Pre-Estimating Self-Localization Error of NDT-Based Map-Matching From Map Only

Ehsan Javanmardi, *Member, IEEE*, Mahdi Javanmardi, *Member, IEEE*, Yanlei Gu, *Member, IEEE*, and Shunsuke Kamijo, *Senior Member, IEEE*

***Abstract*— Map-matching based on light detection and ranging (LiDAR) is a promising method for accurate self-localization and recently has gained a wider focus due to the availability of high definition (HD) maps and price-down of LiDARs. In this method, the input scan of the LiDAR is matched to the prebuilt map to get a centimeter-level accuracy position of the vehicle. However, in some places of the map, due to the lack of features, the presence of the repetitive features, the layout of the features, and other factors, the map-matching error might exceed the required bound for autonomous driving. In our previous work, four criteria for evaluation of the features of the map was introduced and it is shown that by examining the corresponding factors for each criterion, the map-matching error can be modeled. In this work, one of the map criteria called local similarity is further investigated and in order to quantify the fulfillment of this criterion, three new factors, namely *pfh_similarity*, *pfh_entropy*, and *battacharya_similarity* are introduced. In addition to this, a framework for pre-estimation of the map-matching error considering these four criteria based on random forest regression is proposed. To evaluate the accuracy of the framework, experiments were conducted for 3.6 km in Shinjuku, Tokyo. Experimental results show that using the proposed framework, in 64.1% of the cases, the localization error can be estimated with less than 2.5cm of the estimation error.**

***Index Terms*— Autonomous driving, high definition map, self-localization, positioning uncertainty, LiDAR.**

I. Introduction

MAP is one of the pillars of autonomous driving along with sensing and path planning [1]. While vehicles are technically able to drive itself without the help of an HD (high-definition) map, this dream is too far for autonomous vehicles considering current sensing and computation capabilities.

An HD maps have extremely high precision at centimeter-level and contain a lot of information to help autonomous driving in many aspects. For example, precise map information is necessary for the vehicle to localize itself in relation to the surrounding environment [2]. The map can be also used as a virtual sensor for the vehicle to recognize and react to static objects on the roads, such as traffic lights, traffic signs, and road markings far beyond the range of onboard sensors [3]. In addition, by adding the speed profiles derived from actual human drivers to the map, it can be used to for human-like autonomous driving and increasing the safety. And most importantly, the map is necessary for navigation and to compensate the sensor insufficiencies [4].

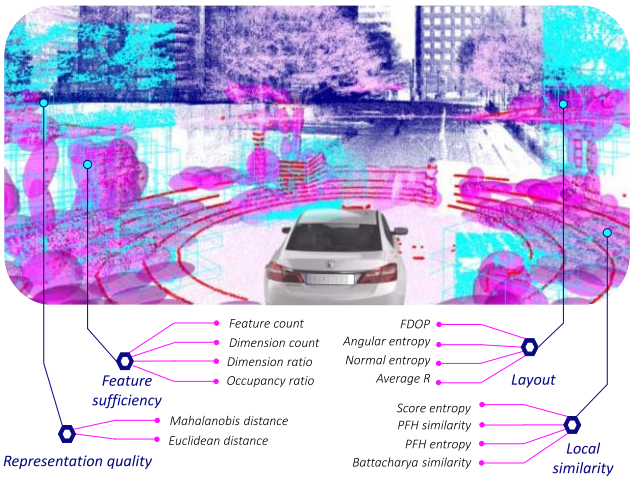

Fig. 1. Overview of the proposed criteria and factros to estimate the self-localization error.

In order to use the HD map for the aforementioned applications, the vehicle needs to localize itself in a centimeter-level accuracy in relation to the map, which is not always easy to achieve.

Light detection and ranging (LiDAR) is one of the most promising technologies for an accurate localization due to its immunity to illumination change, accurate distance measurement, and lower computation complexity. By having a prebuilt map, the observations from the LiDAR can be matched to the map to obtain the position of the vehicle. This map can have different formats, such as point cloud map, occupancy grid map, normal distribution map, etc.

The process of matching the input scan to the map is known as map-matching. Comparing to typical approaches of simultaneous localization and mapping (SLAM) which do not use a prebuild map, self-localization based on map-matching is more practical and robust as it relies on a georeferenced dense map and does not suffer from accumulative error.

In order to achieve globally accurate self-localization results using a map, the map itself should be globally accurate yet precise. However, having an accurate and precise map

Manuscript received June 30, 2019; revised January 5, 2020 and June 10, 2020; accepted June 15, 2020. The Associate Editor for this article was C. Wen. *(Corresponding author: Mahdi Javanmardi.)*

The authors are with the Institute of Industrial Science (IIS), The University of Tokyo, Tokyo 113-8654, Japan (e-mail: ehsan@kmj.iis.u-tokyo.ac.jp; mahdi@kmj.iis.u-tokyo.ac.jp; guyanlei@kmj.iis.u-tokyo.ac.jp; kamijo@iis.u-tokyo.ac.jp).

Digital Object Identifier 10.1109/TITS.2020.3006854

does not guaranty an accurate localization. Tunnels and rural areas without plentiful features are some examples that the localization error might occur even by providing a globally accurate features in the map.

To provide high accuracy self-localization solution everywhere on the map, it is vital to define in advance the areas where the localization error might exceed the required bound for autonomous driving. Based on this knowledge, system designers can do necessary adjustments and optimizations to deal with these errors. There are also real-time error estimation methods that get the laser sweeps and map as an input and estimate the localization error. However, in some applications such as dynamic refinement of map resolution or dynamic determination of laser range for map-matching, the error estimation should be performed beforehand. In order to use real-time methods for estimating the localization error of the map beforehand, the experimental vehicle should run all routes and find the erroneous part of the map.

However, running the experimental vehicle for thousands of kilometers to explore these critical points is very costly, time-consuming and impractical. Therefore it is important to be able to estimate the self-localization error by only investigating the map.

This work is an extension of our previous works [5], [6]. In [5], we introduced the idea of modeling the localization error of a specific point by evaluating the characteristics of the map around that point. In order to evaluate the map, four factors were introduced. The introduced factors are *feature_count*, feature dilution of precision (*FDOP*), *normal_entropy*, and *score_entropy*. The work was later extended by adding more factors and categorizing them into four general map evaluation criteria [6]. These criteria are *feature sufficiency*, *layout*, *local similarity*, and *representation quality* of the map. By investigating these criteria for each point on the map, we are answering questions such as "are the features around point enough for localization?" (*feature sufficiency*), "are the features' distribution or layout suitable?" (*layout*), "does the surrounding environment have repetitive features?" (*local similarity*), and "are the chosen map format or map resolution capable of showing the details of the surrounding environment?" (*representation quality*).

In this work, we introduced several new factors to better model the local similarity criterion. These factors are *pfh_similarity*, *pfh_entropy*, and *battacharya_similarity.* In addition to this, a framework for pre-estimation of the map-matching error considering these four criteria based on random forest regression is proposed. After training the random forest with actual localization errors for certain areas of the map, the proposed framework can be used to predict the localization error for any other points in the map. The density or interval of the estimation depends on the application. In this study, a 1.0m interval is used for the estimation of the localization error.

The rest of this paper is organized as follows:

Section II introduces related works. Section III describes the error source for map matching-based self-localization. Section IV explains the map criteria for evaluation and section V factors for quantifying these criteria are defined. In section VI, these factors are used to estimate the localization error of the map. Section VII evaluates the proposed framework and contribution of each factor to the self-localization error by conducting the experiments. Finally, section VIII concludes this paper.

II. Related Works

Basically, to localize the ego-vehicle using a prebuilt map, observations from the vehicle derived using a camera or laser scanner are associated with the map. If the map is geo-referenced and is accurate, the obtained position is also considered as the globally accurate location of the vehicle. There are various papers in the literature focusing on how to improve localization and map matching algorithms [7]–[9], what map format is suitable for localization [10]–[13], and how to generate a globally accurate map for autonomous driving [14], [15]. However, research which can answer how information on the map can affect the localization is very rare.

The Iterative Closest Point (ICP) is one of the popular registration algorithms which can be used for SLAM and localization. In [7], the authors propose a branch-and-bound framework for global registration called Go-ICP which overcomes the local minima problem of the conventional ICP algorithm. Serafin *et al.* take normal and curvature information into account to recursively align point clouds and improve the speed and robustness of the ICP [8]. Their method uses the Normal Distribution Transform (NDT) representation of the scene. The general idea behind NDT is to represent the scene using a set of Gaussian distributions computed from the point cloud, and apply optimization method to compute the transformation [9]. Hornung *et al.* [10] presented a map format based on octrees and probabilistic occupancy estimation to create 3D maps of the environment from noisy measurements. Some researchers used different road features as map representation. Schreiber *et al.* [11] take curbs and road markings into account as map information to localize the vehicle. Wolcott *et al.* present a multiresolution Gaussian mixture map characterizing the height and reflectivity distribution of the environment [12]. In our previous work [13], a self-localization based on the abstracted map format was proposed. In that research, two abstracted map formats were proposed, namely a multi-layered 2D vector map and a probabilistic planar surface map. The multi-layered 2D vector map represents different heights of the building facades in vector format and a 3D probabilistic planar surface map represents the surfaces of buildings. When abstracting a point cloud map using vectors and planes, the selected resolution for the abstraction has a major effect on localization accuracy. However, this effect is not constant from place to place. Apart from map format and its resolution, it is important to make the prebuilt map globally accurate. One standard method for obtaining global accuracy is the landmark update using Ground Control Points (GCP) [14]. Instead of measuring GCPs by a field survey, high-resolution aerial imagery can be employed to automatically georeferenced the prebuilt map [15].

Although using globally accurate high-definition is essential for autonomous driving, it still cannot guarantee accurate localization. Localization accuracy at each point on the map

is influenced by its surrounding environment. Akai *et al.* [16] estimate the error of the registration at every point on the map experimentally and use the generated uncertainty for the particle filter. In this method, to acquire uncertainty of the registration pre-experiment is necessary. In [17], the authors used Open Street Map (OSM) and information of building footprints to estimate points that are likely to reduce the pose uncertainty and improve the graph-based SLAM. In [18], the authors proposed a method to calculate the quality of the map for visual odometry by heuristically estimating the entropy of the map. In our previous work [6], four map criteria, namely feature sufficiency, layout, local similarity, and representation quality was proposed and for each of them, several map factors were defined to the model capability of the map for self-localization. In order to model the localization error, principal component regression was used. One of the advantages of this method is that the proposed factors can be directly calculated from the map without any pre-experiment.

