2021 IEEE International Conference on Robotics and Automation (ICRA 2021)
May 31 - June 4, 2021, Xi'an, China

# HD Map Update for Autonomous Driving With Crowdsourced Data

Kitae Kim, Soohyun Cho, and Woojin Chung

*Abstract*— **Current self-driving cars can perform precise localization and generate collision-free trajectories using high definition (HD) maps which provide accurate road information. Therefore, keeping HD maps up to date is important for safe autonomous driving. In general, automotive HD maps are built by the use of expensive mapping systems. In addition, a lot of manual modifications are required in many cases. The conventional HD mapping cannot be frequently carried out due to the high cost.**

**In this work, we used a large amount of road data collected by crowdsourcing devices. Crowdsourcing devices consist of low-cost sensors. The devices are mounted on repeatedly traveling vehicles such as buses. Although collected data shows high uncertainty and low accuracy, a large amount of data can be obtained in a short time with low expense. We present a solution that keeps HD maps up to date by using crowdsourced data. The developed solution concentrates on landmark information among crowdsourced data and HD maps.**

**By using uncertainty information, we chose reliable observations for map updating. Observation learner algorithms were carefully designed under the consideration of differences between discrete and continuous landmarks. The triggering condition for the map update can be adjusted by the proposed update mode selection strategy. The proposed map updating scheme has been experimentally verified by the use of crowdsourced data collected from real road environments.**

## I. INTRODUCTION

Recent autonomous vehicles can estimate their precise positions and generate collision-free trajectories with high definition (HD) maps. Construction of HD maps became popular owing to the wide use of mobile mapping systems (MMS) [1]. However, frequently updating HD maps is difficult due to high construction cost of HD maps. In order to guarantee accurate localization and reliable navigation performances, up-to-date HD maps are essential.

Recently, crowdsourced data is receiving much attention for map updates as in [2]. Crowdsourced data is road observation data that has been collected by low-cost crowdsourcing devices. A crowdsourcing device typically includes a low-cost camera and a global navigation satellite system (GNSS) sensor. Crowdsourcing devices are installed on vehicles that frequently and repeatedly travel the same routes. Therefore, a large amount of environmental data can be easily obtained. The major drawback of the crowdsourced data is its high uncertainty. The challenge is appropriate update of HD maps by reflecting environmental changes while overcoming various uncertainties from low-quality observations.

Pannen *et al.* proposed an approach to predicting the probability of change and updating traffic lanes with crowdsourced data and training datasets [2]. Although the proposed scheme in [2] showed excellent performances, a direct application to our problem is difficult. This is because the approach in [2] requires an accurate and up-to-date standard definition (SD) map that represents basic road structures. Therefore, frequent SD map updates are required. However, the proposed scheme can be successfully used even when prior map information is not available. Jo *et al.* presented an algorithm that enables self-driving cars with on-board sensors to perform localization and HD map update simultaneously [3]. It is difficult to directly apply the scheme in [3] to map updating by crowdsourced data because high-cost sensors including light detection and ranging (LiDAR) sensors are required.

Various methods have been proposed for clustering individual traffic lanes for precise map updates. Pannen *et al.* proposed a useful observation clustering algorithm that was named density-based clustering of applications with noise (DBSCAN) in [4]. In [5], an extension with the same lane marking information was proposed. Gupta *et al.* demonstrated a lane extraction algorithm that applies K-means clustering [6] to camera images at the driver's point of view [7]. Lee *et al.* stated a partition-and-group framework that produces line segments with similar geometric characteristics into a cluster [8].

In general, pre-processing of crowdsourced data is required. For example, visual feature extraction is carried out from acquired camera images. As a result, road features are differently encoded in different research groups.

Korean IT company SK Telecom Co., Ltd. has collected crowdsourced data which is called the Road Observation Data (ROD) by installing devices on public buses in Seoul. In this paper, map update is carried out by utilizing ROD. Novel map update schemes are proposed for various landmarks such as traffic lights, traffic signs, and lanes. The key challenge is to overcome various uncertainties of the crowdsourced data. In addition, a strategy for selecting update modes is proposed because map update strategies can vary according to diverse map usages and data characteristics. The verification for the proposed map update scheme is shown by using the data that was collected from the real road environment.

