinese Control and Decision Conference (CCDC)*, pp. 5261–5266, 2020.

[37] L. Sun, D. Adolfsson, M. Magnusson, H. Andreasson, I. Posner, and T. Duckett, "Localising faster: Efficient and precise lidar-based robot localisation in large-scale environments," in *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 4386–4392, 2020.

[38] D. Lowe, "Object recognition from local scale-invariant features," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, vol. 2, pp. 1150–1157 vol.2, 1999.

[39] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool, "Speeded-up robust features (SURF)," *Computer vision and image understanding*, vol. 110, no. 3, pp. 346–359, 2008.

[40] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "ORB: An efficient alternative to SIFT or SURF," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pp. 2564–2571, 2011.

[41] J. N. Sarvaiya, S. Patnaik, and S. Bombaywala, "Image registration by template matching using normalized cross-correlation," in *international conference on advances in computing, control, and telecommunication technologies*, pp. 819–822, 2009.

[42] M. A. Brovelli and G. Zamboni, "A new method for the assessment of spatial accuracy and completeness of openstreetmap building footprints," *ISPRS International Journal of Geo-Information*, vol. 7, no. 8, p. 289, 2018.

[43] Y. Liu, W. Shi, H. Zhang, and M. Zhang, "A multilevel stratified spatial sampling approach based on terrain knowledge for the quality assessment of openstreetmap dataset in hong kong," *Transactions in GIS*, vol. 27, no. 1, pp. 290–318, 2023.

[44] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for Autonomous Driving? The KITTI Vision Benchmark Suite," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3354–3361, 2012.

[45] J. Jeong, Y. Cho, Y.-S. Shin, H. Roh, and A. Kim, "Complex Urban Dataset with Multi-level Sensors from Highly Diverse Urban Environments ," *International Journal of Robotics Research*, vol. 38, no. 6, pp. 642–657, 2019.

[46] G. Kim, Y. S. Park, Y. Cho, J. Jeong, and A. Kim, "Mulran: Multimodal range dataset for urban place recognition," in *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 6246–6253, 2020.

[47] B. Horn, "Closed-Form Solution of Absolute Orientation Using Unit Quaternions," *Journal of the Optical Society A*, vol. 4, pp. 629–642, 04 1987.

[48] M. A. Uy and G. H. Lee, "PointNetVLAD: Deep Point Cloud Based Retrieval for Large-Scale Place Recognition," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4470–4479, 2018.

[49] X. Chen, I. Vizzo, T. Läbe, J. Behley, and C. Stachniss, "Range Image-based LiDAR Localization for Autonomous Vehicles," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2021.

[50] S. Kato, S. Tokunaga, Y. Maruyama, S. Maeda, M. Hirabayashi, Y. Kitsukawa, A. Monrroy, T. Ando, Y. Fujii, and T. Azumi, "Autoware on board: Enabling autonomous vehicles with embedded systems," in *ACM/IEEE International Conference on Cyber-Physical Systems (ICCPS)*, pp. 287–296, 2018.

[51] I. D. Miller, A. Cowley, R. Konkimalla, S. S. Shivakumar, T. Nguyen, T. Smith, C. J. Taylor, and V. Kumar, "Any way you look at it: Semantic crossview localization and mapping with lidar," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 2397–2404, 2021.

[52] G. Kim and A. Kim, "Scan Context: Egocentric Spatial Descriptor for Place Recognition Within 3D Point Cloud Map," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 4802–4809, 2018.

[53] I. T. Forum, *Road Safety Annual Report*. 2022.

[54] K. Wong, E. Javanmardi, M. Javanmardi, Y. Gu, and S. Kamijo, "Evaluating the capability of openstreetmap for estimating vehicle localization error," in *IEEE Intelligent Transportation Systems Conference (ITSC)*, pp. 142–149, 2019.