III. Error Sources

In this paper, map-matching is performed by 3D-NDT matching. Generally, NDT models the distribution of 3D point cloud by a collection of local normal distributions. This representation of the map can benefit the map-matching algorithm in two aspects.

- First, no explicit correspondences between points or features have to be established. This makes NDT faster and robust compared to other matching techniques.
- Second, having a normal distribution, we can calculate all derivatives analytically. This makes NDT fast and accurate.

NDT map is generated as follow:

1. The point cloud is subdivided regularly into smaller cells with constant size l.
2. Calculate the mean and covariance for each cell
3. Model each cell by the normal distribution

We now have a piecewise continuous and differentiable description of the point cloud map which is called an ND map.

Matching of the laser sweeps with the ND map is performed as follows:

1. Laser sweep is down-sampled using fixed grid size l.
2. Laser sweep is transformed over the ND map using the initial guess transformation matrix.
3. Determine the corresponding normal distributions for each point in the sweep.
4. The score of matching is calculated based on Log-Likelihood function
5. Calculate a new transformation matrix by trying to optimize the score. This is done by performing one step of Newton's Algorithm.
6. Goto 4 until a convergence criterion is met.

Readers are referred to [20] for more discussion about the 3D-NDT algorithm.

In localization based on NDT map-matching, an error comes from four main sources. These sources are input scan, matching algorithm, dynamic phenomena, and map. The role of each phenomenon is explained in this section.

A. Input Scan

1) Input Scan Quality: The quality of the input scan affects the localization error. Denser input scan can have more details of the environment. For example, an input scan from Velodyne HDL-64 which has 64 channels (layers of the beam) can capture the surrounding environment denser than its VLP-16 model which has only 16 channels. More details from surrounding environments help to match input scant to prebuilt map more accurately. In addition to the density, other sensor-related parameters such as vertical and horizontal field of view, horizontal resolution, laser range, and sensor setup parameters such as pitch and roll are important.

2) Downsampling of the Input Scan: Generally, in the matching algorithm, the input scan is down-sampled for two reasons. One is to decrease the computation complexity and the other is to make the input scan more even.

This phase removes some details and adds uncertainty to the input scan.

3) Distortion of the Input Scan: Distortion happens when the vehicle is moving. Distortion is related to the frequency of the scanning. Suppose that the scanning frequency is 10Hz which is optimum considering the computation time of the matching, and the speed of a vehicle is 36Km. In this case, in each rotation of the laser beams, the vehicle moves 1.0m, and the first point and the last point of the laser input which point to the same object has 1.0m shift. This means that the scan points have an average shift of 50 cm and thus localization error might be near 50cm.

By having access to an odometer, IMU, or GPS data, this distortion can be partially compensated. The effect of scan distortion on the map-matching depends on the map features. For example, for parallel walls to the moving direction, scan distortion will not affect the scan-matching, while walls perpendicular to the moving direction of the vehicle are prone to relay the scan distortion to matching error.

B. Matching Algorithm

1) Robustness and Accuracy: A matching algorithm can affect the localization accuracy as well. Some matching algorithms are more robust to local optimum and some of them are more robust to partial overlapping.

2) Initial Guess: For map-matching techniques before the matching of input scan to the map, the initial guess of the input scan should be defined. Initial guess has a huge effect on the matching especially for the optimization-based techniques because if the initial guess is not close enough, it might be stuck on the local optimum. This initial guess is very important in the NDT-based method as well.

C. Dynamic Phenomena

Dynamic phenomena in the environment can be a source of error. This can be environment change due to the construction and destruction of the buildings and road facilities or seasonal changes such as leaf fall of the trees. In addition to this,

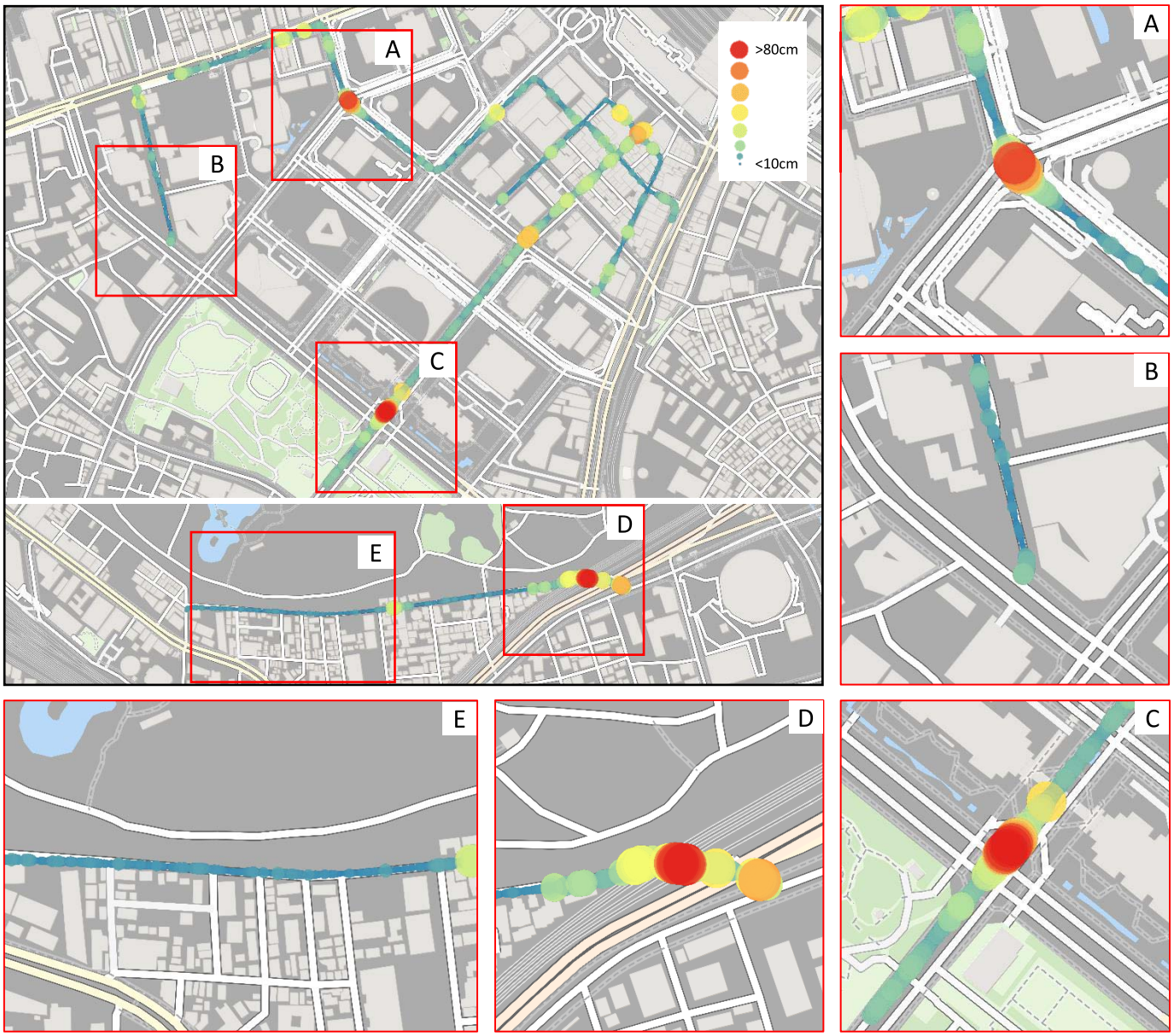

Fig. 2. Self-localization error in Shinjuku, Tokyo. Color and size of circles shows the localization error. Higher error has bigger size with color more close to red.

more frequent changes in the environment due to the presence of dynamic objects such as cars and pedestrians are another source of error.

D. Prebuilt Map

In map-matching methods, the prebuilt map plays a significant role in localization accuracy. For accurate self-localization, the global and local accuracy and the precision of the map is essential. In fact, in order to have a centimeter-level of global accuracy, firstly our prebuild map should satisfy this requirement. However, the highly accurate map does not guarantee the accuracy of the localization [19]. In other words, map accuracy is different than the ability of the map for localization. To clarify the discussion, in Fig. 2, 3D localization error based on map-matching for 5 paths in Shinjuku, Tokyo is shown. In this experiment, an input scan from Velodyne's VLP-16 (Fig. 10(center)) is matched with the normal distribution map with 2.0 m grid size using a normal distribution transform (NDT) algorithm [20]. For more discussion about ground truth and experimental setup, readers are referred to section VI. In Fig. 2, the color and size of the circles show the localization error. A higher error has a bigger circle and color closer to red. Percentile rank error for each path is shown in Table I. For example around 17.9 % of positions in the path III has localization error more than 20cm. In 93.7% of the positions for all paths, the localization errors are smaller than 25 cm. However, in some areas such as A, C, and D, the localization errors reach to near 1.0m. The actual environments for these scenarios are shown in Fig. 3. In Fig. 3(A), vehicle goes to a very wide intersection with very few features. The buildings are too far for an accurate map-matching. In C, the vehicle goes under the bridge and in D the vehicle goes to a short tunnel. Both of these scenarios

Fig. 3. Scenarios in which, localization become challenging and erroneous. Labels of the pictures are based on Fig. 2.

TABLE I
PERCENTILE RANK OF 3D LOCALIZATION ERROR IN THE 5 PATHS FOR DIFFERENT ERROR THRESHOLDS

Path index	I	II	III	VI	V
Percentile rank (>25cm) [%]	2.2	1.9	17.9	1.9	7.0
Percentile rank (>20cm) [%]	4.3	7.3	20.7	4.12	15.0
Percentile rank (>15cm) [%]	11.1	17.9	23.2	8.4	33.4
Percentile rank (>10cm) [%]	28.9	40.8	27.6	18.7	86.0

look challenging for the matching. Fig. 3(top-right) shows the environment of path III. In path III, the right side of the road are filled with vegetation and trees. Table I shows localization error in path III is very higher than the other paths. In 17.9% of the cases in the path III, the error of localization is higher than 25 cm and this is five times of average of other paths.