This work was supported in part by the SK Telecom Co., Ltd., the Agriculture, Food and Rural Affairs Research Center Support Program (714002-07) by MAFRA, the National Research Foundation of Korea (NRF) grant funded by the Korea government(MSIT) (NRF-2021R1A2C2007908), and was also supported by the Industry Core Technology Development Project (20005062) by MOTIE.

Kitae Kim, Soohyun Cho, and Woojin Chung are with the Department of Mechanical Engineering, Korea University, 145, Anam-ro, Seongbuk-gu, Seoul, Republic of Korea. smartrobot@korea.ac.kr

## II. OBSERVATION LEARNER

### A. HD Map and Crowdsourced Data

This section describes the characteristics of HD map and crowdsourced data used in this paper. Typically, an initial HD map with high accuracy is produced using expensive MMS vehicles equipped with LiDARs, GNSS sensors, inertial measurement units (IMUs), and wheel encoders. The HD map possesses landmark information including its types and accurate global positions. Landmark types in the HD map include traffic lights, traffic signs, and lanes. In this paper, we utilized an open-source HD map produced by the National Geographic Information Institute, Korea [9].

On the other hand, crowdsourcing devices typically consist of a low-cost camera and a GNSS sensor. Crowdsourced data is generated through a series of on-board processing at the crowdsourcing devices. Analysis of the latest camera image is carried out by the application of deep learning softwares. Then, the landmark types are classified. Finally, the observations are stored in customized formats as described below.

Crowdsourced data contains global positions and covariance matrices of a crowdsourcing device which is defined as a keyframe. Traffic lights and signs are defined as discrete landmarks. A discrete landmark is shown as a single point with the relative position to the keyframe and the covariance matrix. Each discrete landmark has no additional distinction other than its position. A continuous landmark that implies a traffic lane is expressed as a pointset. Every point in a pointset includes the relative position to the keyframe and the covariance matrix. To sum up, the landmark information that is accumulated in crowdsourced data has different geometric properties by type.

Each of our crowdsourced data stores only one information with the lowest observation uncertainty per landmark, even if the landmark is observed many times in several adjacent keyframes. Studies on map updates considering the unique expression of crowdsourced data are rare and some concurrent works are actively carried out including [2].

### B. Overview of Map Update

It is advantageous to adopt a topological map structure as explained in [2]. Since we are focusing on urban roads, the maps are composed of road segments and intersections. For example, each road segment or intersection can be one node of a global topological map.

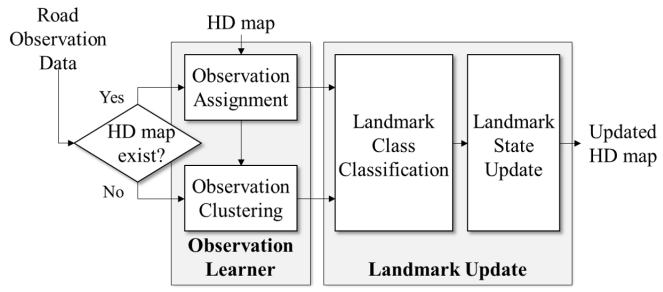



Fig. 1: Overview of the HD map update

An overview of the proposed map update scheme is shown in Fig. 1. The first step is to load the local ROD of our interest. One local map corresponds to one node in a global topological map that is the combination of local maps. The local map in this paper is independent of the SD map. Depending on the availability of the local HD map, the next process of the map update becomes different. When the local HD map exists, observations from ROD are assigned to the HD map. If there is no local HD map, this process is skipped.

In the next step, observations that are not assigned to the HD map are clustered. According to the results of the observation assignment and clustering, landmark classes are classified to detect any changes in the presence of landmarks. Once the landmark class classification is completed, the states of landmarks are updated. This step implies the computation of landmark poses and uncertainties under the consideration of the clustered crowdsourced data. Finally, the HD map is updated.

### C. Observation Assignment

In the observation assignment step, observations with the highest correspondence to the HD map landmarks are assigned to the HD map. In order to effectively reflect the geometric characteristics of each landmark type, different cost functions were designed.

*1) Discrete landmark:*

$$\begin{aligned} d_{1,ij} &= \Delta t_{ij}{}^T \Sigma_{ij}^{-1} \Delta t_{ij} \times \mathtt{tr}(D_{ij}) \\ \Sigma_{ij} &= \Sigma_i + \Sigma_j = V_{ij} D_{ij} V_{ij}{}^T \end{aligned} \tag{1}$$

*i*: reference landmark index
*j*: comparison landmark index

We designed cost function for discrete landmarks as $d_{1,ij}$ by using positions and covariance matrices of two landmarks *i*, *j*. Two landmarks *i* and *j* correspond to a reference and comparison landmark in equation (1), respectively. $\Delta t_{ij}$ is Euclidean distance between two landmarks. Covariance matrices $\Sigma_i$ and $\Sigma_j$ indicate the positional uncertainty of the crowdsourcing device and its observation uncertainty, respectively. $d_{1,ij}$ uses Mahalanobis distance $\Delta t_{ij}{}^T \Sigma_{ij}^{-1} \Delta t_{ij}$ between landmarks and $\mathtt{tr}(D_{ij})$ which is trace of diagonal matrix $D_{ij}$ that includes singular values of $\Sigma_{ij}$. $\Delta t_{ij}{}^T \Sigma_{ij}^{-1} \Delta t_{ij}$ takes probabilistic distribution into account. $\mathtt{tr}(D_{ij})$ represents the stochastic distribution with observation uncertainty which is proportional to the covariance of observations. For discrete landmark *j* in the comparison group, $d_{1,ij}$ is computed. If minimum cost $\min(d_{1,ij})$ is less than threshold $d_{1,thr}$, landmark *j* is assigned to landmark *i*. In the assignment step, landmarks in the HD map and observations from ROD correspond to the reference and comparison landmarks in equation (1), respectively.

$d_{1,ij}$ allows observations to be assigned to the reference, with low measurement uncertainties relative to Mahalanobis distance. That is, with the proposed cost function, not only can we maximize the merits of crowdsourced data that can be collected in large quantities, but also map updates that accurately reflect the real road environment are possible.

*2) Continuous landmark:* Lane pointsets in ROD include high-uncertainty observations. When a lane is located far from keyframes, uncertainty increases. In our crowdsourced data, the number of points in a lane decreases when the uncertainty of an observation is high. Therefore, lane pointsets are filtered out if the number of points in a lane is less than a threshold. The lane filtering is available because the amount of lanes in crowdsourced data is huge enough compared to discrete landmarks.

In the road, there are various shapes of lanes such as straight, curved, and double lanes. Every lane type shows different geometric characteristics. In order to update a variety of lanes precisely, diverse information about crowdsourced lane data needs to be considered. As a means of finding the correspondences between landmarks, there are a lot of existing approaches. The simplest approach is to interprete the reference as a polyline and calculate the orthogonal distance between data and the polyline. This approach has fundamental limitations because it is difficult to evaluate the correspondence to the longitudinal direction of lanes.

Grid-based structure with cartesian coordinates or the K-dimensional tree in [10] is also widely used. However, these schemes are inefficient to cover our problem due to the huge amount of sparse crowdsourced lane data. Conventional scan matching approaches, including generalized-ICP in [11], can be considered. However, scan matching methods result in unsatisfactory performance owing to the low quality of crowdsourced data.