There are some hypothesises about why those scenarios make the map-matching challenging. For example in A, the vehicle goes to a very wide intersection. In this case, buildings are too far from vehicle and other features for matching are few as well. In C and D, the vehicle is surrounded by two long walls. In these scenarios, laser scanner sweeps barely capture a feature perpendicular to the moving direction of the vehicle because of these two long walls. We call these perpendicular features as lateral features. Lateral features are very helpful for accurate matching in a longitudinal direction which is the moving direction of the vehicle. This is because the input scan can find more lateral correspondences to the map, in order to adjust its longitudinal position. Therefore, in C and D, few lateral features make longitudinal positioning very challenging.

In path III, one side of the road is fully occupied with the bushes and trees which are uncertain features for matching. As shown in Table I, in 86.0% of the path V, localization error is higher than 10cm and this is extremely higher than other paths. This is possible because in path V, the road is wide and features are sparse, and additionally, on several occasions, vehicle passes under a bridge.

In the aforementioned scenarios, the map is not able to provide an accurate self-localization. To achieve accurate self-localization within a map, the map should satisfy some requirements. In other words, the map should meet some specific criteria which define the ability of the map for self-localization. These criteria are explained in section IV.

IV. MAP CRITERIA

In order to evaluate the capability of the map for localization four criteria are proposed. These criteria are *feature sufficiency*, *layout*, *local similarity*, and *representation quality* of the map. These criteria can be assumed as requirements of the map for highly accurate self-localization capability and should be investigated for different positions in the map.

To have an accurate self-localization within the map, the map should contain sufficient features in each position. However, in some cases or better to say, in some positions in the map, due to the characteristics of the environment or the method of map generation, these features are not enough and map-matching takes place with an error. For example, in the urban area, there are more structured features such as buildings and poles comparing to the rural area. Therefore, self-localization based on a map in the city environments has a lower error comparing to the rural area. On the other hand, the quality of each feature is important as well. Quality of features relates to the matching algorithms, but in general, two aspects can be investigated when considering the feature quality in map-matching. These aspects are precision and the direction of matching. A combination of these two aspects can be assumed as the uncertainty of the matching. In fact, map-matching is a process of matching individual features extracted from the input scan to their corresponding features on the map. Each of these individual features proposes a transformation matrix to align best to their corresponding features, and the final transformation matrix calculated by averaging them. However, the accuracy and the possible direction of matching for these individual features are different. For example, the features extracted from bushes are more uncertain than the features extracted from walls, therefore, the transformation matrix generated from features extracted from the latter one is more precise. On the other hand, wall-like features are only capable of proposing the transformation matrix in one direction (parallel to its normal), while pole-like features can propose in two directions (perpendicular to the pole). The *feature sufficiency* criterion investigates the quantity and the quality of the features for accurate self-localization.

In addition to the number and quality of the features, layout and the distribution of these features in the space is important as well. In some parts of the map, there might be plenty of high-quality features. However, as the features are all placed in one corner of the map, the quality of the matching degrades.

In fact, the concept of the layout of the features comes from a global positioning system (GPS), in which a wider spread of the satellites results in better localization accuracy. This is formulated by geometrical dilution of precision (GDOP) [21].

In addition to the distribution of the features, in some parts of the map, there might be a situation such as tunnels, urban canyons, and highways in which the layout of the buildings is not suitable for longitudinal positioning (C and D in Fig. 3). In these situations, the layout of the buildings is so that the

lateral position of the vehicle can be obtained accurately but not longitudinal. Therefore, the facing direction of the features is important. The *layout* criterion is defined to investigate these requirements of the map.

The third criterion is the *local similarity* of the map. By investigating these characteristics of the map, the degree of similarity of features in the map can be obtained. In other words, this criterion shows how much the features of the map are similar to each other. If the map has features that are spatially close and at the same time similar characteristics or shape, it is difficult to find correct correspondents when matching the input scan to it. Apparently, the wrong answer for corresponding points causes an error for matching and localization.

Tunnels or two poles stand next to each other are the cases that the environment has a local similarity and map-matching becomes erroneous. In addition to the environment itself, mapping process, map format, and abstraction ratio can cause an unwanted local similarity of the map as well. For example, if the abstraction ratio is high (low resolution), details of the environment disappear and produce more similar features that are unwanted for map-matching.

Finally, the *representation quality* criterion is defined to investigate the effect of the map format and abstraction ratio on the self-localization. Generally, LiDAR-based map-matching techniques do not use the original point cloud as a map. They either down-sample the original point cloud or convert it to other map formats such as normal distribution (ND) map, occupancy grid map, vector, and so forth to reduce the memory and computation time. However, this abstraction or change in format removes some details of the map which might be important for self-localization. In other words, by applying abstraction to the original point cloud or producing a new map format from the point cloud, information loss happens. This information loss varies from place to place as the details of the environment are different and its effect to the localization is also different. In order to know how much the current abstraction ratio or map format preserves the details of the map, the representation quality criterion should be investigated.

So far, four map criteria are defined and explained. In order to evaluate the capability of the map for self-localization, these criteria should be investigated and considered together. To investigate the fulfillment of these criteria, each of them is quantified based on several factors.

To evaluate the map in the sample point P, the map elements in the local vicinity range of the point P is selected. Local vicinity range is obtained from the range of laser scanner and vertical field of view of the sensor used for localization. In this paper, Velodyne's VLP-16 is used which has $\pm 15^\circ$ of vertical field of view. Extraction of the features in local vicinity is shown in Fig. 4. Then from these selected elements in the local vicinity range, for each criterion, the map factors are calculated. By having the values of each map factors, the error of the localization for the sample point P is estimated and consequently, the capability of the map for accurate self-localization for that point is evaluated.

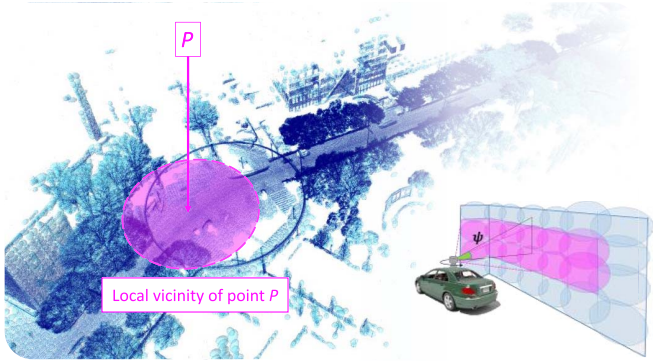

Fig. 4. Local vicinity region of point P. Features in this region is used to calculate the factors.

V. Quantification of Map Criteria

Map factors are calculated based on the normal distribution map.

A. Quantification of Feature Sufficiency Criterion

For *feature sufficiency* criterion, 8 factors are defined in four categories.

1) occupancy_ratio: This factor shows how much the surrounding environment which can be seen by the laser scanner is occupied by the map elements. In order to calculate the $occupancy\_ratio$, 3D space of map elements in the local vicinity is converted to the 2D depth image. $occupancy\_ratio$ is the ratio of the occupied cell over all cells.

$$occupancy_{ratio} = \frac{occupied\_cell}{all\_cells}, \tag{1}$$

2) feature_cnt: This factor shows the total number of the map elements in the local vicinity range.

3) dimention_cnt (D1_cnt, D2_cnt,D3_cnt): In order to investigate the quality of the features, features are divided into three categories based on their shapes. These categories are *1D*, *2D*, and *3D*. *1D features* are pole-like features. *2D features* are wall-like and *3D features* are bushes or trees. Having more of *1D* and *2D features* in the map can cause better matching as they are more certain features. *2D features* can match in one direction while *1D features* capable of matching in two directions. Dimension value of each normal distribution is calculated based on its Eigenvalues [6]. Consider that the Eigenvalues are λ_1, λ_2, and λ_3. From Eigen values the standard deviations σ for each of the Eigenvectors are calculated as follows:

$$\forall i \in [1,3] \quad \sigma_i = \sqrt{\lambda_i}, \tag{2}$$

where i is the index of Eigenvectors. Using the standard deviation, three dimension behavior is defined as follows:

$$a_{1D} = \frac{\sigma_1 - \sigma_2}{\sigma_1}, \quad a_{2D} = \frac{\sigma_2 - \sigma_3}{\sigma_1}, \; a_{3D} = \frac{\sigma_3}{\sigma_1}, \tag{3}$$

Based on the highest dimension behavior, the map elements are divided into pole-like features (*1D features*), wall-like features (*2D features*), and scatters (*3D features*). For instance, in *2D features*, a_{2D} is higher than other values. A number of each feature type (*1D*, *2D*, and *3D*) are defined as $dimension\_cnt$ ($1D\_cnt$, $2D\_cnt$, and $3D\_cnt$).

4) dimention_ratio (D1_ratio, D2_ratio, D3_ratio): In addition to the number of each feature ($dimension\_cnt$), the proportion of each feature is calculated by dividing the number of each feature type over all features. This factor is called $dimension\_ratio$ ($1D\_ratio$, $2D\_ratio$, and $3D\_ratio$).

$$Dm_{ratio} = \frac{Dm_{cnt}}{feature_{cnt}}, \tag{4}$$

where m is the number of dimensions and can be 1, 2, or 3.

B. Quantification of Layout Criterion

In order to quantify layout criterion, four aspects of the layout of the features are investigated and for each of them, a map factor is formulated.