Thus, in this paper, we propose a novel lane learner algorithm that takes into account the geometric characteristics of all crowdsourced lane data. The proposed approach is based on the shell structure that was proposed by Pannen *et al.* as a method to fit graphs in the shell graph fitting algorithm [5]. We modified the proposed structure as a means of assigning and clustering the whole types of lanes. The proposed approach processes in three steps as follows:

1) Shell construction by a reference pointset
2) Projection of a comparison pointset on the shell
3) Cost computation

As the graph fitting algorithm in [5], a centroid of reference pointset $P_{ref}$ is calculated to construct the shell structure that is composed of concentric circles with constant gap $R$. Our proposed shell structure $S$ is created by additionally dividing the shell structure into quadrants. By further segmenting the shell with quadrants, it becomes effective to determine the similarity between longitudinal directions of two lanes. $S$ is shown as black lines in Fig. 2. Each sector that constitutes $S$ is defined as shell element $s_k \in S$. Lanes can be assigned or clustered by computing costs with $S$, based on the similarity of the entire shape between lanes as well as the distance from the centroid of $P_{ref}$.

As a next step, cost $d_2$ is computed for every comparison pointset $P_{com}$ that is projected on $S$. The computation process for $d_2$ is shown in Algorithm 1. The cost function is designed as line 12 of Algorithm 1, which uses average Euclidean distance between points $a_i$ and $b_j$. $n_{ab}$ represents the number of matched points of $P_{com}$ located in every

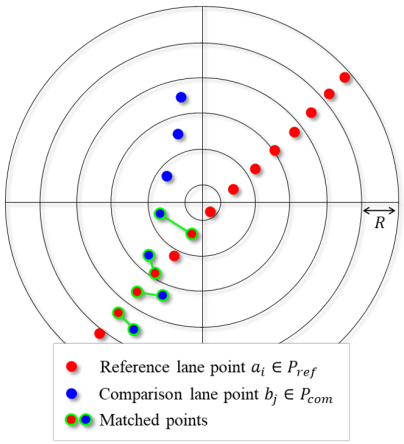



Fig. 2: Illustration of shell structure by continuous landmark: Every sector that is surrounded by black lines is a shell element $s_k \in S$.

**Algorithm 1:** Cost computation for continuous landmark assignment and clustering

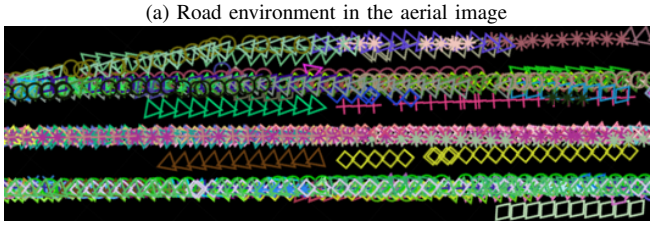
**input :** Reference pointset $P_{ref} = \{a_i\}$,
Comparison pointset $P_{com} = \{b_j\}$
**output:** Cost $d_2$

1 **Initialization:** $n_{ab} = 0$, $n_b = 0$, $dist = 0$
2 Construct a shell structure $S = \{s_k\}$ based on $P_{ref}$
3 **foreach** $b_j \in P_{com}$ *in* $s_k$ **do**
4  **if** *any* $a_i \in P_{ref}$ *exists in* $s_k$ **then**
5  $dist \mathrel{+}= \min(|a_i - b_j|)$
6  $n_{ab} \mathrel{+}= 1$
7  **else**
8  $n_b \mathrel{+}= 1$
9  **end**
10 **end**
11 **if** $n_{ab} \neq 0$ **then**
12  $d_2 \leftarrow \frac{dist}{n_{ab}} \times \sqrt{\frac{n_{ab} + n_b}{n_{ab}}}$
13 **else**
14  $d_2 \leftarrow \infty$
15 **end**