1) FDOP (Feature Dilution of Precision): In addition to the number and quality of the features, the distribution of the features in the space needs to be investigated. $FDOP$ and $angular\_entropy$ factors which are explained next, investigate and quantify this aspect of the layout criterion. $FDOP$ is inspired by PDOP (position dilution of precision) of GNSS system. In GNSS systems, when visible navigation satellites are close together in the sky, the geometry is said to be weak and the PDOP value is high; when far apart, the geometry is strong and the PDOP value is low. Using this idea, we defined FDOP as a measure to evaluate how much the features in the local vicinity range are evenly distributed in 3D space considering their x, y, and z position. Assume that the position of the i_{th} ND feature in the local vicinity of sample point $P = \{X, Y, Z\}$ is shown as $ND_i = \{x_i, y_i, z_i\}$. $FDOP$ is calculated as follows:

$$FDOP = 1/\sqrt{\tilde{\sigma}_x^2 + \tilde{\sigma}_y^2 + \tilde{\sigma}_z^2}, \tag{5}$$

where $\tilde{\sigma}_x^2$, $\tilde{\sigma}_y^2$, and $\tilde{\sigma}_z^2$ are the variance of the x, y, and z of the features and calculated as follows:

$$\begin{aligned}
\tilde{\sigma}_x^2 &= \frac{1}{n}\sum_{i=1}^{n}(x_i - X)^2, \\
\tilde{\sigma}_y^2 &= \frac{1}{n}\sum_{i=1}^{n}(y_i - Y)^2, \\
\tilde{\sigma}_z^2 &= \frac{1}{n}\sum_{i=1}^{n}(z_i - Z)^2,
\end{aligned} \tag{6}$$

If features are distributed uniformly in the 3D space, the values of $\tilde{\sigma}_x^2$, $\tilde{\sigma}_y^2$, and $\tilde{\sigma}_z^2$ are high and therefore, $FDOP$ is low. In Fig. 5 several scenarios for FDOP are shown. In the most right figure, the distribution of the features are more even, and $FDOP$ becomes smaller.

2) angular_entropy: In addition to the $FDOP$, $angular\_entropy$ is defined to quantify the distribution of the features and show how even the features in the local vicinity range surround the sample point P. In fact, $angular\_entropy$ investigate the distribution of factors from different aspect and fills some shortness of $FDOP$. To calculate $angular\_entropy$ of the features in the local vicinity, all features are divided into m bins based on their azimuth angle (θ) to the sample point P (Fig. 6). Based on this histogram, angular entropy is calculated as follows:

$$H_{angular} = -\sum_{i=1}^{m} P(\theta_i)\left(log_2 P(\theta_i)\right), \tag{7}$$

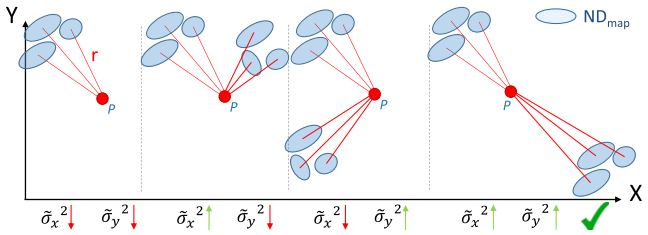

Fig. 5. Comparison of the layout (distribution) of the features with lateral and longitudinal *FDOP*.

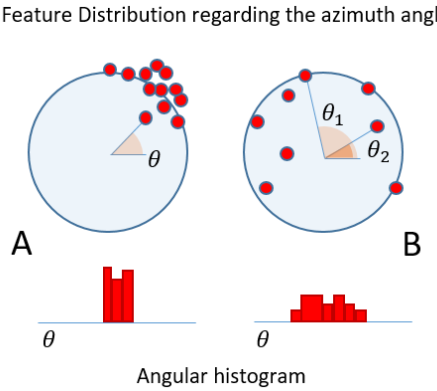

Fig. 6. Angualr entropy of two scenarios. In A, featuers are all palcesd in one corner, therefore some bins of the histogram fills more than the others. The entropy of B is higher than A as the histogram of B distributed more evenly.

where $H_{angular}$ is $angular\_entropy$, m is the number of bins which is 90, and $P(\theta_i)$ is the probability of occurrence of i_{th} bin (azimuth angle $= \theta_i$) and calculated using following formula:

$$P(\theta_i) = \frac{h(\theta_i)}{\sum_{i=1}^{m} h(\theta_i)}, \tag{8}$$

where $h(\theta_i)$ is the number of features in the i_{th} bin.

In Fig. 6 two distribution of features in the local vicinity is shown. If the features distributed evenly (Fig. 6 (B)), the histogram bins are filled evenly and the results of entropy over this histogram ($angular\_entropy$) become higher. In Fig. 6 (A), the features are distributed more evenly compared to B, therefore $H_{angular}[B] > H_{angular}[A]$.

The more features are distributed uniformly around the vehicle, the higher $angular\_entropy$ become. Higher $angular\_entropy$ value may have positive effects on the localization accuracy.

3) normal_entropy: One of the important questions that should be answered in order to evaluate the *layout* criterion of the map is that, is the layout of the environment capable of providing features for both longitudinal and lateral positioning. In Fig. 7, two layouts of the buildings are shown. In Fig. 7 (A) two walls are parallel, and both are only capable of obtaining the position of the vehicle in the longitudinal direction. Therefore, the error of localization in the longitudinal direction (perpendicular to the normal of the wall) is more than the error in the lateral direction. However, in Fig. 7 (B), the layout of the walls are changed and they become perpendicular to each other. In this case, each wall is capable of obtaining a position in a different direction.

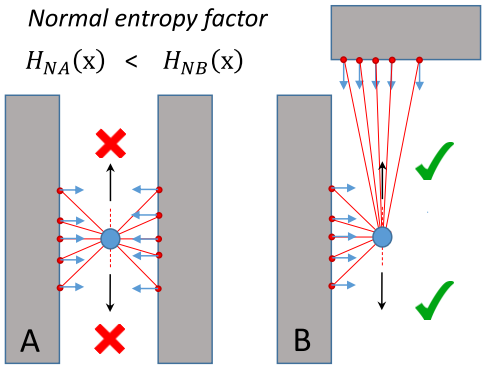

Fig. 7. Normal entropy for two different layout of the buikldings. In A, both walls are facing to the same direction. As there is no any lateral featuers here, the self-localization has longitudinal error. however in B, walls are facing to the different direction. In B normal entropy is higher and error is lower.

According to this, in the matching based methods, the uncertainty of the positioning is related to the features facing direction. Consequently, if the features in the local vicinity face more directions, the positioning uncertainty decreases. To quantify the degree of dispersity of the features' facing direction, *normal_entropy* is used. To calculate *normal_entropy*, first, normal of each feature in the local vicinity range are calculated. Then the azimuth angle ϕ and elevation angle Ψ of the normal is calculated. According to ϕ and Ψ, the normals are stored in $m \times m$ bins histogram.

From this histogram, *normal_entropy* is calculated as follows:

$$H_{normal} = -\sum_{i=1}^{m}\sum_{j=1}^{m} P\left(\emptyset_i, \Psi_j\right)\left(log_2 P\left(\emptyset_i, \Psi_j\right)\right), \quad (9)$$

where H_{normal} is *normal_entropy*, m is the number of ϕ bins and Ψ bins, and $\mathrm{P}\left(\emptyset_i, \Psi_j\right)$ is the probability of occurrence of normal with the azimuth angle and elevation angle of $\emptyset_i$ and Ψ_j respectively and calculated using the following equation:

$$P\left(\emptyset_i, \Psi_j\right) = \frac{h\left(\emptyset_i, \Psi_j\right)}{\sum_{i=1}^{m}\sum_{j=1}^{m}\left(\emptyset_i, \Psi_j\right)}, \quad (10)$$

where $h\left(\emptyset_i, \Psi_j\right)$ is the value of the (i_{th}, j_{th}) bin of the normal angle histogram. The higher value for the *normal_entroy* means more even distribution of normal in the histogram and results in better for self-localization.

In this paper, *normal_entropy* is calculated for three different number of bins ($m = 8, 16, 90$). These factors are *normal_H_8*, *normal_H_16*, and *normal_H_90*.

4) r_average: One of the important aspects, when the layout of the features are considered, is the distance of the features to the sensor. In map-matching based on LiDAR, as this distance increases, input scan becomes sparser and the details vanish. Therefore it is better to have closer features to the center of the sensor for the localization. In order to investigate and quantify this characteristic of the map, *r_average* factor is proposed. *r_average* is the average of the distance to the features in the local vicinity from the point P (center of local vicinity) and calculated as follows:

$$r_{average} = \sum_{i=1}^{m} r_i, \quad (11)$$

where r_i is the distance of i_{th} feature to the sample point P.

C. Quantification of Local Similarity

In map-matching methods, the presence of similarities in the map makes local optimum for matching as the matching process suffers from poor correspondence finding. For investigating *local similarity* phenomena in the map, three factors are defined. These factors are *score_entropy*, *pfh_similarity*, and *Battacharyya_similarity*. Among these, *score_entropy* investigates the similarity in the local vicinity region considering whole features together while the latter two investigates similarity more locally.

1) score_entropy: In order to calculate the degree of similarity in the local vicinity region, an original point cloud of the map for this region S_{map} and map features for the same region f_{map} is used. S_{map} is iteratively moved over the f_{map} in x and y direction with vector $\vec{v}$ and the score of each matching is calculated as follows:

$$score\left(\vec{v}\right) = -\sum_{i=1}^{n} \tilde{p}(T(\vec{v}, S_{map}(i))), \quad (12)$$

where $S_{map}(i)$ is the i_{th} point in the S_{map} point cloud, $\tilde{p}$ is the simplified log-likelihood function of the nearest feature to the point $S_{map}(i)$ in the f_{map} and T is the transformation function which transforms $S_{map}\,(i)$ with the 2D transformation vector $\vec{v}\,(x, y)$ [6]. Entropy over this score function is calculated as follows:

$$H_{score} = -\sum\sum_{\vec{v}} P(score(\vec{v}))log_2(P(score(\vec{v}))), \quad (13)$$

where H_{score} is *score_entropy* and $P(score(\vec{v}))$ is the probability of having $score(\vec{v})$ in the matching of S_{map} over f_{map}. Range and steps of $\vec{v}$ can be determined based on the error of an initial guess. In this paper, its range is $[-2m, 2m]$ for both direction x and y with steps of $0.2m$. If local vicinity has similar features (high local similarity), when S_{map} is moved over f_{map}, more score peaks happen. In fact, these score peaks are the local optimums for the matching process and the cause of the localization error. More score peaks make $P(score(\vec{v}))$ more even and as a result *score_entropy* becomes higher.

2) pfh_similarity: In addition to the *score_entropy* which investigates the local similarity more widely, *pfh_similarity* factors consider more local similarities in the map. In *pfh_similarity* characteristics of each feature are obtained as a histogram using PFH (point feature histogram) method [22], [23]. Original PFH method is used for defining the characteristics of each point in the point cloud based on the relationships between the points in the k-neighborhood and their estimated surface normal. However, in this paper, PFH is directly applied to the features. The characteristics of each feature are obtained based on the features in the 5m range. By having a histogram for each feature, and comparing them with each other the local similarity of the features can be obtained. For each feature, its histogram is compared with other histograms in the range of 5m. In this paper, for comparison of two histograms, Manhattan distance is used.

Manhattan distance of two histogram pfh_a and pfh_b are calculated using the following formula:

$$L^1\left(pfh_a, pfh_b\right) = \sum_{i=1}^{m} \left|pfh_a(i) - pfh_a(i)\right|, \quad (14)$$

where pfh_a(i) is the i_{th} the bin of the pfh_a histogram and m is the total number of bins. Using (14), the similarity of feature a with its neighbors in the 5m range is obtained as follows:

$$pfh_{similarity}\,(a) = \sum_{k=1}^{n} L^1(pfh_a, pfh_k), \qquad (15)$$

where pfh_a is the histogram of feature a, pfh_k is the histogram of k_{th} feature and m is the number of neighbors in 5m range of feature a. Accordingly, the total similarity value of the local vicinity region of point p is obtained by summing $pfh_{similarity}$ of all features as follows:

$$pfh\_similarity_{total} = \sum_{k=1}^{N} pfh_{similarity}\,(k), \qquad (16)$$

where N is the total number of features in the local vicinity of sample point P.

3) pfh_entropy: In the PFH histogram, if the bins of the histogram of a feature are filled evenly, it means that this feature is surrounded by different characteristics of features. In other words, similarity around this feature is very low. Contrary to this, if a feature is surrounded by features with similar characteristics, specific bins of its histogram are filled. In order to see whether the surrounding features are similar or not, entropy is used. Similar to other factors, the histogram of each feature is assumed as a probability distribution and entropy is calculated. This factor is called $pfh\_entropy$. High similarity in the map results in low $pfh\_entropy$ value.

4) battacharyya_similarity: The other approach for calculating the local similarity is to calculate the Bhattacharyya distance for each pair of the map features. Bhattacharyya distance is used to calculate the similarity (distance) between two discrete and continuous probability distribution function. Bhattacharyya distance $DB(p,q)$ between two normal distribution $p(x) = N\left(x \mid \mu_p, \Sigma_p\right)$ and $q(x) = N(x|\mu_q, \Sigma_q)$ is defined as:

$$DB(p,q) = \frac{1}{8}(\mu p - \mu q)^T \left(\frac{\Sigma p + \Sigma q}{2}\right)^{-1} (\mu p - \mu q) + \frac{1}{2} log\left(\frac{\left|\frac{\Sigma p + \Sigma q}{2}\right|}{|\Sigma p|^{0.5}\,|\Sigma q|^{0.5}}\right), \qquad (17)$$

Using equation (17), $battacharya\_similarity$ for the map around the sample point P is calculated as follows:

$$Similarity_{Batt} = \sum_{i=1}^{N} \sum_{j \epsilon allneighborsofi} DB(i,j), \qquad (18)$$

D. Quantification of Representation Quality

In the *representation quality* criterion, the amount of information loss in the map should be investigated. This information loss shows how much details of the map elements are removed due to abstraction. Representation quality criterion is quantified using $mahalanobis\_distance$ and $euclidean\_distance$ factor.

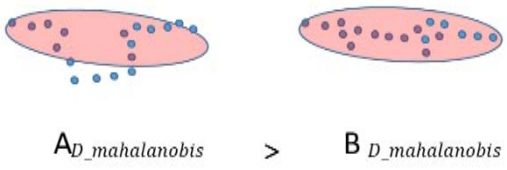

Fig. 8. Effect of Mahalanobis distance for two different features.

1) mahalanobis_distance: To formulate this, we need to evaluate how much current map elements are close to the original raw point cloud of the environment. Therefore, we need to have the original raw point cloud of the map. For each point in the point cloud, distance to nearest map elements is calculated. In the case of normal distribution map, this distance should be calculated using Mahalanobis distance as it considers the scaling parameters. Mahalanobis distance d_{mah} of the point x_i to the ND_j is calculated as follows:

$$d_{mah}\,(x_i) = \sqrt{(x_i - \mu_j)^T \sum_J^{-1} (x_i - \mu_j)}, \qquad (19)$$

where μ_j and $\sum_j$ are mean and covariance of the nearest ND_j to x_i respectively. Average of $d_{mah}\,(x_i)$ for all points in local vicinity range is considered as $mahalanobis\_distance$ factor and calculated as follows:

$$D_{mahalanobis} = \frac{\sum_{i=1}^{n} d_{mah}(x_i)}{n}, \qquad (20)$$

where n is the total number of points in the local vicinity region of the sample point P. In Fig. 8, two map elements with their corresponding point cloud are shown. Blue points are original point cloud and red ellipsoids are their corresponding map elements. In Fig. 8(B), the map is more close to the original point cloud than Fig. 8(A). Therefore, the $mahalanobis\_distance$ factor of A is higher than B.

2) euclidean_distance: By changing the distance metric in (20), the $Euclidean\_distance$ factor can be calculated as follows:

$$D_{mahalanobis} = \frac{\sum_{i=1}^{n} d_{Euclidean}(x_i)}{n}, \qquad (21)$$

where n is the total number of points in the local vicinity region of the sample point P, $d_{Euclidean}\,(x_i)$ is the Euclidean distance of a point x_i of point cloud map to its nearest map element.

The list of factors is shown in Table II.

VI. Estimation of Self-Localization Error

A. Methodology

The capability of the map for an accurate self-localization can be evaluated for each sample point in the map by estimating the localization error using the map factors. If the estimated error is higher than the predefined threshold, then map features in the surrounding of that sample point are not capable of performing an accurate map-matching. The results of this evaluation in each point can be used for defining abstraction (downsampling) ratio of the map, or to assist the self-localization process by suggesting where to trust more on map-matching.

TABLE II
LIST OF MAP FACTORS FOR EACH CRITERION WITH THEIR DESCRIPTION AND FORMULA

Feature sufficiency	$feature\_cnt$	Count of ND features in the local vicinity	-	
	$dimention\_cnt$ $(D1\_cnt, D2\_cnt, D3\_cnt)$	Count of 1D, 2D and 3D ND features separately	$a_{1D} = \frac{\sigma_1 - \sigma_2}{\sigma_1}, a_{2D} = \frac{\sigma_2 - \sigma_3}{\sigma_1}, a_{3D} = \frac{\sigma_3}{\sigma_1}$	
	$dimension\ ratio$ $(D1\_ratio, D2\_ratio,$ $D3\_ratio)$	Ratio of 1D, 2D, and 3D ND features	$Dm_{ratio} = \frac{Dm_{cnt}}{feature_{cnt}}$	
	$occupancy\_ratio$	Ratio of occupied cells in depth image of the local vicinity	$occupancy_{ratio} = \frac{occupied\_cell}{all\_cells}$	
Layout	$FDOP$	Describes the relative position of the ND features. If features are spread apart, FDOP is good	$FDOP = 1/\sqrt{\tilde{\sigma}_x{}^2 + \tilde{\sigma}_y{}^2 + \tilde{\sigma}_z{}^2}$	
	$normal\_entropy$ $(normal\_H\_m,$ $m = 8, 16, 90)$	Entropy NDs' normal directions	$H_{norm} = -\sum_{i=1}^{b} P_{norm}(i)(log_2 P_{norm}(i))$	
	$angular\_entropy$	Entropy of availability of ND features in 360 degree	$H_{ang} = -\sum_{i=1}^{b} P_{ang}(i)(log_2 P_{ang}(i))$	
	$r\_average$	Average distance of NDs from vehicle position	$r_{average} = \sum_{i=1}^{m} r_i$	
Local similarity	$score\_entropy$	Entropy of score function of registration	$H_{score} = -\sum_{\vec{v}} P_{score}(\vec{v}) log_2 (P_{score}(\vec{v}))$	
	$pfh\_similarity$	Similarity of features using point feature histogram (PFH)	$pfh\_similarity_{total} = \sum_{k=1}^{N} pfh_{similarity}(k)$	*
	$pfh\_entropy$	Similarity of features by calculating entropy over point feature histograms	Entropy of PFH	*
	$bhattacharya\_similarity$	Similarity of NDs calculated based on Battacharya distance	$Similarity_{Batt} = \sum_{i=1}^{N} \sum_{j \in \text{ all neighbors of } i} DB(i, j)$	*
Representation quality	$mahalanobis\_distance$	Mahalanobis distance between point cloud map and ND map	$D_{mahalanobis} = \frac{\sum_{i=1}^{n} d_{mah}(x_i)}{n}$	
	$euclidean\_distance$	Bhattacharyya distance between point cloud map and ND map	$D_{euclidean} = \frac{\sum_{i=1}^{n} d_{Euclidean}(x_i)}{n}$	*

* Factors newly proposed in this paper

In order to estimate the error of self-localization, from map factors, weights and effects of each factor in the error should be investigated and defined. In our previous work, the weights of the factors are modeled by using PCR (Principle Component Regression) and actual error. In this work, the contribution of the factors is learned by RFR (Random Forest Regression). To do this, for a numerous number of points on the map, the map factors for each criterion are calculated. For the same point, actual localization error is calculated and RFR is trained with this data.

In this paper, the normal distribution map is evaluated. Therefore, the factors are calculated based on factors extracted from normal distribution map, and for the self-localization (map-matching), normal distribution transform is used. However, with the same methodology and map criteria, other formats of the map can be evaluated. In that case, map factors should be adapted to the new format and the error should be obtained with corresponding localization method.