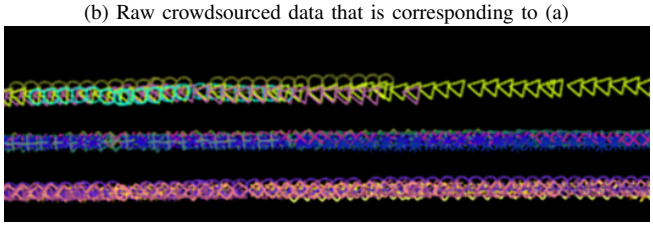
$s_k$. In some cases, $n_{ab}$ is significantly small even though the value of $\frac{dist}{n_{ab}}$ is small. This result is caused by the low quality of the crowdsourced data. In order to avoid the assignment of the crowdsourced lanes to the HD map in this situation, $\sqrt{\frac{n_{ab}+n_b}{n_{ab}}}$ was multiplied as shown in line 12. When $d_2$ is less than threshold $d_{2,thr}$, it is assumed that $P_{ref}$ and $P_{com}$ belong to the same continuous landmark. In the assignment stage, the HD map lanes and observations from ROD correspond to the reference and comparison lanes in $d_2$, respectively.

One of the advantages of the proposed scheme is that high-uncertainty observations are filtered out. Since a large number of crowdsourced data are available in most cases, exploitation of high-quality data is essential in practical

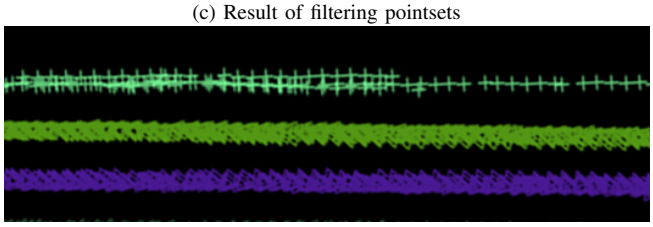
(a) Road environment in the aerial image



(b) Raw crowdsourced data that is corresponding to (a)

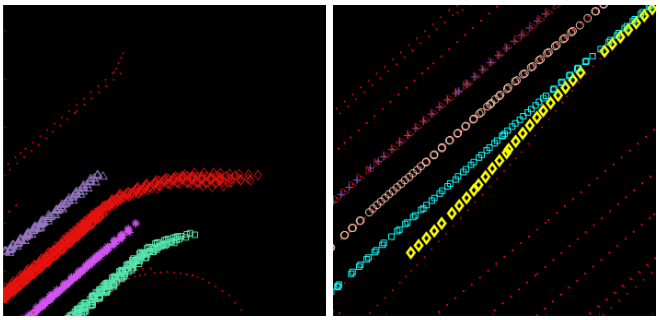


(c) Result of filtering pointsets



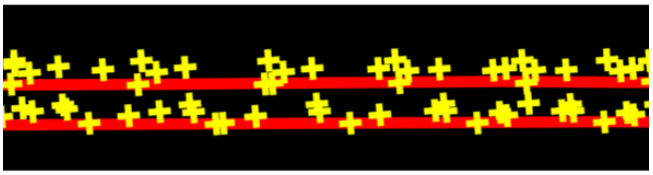
(d) Result of assignment by the proposed method

Fig. 3: Continuous landmark assignment process: Each marker with the same color indicates an individual pointset.

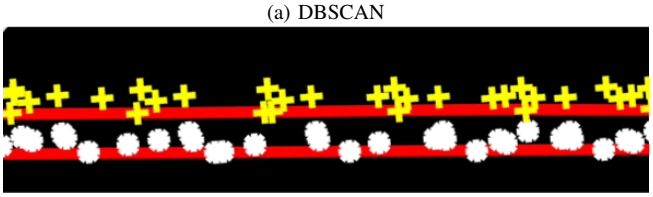


(a) Curved lane (b) Splitted lane

Fig. 4: Assignment results for various lane types. Each assigned pointset is shown with the same colored markers. Red dots indicate ground truth lanes.



(a) DBSCAN



(b) The proposed method

Fig. 5: Visualization of assignment results between DBSCAN and the proposed method. Red lines indicate two ground truths of neighboring parallel lanes with an interval of 0.2 *m*. Other markers are the results of the observation assignment to HD map lanes. Each marker belongs to a distinct pointset.

applications. Moreover, $d_2$ takes account of not only the average Euclidean distance but also the similarity in appearances of $P_{ref}$ and $P_{com}$. This feature implies that assignments can be successfully carried out for various lane shapes.