B. Data

In order to train random forest regression, the actual map-matching error is collected for 5 paths in Shinjuku, Tokyo, Japan. These 5 paths are shown in Fig. 9. The total length of these 5 paths is 3.6 km. These paths are selected so that they contain different environments to make the estimation more robust to change of environment. The description of these paths is shown in Table III.

For each path, the point cloud map is generated using our MMS (mobile mapping system) shown in Fig. 10 and MMS registration framework in [15] which utilize airborne imagery for fine-tuning of the point cloud map gathered from MMS. In this MMS, integration of the stereo camera, IMU,

TABLE III
DESCRIPTION OF ALL 5 PATHS

Path #	Map format	Length	Total number of sample points	Description
Path I	Normal distribution map (2.0m grid size)	1.16 km	1163 points	Narrow and dense shopping area / wide roads with hedges and trees along the roadside / passing by several skyscrapers
Path II		0.73 km	734 points	Narrow road / wide pavements populated with bollards, lamp posts, and trees / passing by number of skyscrapers,
Path III		0.27 km	269 points	Residential area / short tunnel in the beginning / one side is totally filled with residential area and the other side is full of trees
Path IV		0.65 km	652 points	Fully contained with narrow and dense shopping area
Path V		0.79 km	794 points	Narrow and dense shopping are / multi-lane roads /passing by number of skyscrapers / passing under several bridge
Total		**3.60 km**	**3612 points**	

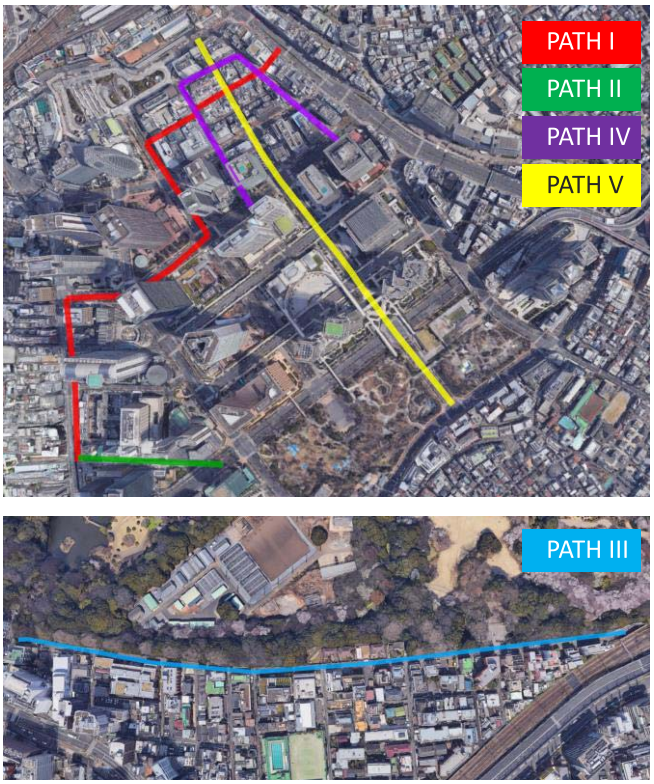

Fig. 9. Experimental area in Shinjuku, Tokyo.

CAN and Velodyne VLP-16 (Fig. 10 (center)) are used to get the local movement of the MMS. However this local movement of the vehicle cannot be used a final position of the MMS as it has an accumulative error and it is not accurate enough for mapping. Therefore, position of the MMS should be rectified with some reference data. In this mapping framework, airborne imagery are used as a reference data [15]. Sweep from SICK laser scanner and Velodyne VLP-16 in back of the car (Fig. 10 (left)) are used to capture the road markings. These road markings are then matched with the road markings extracted from airborne imagery of the same region using sliding window matching strategy in [15]. In this framework, GPS is only used for rough estimation of the MMS position. Using this mapping framework, generated map has both higher precision and higher global accuracy comparing to other conventional MMS systems. Later, from this point cloud map, normal distribution map with 2.0m grid was made.

To get actual localization error, our mapping system moves through these 5 paths and the sweeps from VLP-16 mounted on the top of the vehicle is collected (Fig. 10 (center)). To avoid the scan distortion due to the movement of the vehicle, the vehicle's velocity was below 2 m/s while the frequency of the laser scanner was set to 20 Hz which limits the distortion in each scan to less than 10 cm. For each meter, around 10 to 15 sweeps (samples) are gathered. In most of the cases, environment changes for less than 1.0 m shift are very small. Therefore, sample point data is obtained with an interval of 1.0 m. Totally, 3612 sample points are gathered.

C. Localization Error

To calculate the 3D localization error for a sample point P, map-matching is performed from different initial guesses around P. These initial guesses (particles) are distributed around the sample point P within the range of 2.0m and 0.2m of intervals. We call these initial guesses around the sample point P, initial guess particles. The total number of initial guess particles for each point is 317. The range of distribution of initial guess particles can be obtained by the average error of the GPS system or other modality used for obtaining the rough location of the vehicle, however, in this paper 2.0m is used based on experience. In order to have a more accurate estimation of the error, the interval can be smaller.

Having 317 initial guess particles means for each sample point P, map-matching is performed 317 times. Fig. 11 shows the distribution of sample points and their corresponding 3D localization errors. 3D self-localization error for sample point P is then calculated by averaging the errors obtained by map-matching with these 317 initial guesses.

For calculating an actual self-localization error in each sample point P, an accurate ground truth should be obtained. In this study, we focused on the urban areas of Tokyo where GPS signals are mostly blocked or received from multi-path, which makes the error more than several meters. RTK-GPS also cannot be used in these areas and our investigation shows that RTK-GPS also has more than a meter error in these areas [15].

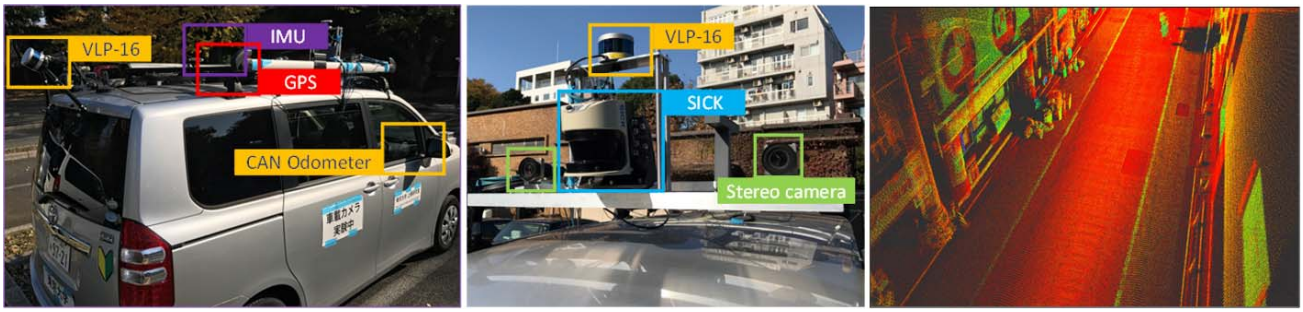

Fig. 10. Experimental vehicle and point cloud map. Normal distribution map is generated using this point cloud map.

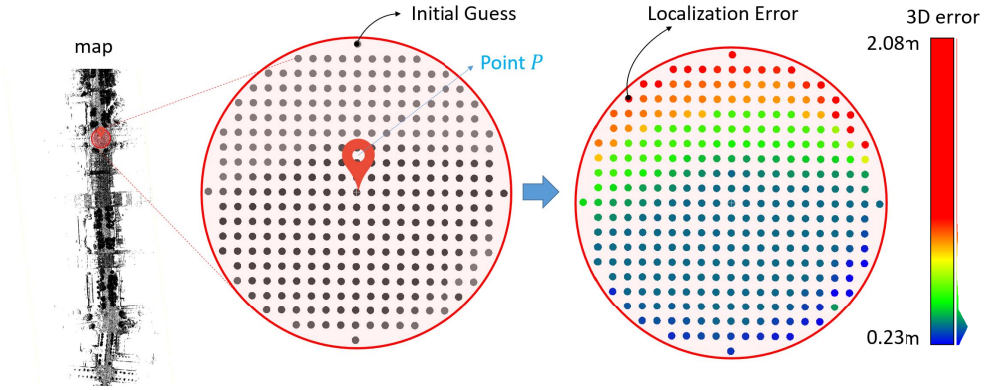

Fig. 11. Mean error calculation for each sample point P. For each sample point, map-matching is performed with different initial guesses. The results of each matching is shown in the right circle. Color of matching shows the localization error. Color close to blue shows lower error comparing to red.

Therefore, in this study, ground truth is obtained by iterative scan-matching strategy in two steps. In the first step, the input scan is matched to the normal distribution map with a smaller grid size (higher resolution) than 2.0m. Here we use a normal distribution map with 1.0m grid size for ground truth. Normal distribution map with 1.0m grid size needs more memory and processing time for scan-matching, but it has lower information loss in each grid compared to 2.0m grid size and higher matching accuracy. This position is then used as rough ground truth in the second step.

In the second step, in order to evaluate whether the matching result stucks in local optimum or not, near this rough ground truth (within range of 50cm), particles are distributed with a 5cm interval. Then matching score of each particle is calculated and a particle with maximum score becomes ground truth. By doing this two-step strategy, accurate ground truth can be obtained.

D. Modeling Error Using Map Factors

Since localization error is a continuous value, a regression model is required. For this problem, the number of samples is relatively low and the dimensionality is low (20 factors). Some factors are more important than the other and they also have some degree of correlation. Considering these conditions, an ensemble learning method is appropriate. In this study, random forest regression is used as the prediction model for localization error.

Random forest regression is an ensemble learning method that uses multiple decision trees and bootstrap aggregation to form a prediction. Random forest regression aims to reduce variance in predictions while maintaining a low bias, thus controlling overfitting. For the implementation, scikitlearn (a machine learning Python library) is used. As random forest regression is a supervised learning method, a training dataset is required. This training dataset is a set of the calculated value of the factors and actual errors. The input to the RFR is values of each factor and the output (label) is an actual error.

In this study, other regression models were also tested (Multi-layer perceptron, Support vector machine, Stochastic gradient descent), but did not achieve comparable results without further intensive manual tuning of the models.

VII. Results And Discussion

Error estimation is evaluated by dividing the total sample points in 75% of training data and 25% of test data. With the first randomly selected 75% (2709 points) of the sample points, the random forest regression is trained and with the rest of 25% (903 points) the estimation is evaluated.

The actual 3D Error of some sample points in all 5 paths are shown in Fig. 12. As expected from Table I, Error of Path III are higher than other paths. These 3D errors are estimated and evaluated in this section.

Error estimation is evaluated in four steps. The first evaluation is done by only considering the factors of the feature sufficiency (FS) criterion. As shown in Table IV, for estimating error using only feature sufficiency criteria, 8 factors are used. Factor importance in Table IV shows among these factors, $D1\_ratio$ has more effect on the localization error. In other words, in the feature sufficiency criterion, the ratio of pole-like features in the surrounding is very important. By only considering factors of FS, 59.4 % of the cases, localization error can be estimated with a prediction error of less than

TABLE IV
RESULTS OF 3D ERROR ESTIMATION FOR DIFFERENT COMBINATION OF FACTORS

Criteria	*Factor*	*FS*	*FS + RQ*	*FS + RQ + LY*	*FS + RQ + LY + LS*	*Importance**
Feature sufficiency	*feature_count*	x	x	x	x	0.01
	D1_count	x	x	x	x	0.01
	D2_count	x	x	x	x	0.02
	D3_count	x	x	x	x	0.02
	D1_ratio	x	x	x	x	0.05
	D2_ratio	x	x	x	x	0.01
	D3_ratio	x	x	x	x	0.02
	occupancy_ratio	x	x	x	x	0.04
Representation quality	*euclidan_dist*		x	x	x	0.03
	mahala_dist		x	x	x	0.02
Layout	*r_ave*			x	x	0.02
	normal_H_8			x	x	0.05
	normal_H_16			x	x	0.05
	normal_H_90			x	x	0.03
	layout_ang_H			x	x	0.16
	FDOP			x	x	0.03
Local similarity	*score_H*				x	0.02
	pfh_sim_H				x	0.10
	pfh_sim_mnht				x	0.06
	bhatt_sim				x	0.23
Pred error < 5 cm		81.8 %	83.9 %	84.7 %	85.7 %	
Pred error < 2.5 cm		**59.4 %**	**60.1 %**	**63.6 %**	**64.1 %**	
RMSE (cm)		5.75 cm	5.40 cm	4.93 cm	4.83 cm	
MAE (cm)		3.39 cm	3.14 cm	2.92 cm	2.83 cm	

**This column shows the factor importance for the scenario in which all factors are used for training.*

2.5cm. RMSE of the prediction of error using only FS factors are 5.75 cm.

The second evaluation was done by adding the representation quality criterion (RQ) to the estimation. By adding the factors for these criteria, a minor improvement can be achieved. The reason behind why the improvement is minor is that the representation quality factor has some overlap with the factors defined in FS. For example, dimension ratio factors have a high correlation with representation quality because the ratio of $3Dfeatues$ can affect the representation quality.

In the third step, the factors of FS, RQ, and layout (LY) criterion are considered all together. In Table IV, by checking the importance of the factors for the layout criterion, $angualr\_H$ plays higher roll in the localization error. Among other factors importance of this factor is higher as well. It means, in the localization error, the angular distribution of the features among the vehicle is very important for the accurate localization. By adding the factors of LY to estimation process, RMSE of estimation can be reduced to 4.93 cm. In 63.6 % of the test data, we can predict the localization error with the prediction error of 2.5 cm. This improvement shows the role of layout criterion comparing to the RQ criterion.

And finally, by adding factors for local similarity (LS) criterion, all 20 factors are used for estimating the localization error. Among the factors for local similarity criterion, $bhatt\_sim$ and Pfh_sim_H which are proposed in this paper has higher importance comparing to $score\_H$. By adding the factors of LS, RMSE of prediction can be reduced to 4.83 cm. According to the results of Table IV, using all 20 factors of FS, RQ, LY, LS, in 85.7 % of the cases, a map-matching error can be estimated with the estimation error less than 5 cm, and in 64.1 % for less than 2.5 cm error. Mean absolute error (MAE) of the prediction is 2.83cm.

Fig. 13 shows the actual 3D localization error for some sample points from all 5 paths compared to modeled error from random forest regresion. As expected from Table IV, in most of the cases, modeled error can explain the actual 3D error with a very good accuracy.

Fig. 14 shows the factor importance of all 20 factors for four criteria. Each criterion is shown with distinguished color. Among four criteria, factors for LS and LY play more roll to the localization error. It means the layout of the features and the local similarity in the map are very important for the accurate self-localization. Among all criteria, $bhatt\_sim$ has more effect on the localization accuracy.

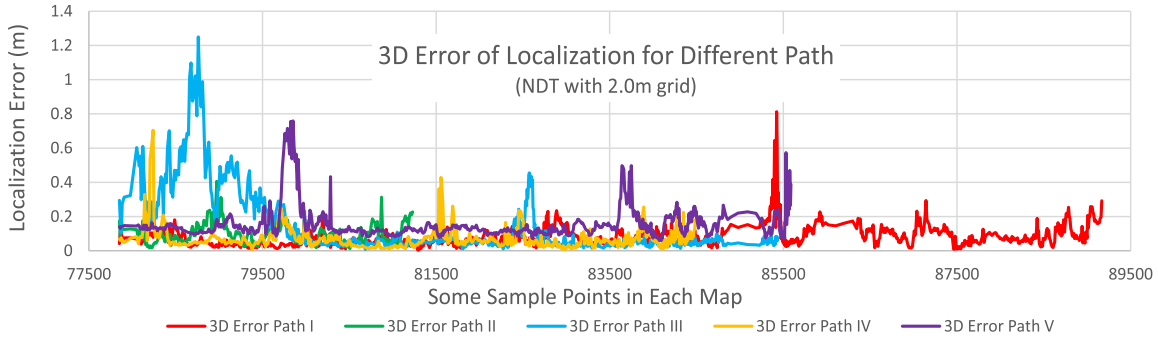

Fig. 12. 3D error of self-localization for different paths. The color of graphs are same as the color of path in Fig. 9. Horizontal axis shows the ID of sample points in the map.

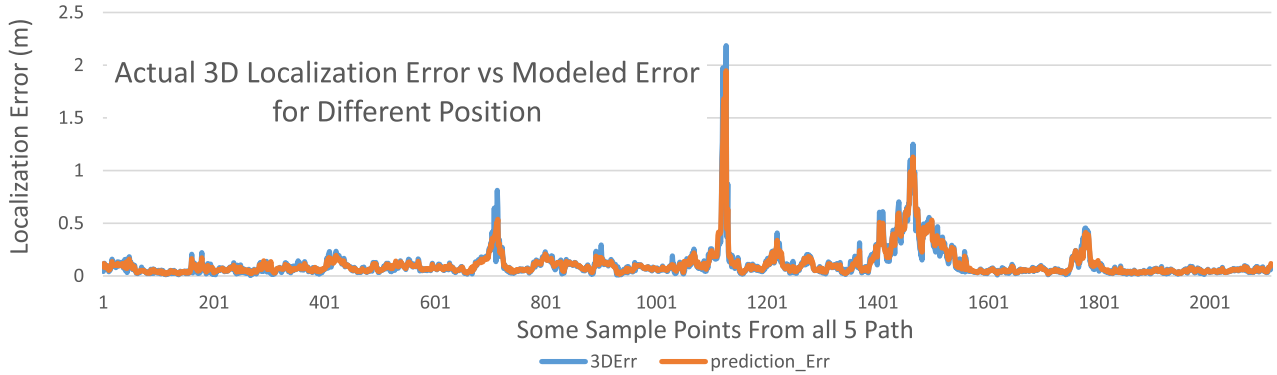

Fig. 13. Actual 3D error vs modeled error using random forest regression. Blue graph is actual 3D error and orange is modeled error.

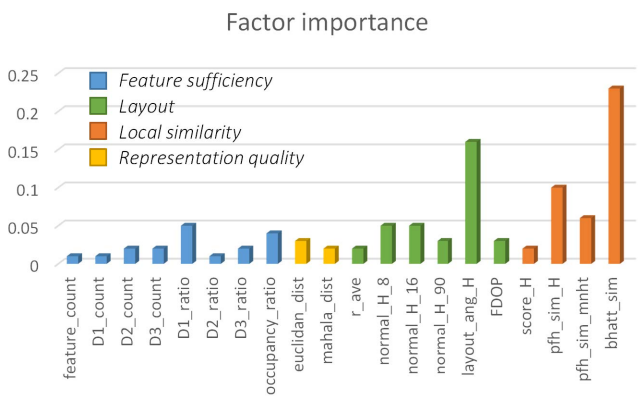

Fig. 14. Factor importance for different map criteria. Higher value shows more contribution to the error.

VIII. Conclusion

In this work, firstly, error sources of the self-localization based on map matching have been discussed. Secondly, we have focused on the map, as one of the high potential sources of error in the map-matching, and requirements of the map for highly accurate self-localization are defined using four map criteria. These criteria are *feature sufficiency*, *layout*, *local similarity*, and *representation quality* of the map. To quantify each of these criteria, several map factors have been defined. In this work, three new factors have been proposed to better quantify the local similarity criterion of the map. These factors are *pfh_similarity*, *pfh_entropy*, and *battacharya_similarity*. Thirdly, a framework based on random forest regression has been used to estimate the localization error by solely investigating the map.

Random forest regression has been trained with totally 3.6 km of the data gathered in Shinjuku, Tokyo. In these 3.6 km, 3612 positions are selected and actual self-localization errors are calculated for them. Among these positions, 75% used for training the random forest regression, and the rest of 25% was used to test the error estimation framework. Results showed that factor importance of newly defined factors, *pfh_similarity*, *pfh_entropy*, and *battacharya_similarity* are 0.06, 0.10, and 0.23 respectively. This shows their high contribution in estimating the error. Experiments have shown that using the proposed framework, in 64.1% of the cases, localization error within the prebuilt map can be predicted with less than 2.5cm of estimation error. RMSE of error estimation when all 20 factors are used is 4.83cm which is promising.

In this work, we have shown that by investigating a total of 20 factors, which are solely extracted from the map, the self-localization error can be predicted. This error estimation can be used to evaluate the prebuild map, considering its capability of an accurate self-localization. Also, it can be used to determine the best format and abstraction ratio of the map. On the other hand, estimation of error using this framework can be used for sensor fusion in the localization phase.

References

[1] C. Fernández, D. Fernández-Llorca, and M. A. Sotelo, "A hybrid vision-map method for urban road detection," *J. Adv. Transp.*, vol. 2017, Oct. 2017, Art. no. 7090549. [Online]. Available: https://www.hindawi.com/journals/jat/2017/7090549/

[2] M. Xie, H. Chen, X. F. Zhang, X. Guo, and Z. P. Yu, "Development of navigation system for autonomous vehicle to meet the DARPA urban grand challenge," in *Proc. IEEE Intell. Transp. Syst. Conf.*, Sep. 2007, pp. 767–772, doi: 10.1109/ITSC.2007.4357742.

[3] H. G. Seif and X. Hu, "Autonomous driving in the iCity–HD maps as a key challenge of the automotive industry," *Engineering*, vol. 2, no. 2, pp. 159–162, Jun. 2016, doi: 10.1016/J.ENG.2016.02.010.
[4] F. Jomrich, A. Sharma, T. Rückelt, D. Burgstahler, and D. Böhnstedt, "Dynamic map update protocol for highly automated driving vehicles," in *Proc. 3rd Int. Conf. Vehicle Technol. Intell. Transp. Syst.*, vol. 2, Sep. 2017, pp. 68–78, doi: 10.5220/0006279800680078.
[5] E. Javanmardi, M. Javanmardi, Y. Gu, and S. Kamijo, "Evaluation of digital map ability for vehicle self-iocalization," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Jun. 2018, pp. 708–715, doi: 10.1109/IVS.2018.8500389.
[6] E. Javanmardi, M. Javanmardi, Y. Gu, and S. Kamijo, "Factors to evaluate capability of map for vehicle localization," *IEEE Access*, vol. 6, pp. 49850–49867, 2018, doi: 10.1109/ACCESS.2018.2868244.
[7] J. Yang, H. Li, and Y. Jia, "Go-ICP: Solving 3D registration efficiently and globally optimally," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2013, pp. 1457–1464, doi: 10.1109/ICCV.2013.184.
[8] J. Serafin and G. Grisetti, "NICP: Dense normal based point cloud registration," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Sep. 2015, pp. 742–749, doi: 10.1109/IROS.2015.7353455.
[9] M. Magnusson, N. Vaskevicius, T. Stoyanov, K. Pathak, and A. Birk, "Beyond points: Evaluating recent 3D scan-matching algorithms," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2015, pp. 3631–3637, doi: 10.1109/ICRA.2015.7139703.
[10] A. Hornung, K. M. Wurm, M. Bennewitz, C. Stachniss, and W. Burgard, "OctoMap: An efficient probabilistic 3D mapping framework based on octrees," *Auto. Robots*, vol. 34, no. 3, pp. 189–206, Feb. 2013, doi: 10.1007/s10514-012-9321-0.
[11] M. Schreiber, C. Knoppel, and U. Franke, "LaneLoc: Lane marking based localization using highly accurate maps," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Jun. 2013, pp. 449–454, doi: 10.1109/IVS.2013.6629509.
[12] R. W. Wolcott and R. M. Eustice, "Robust LIDAR localization using multiresolution Gaussian mixture maps for autonomous driving," *Int. J. Robot. Res.*, vol. 36, no. 3, pp. 292–319, Mar. 2017, doi: 10.1177/0278364917696568.
[13] E. Javanmardi, Y. Gu, M. Javanmardi, and S. Kamijo, "Autonomous vehicle self-localization based on abstract map and multi-channel LiDAR in urban area," *IATSS Res.*, vol. 43, no. 1, pp. 1–13, Apr. 2019, doi: 10.1016/j.iatssr.2018.05.001.
[14] N. Kajiwara, J. Takiguti, R. Hirokawa, and Y. Shima, "Landmark update of GPS/INS for mobile mapping system using a nearly horizontal single camera and 3D point-cloud," in *Proc. Int. Symp. GPS/GNSS*, vol. 1, 2008, pp. 1130–1138.
[15] M. Javanmardi, E. Javanmardi, Y. Gu, and S. Kamijo, "Towards high-definition 3D urban mapping: Road feature-based registration of mobile mapping systems and aerial imagery," *Remote Sens.*, vol. 9, no. 10, p. 975, Sep. 2017, doi: 10.3390/rs9100975.
[16] N. Akai, L. Y. Morales, E. Takeuchi, Y. Yoshihara, and Y. Ninomiya, "Robust localization using 3D NDT scan matching with experimentally determined uncertainty and road marker matching," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Jun. 2017, pp. 1356–1363, doi: 10.1109/IVS.2017.7995900.
[17] O. Vysotska and C. Stachniss, "Improving SLAM by exploiting building information from publicly available maps and localization priors," *PFG-J. Photogramm., Remote Sens. Geoinf. Sci.*, vol. 85, no. 1, pp. 53–65, Feb. 2017, doi: 10.1007/s41064-017-0006-3.
[18] H. Merzic, E. Stumm, M. Dymczyk, R. Siegwart, and I. Gilitschenski, "Map quality evaluation for visual localization," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2017, pp. 3200–3206, doi: 10.1109/ICRA.2017.7989363.
[19] P. Barrios, M. Adams, K. Leung, F. Inostroza, G. Naqvi, and M. E. Orchard, "Metrics for evaluating feature-based mapping performance," *IEEE Trans. Robot.*, vol. 33, no. 1, pp. 198–213, Feb. 2017, doi: 10.1109/TRO.2016.2627027.
[20] M. Magnusson, A. Lilienthal, and T. Duckett, "Scan registration for autonomous mining vehicles using 3D-NDT," *J. Field Robot.*, vol. 24, no. 10, pp. 803–827, Oct. 2007, doi: 10.1002/rob.20204.
[21] D.-J. Jwo, "Efficient DOP calculation for GPS with and without altimeter aiding," *J. Navigat.*, vol. 54, no. 2, pp. 269–279, 2001. [Online]. Available: https://www.cambridge.org/core/journals/journal-of-navigation/article/efficient-dop-calculation-for-gps-with-and-without-altimeter-aiding/8FD31C94F27E922CD1401FD3EEA5BE70
[22] M. A. Savelonas, I. Pratikakis, and K. Sfikas, "Fisher encoding of differential fast point feature histograms for partial 3D object retrieval," *Pattern Recognit.*, vol. 55, pp. 114–124, Jul. 2016, doi: 10.1016/j.patcog.2016.02.003.
[23] R. B. Rusu, N. Blodow, and M. Beetz, "Fast point feature histograms (FPFH) for 3D registration," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2009, pp. 3212–3217.

Ehsan Javanmardi (Member, IEEE) received the M.E. degree in computer architecture from the Amirkabir University of Technology, Iran, in 2012, and the Ph.D. degree in information and communication engineering from The University of Tokyo, Japan, in 2018.

From 2014 to 2015, he was a Visiting Research Student with The University of Tokyo. He was also a Visiting Student Researcher with the University of California, Berkeley, from 2016 to 2017. He also completed the Graduate Program for Social ICT Global Creative Leaders with The University of Tokyo in 2018. He has been a Post-Doctoral Researcher with the Institute of Industrial Science, The University of Tokyo, since 2018. His research interests include intelligent vehicles, autonomous vehicle's self-localization, mobile mapping systems, ADAS map, mapping, sensor fusion, and vehicle perception. He was a recipient of the IEEE Intelligent Transportation Systems Society Best Student Paper Award in 2017.

Mahdi Javanmardi (Member, IEEE) received the M.Sc. degree from the Sharif University of Technology, Iran, in 2013, and the Ph.D. degree in information and communication engineering from The University of Tokyo, Japan, in 2017.

He was a Visiting Student Researcher with the University of California, Berkeley, from 2016 to 2017. He has also completed the Graduate Program for Social ICT Global Creative Leaders with The University of Tokyo, in 2017. He has been a Post-Doctoral Researcher with the Institute of Industrial Science, The University of Tokyo, since 2017. His research interests include localization and mapping for an autonomous vehicle, autonomous vehicle's perception, and computer vision. He is a member of the IEEE ITS Society.

Yanlei Gu (Member, IEEE) received the M.E. degree from the Harbin University of Science and Technology, China, in 2008, and the Ph.D. degree from Nagoya University, Japan, in 2012. He has been a Post-Doctoral Researcher with the Institute of Industrial Science, The University of Tokyo, since 2013. He became a Lecturer with Ritsumeikan University from 2019. His research interests include GNSS, computer vision, and deep learning and their applications to ITS. He is a member of the IEEE ITS Society. He has served as the Organizing Committee Member for IEEE ICVES2015 and ITSC2017.

Shunsuke Kamijo (Senior Member, IEEE) received the B.S. and M.S. degrees in physics and the Ph.D. degree in information engineering from The University of Tokyo, Tokyo, Japan, in 1990, 1992, and 2001, respectively.

He began working for Fujitsu Ltd., in 1992 as a Processor Design Engineer. He was an Assistant Professor from 2001 to 2002 and has been an Associate Professor since 2002. His research interests include computer vision, wireless communications, and their applications to ITS. His research focuses are autonomous vehicles, traffic video surveillance, traffic signal control, V2X communications, pedestrian, and car navigations. He is a member of the IEEE ITS Society, TRB, IEICE, and IATSS. He joined the International Program Committee of ITS World Congress in 2011. He has been a member of the Board of Governors of the IEEE ITS Society since 2015 and an Executive Committee Member of the Society since 2017. He has served as the Vice-Chairman of the Program Committee for the ITS World Congress Tokyo 213, a General Co-Chair for the IEEE ICVES2015 and ITSC2017, and the International Program Chair for the IEEE IV2017. He is an Editorial Board Member of the *International Journal on ITS Research* and *Multimedia Tools and Applications*.