Fig. 3 shows the result of the entire process of lane assignment by using a number of collected data in the real road. ROD was collected 26 times in street environments in Korea. Every raw data that is collected by crowdsourcing devices presents the same lane differently, as shown in Fig. 3b. As the result of lane filtering, the lanes with high uncertainty were deleted in Fig. 3c. When the proposed algorithm is applied in Fig. 3d, the pointsets that belong to the same ground truth lanes were assigned to the same HD map lane. The pointsets for different lanes were clearly separated.

Fig. 4 shows the assignment results for various shapes of lanes. Even though each lane possesses different geometric characteristics, low-quality pointsets are clearly assigned with corresponding lanes, not only for straight lanes but also for curved and split lanes.

Fig. 5 and TABLE I provide qualitative and quantitative comparisons of the proposed method with DBSCAN in [4] that is widely used for lane point cloud clustering. The result from clustering all reference pointsets and crowdsourced data

TABLE I

QUANTITATIVE COMPARISON RESULTS OF TWO METHODS: DBSCAN AND THE PROPOSED METHOD. THE PERFORMANCES ARE EVALUATED BY COMPARING THE ASSIGNMENT RESULTS WITH THE GROUND TRUTHS.

| | DBSCAN | Proposed |
|---|---|---|
| Accuracy | 88.98% | 99.83% |
| Precision | 96.80% | 99.77% |
| Recall | 93.97% | 99.93% |
| Specificity | 99.60% | 99.99% |
| F1-score | 98.08% | 99.84% |

pointsets in the region of interest at once with DBSCAN is compared with the result from the proposed method. Fig. 5 shows the lane assignment results when two parallel lanes are closely located. From Fig. 5a, it is clear that DBSCAN could not precisely identify two parallel lanes that are separated by 0.2 *m*. As a result, only one thick lane was extracted. On the other hand, Fig. 5b clearly shows that the proposed algorithm accurately assigned crowdsourced data into two pointsets for two neighboring parallel lanes. TABLE I clearly indicates the superior performances of the proposed method quantitatively for the results in Fig. 5.

DBSCAN utilizes points while the proposed scheme deals with pointsets. Therefore, the proposed scheme possibly shows better performances owing to the utilization of the

**Algorithm 2:** Clustering procedure to create candidates to become new landmarks

**input :** Unassigned observations $P_{\mathrm{un}} = \{P_{\mathrm{un},j}\}$
**output:** New landmark candidates $P_{\mathrm{new}} = \{P_{\mathrm{new},i}\}$

1 Choose an observation $P_{\mathrm{un},r} \in P_{\mathrm{un}}$ arbitrarily
2 **Initialization:**
3 Initial new landmark candidate: $P_{\mathrm{new},1} = P_{\mathrm{un},r}$,
4 The number of new landmark candidates: $N_{\mathrm{new}} = 1$,
5 The number of observations in $P_{\mathrm{new},1}$: $n_{\mathrm{new},1} = 1$
6 **foreach** $P_{\mathrm{un},j} \in P_{\mathrm{un}}$, $j \neq r$ **do**
7  **for** $i = 1$ to $N_{\mathrm{new}}$ **do**
8  $d_{ij} \leftarrow d(P_{\mathrm{new},i},\ P_{\mathrm{un},j})$
9  **end**
10  **if** $\min(d_{ij}) < d_{thr}$, $i \in \{1, \ldots, N_{\mathrm{new}}\}$ **then**
11  $k \leftarrow \mathrm{argmin}_i(d_{ij})$
12  $P_{\mathrm{new},k} = P_{\mathrm{new},k} \cup P_{\mathrm{un},j}$
13  $n_{\mathrm{new},k}$ += 1
14  **else**
15  $N_{\mathrm{new}}$ += 1
16  $P_{\mathrm{new},N_{\mathrm{new}}} = P_{\mathrm{un},j}$
17  $n_{\mathrm{new},N_{\mathrm{new}}} = 1$
18  **end**
19 **end**

features of continuous landmarks in the HD map and crowdsourced data.

### D. Observation Clustering

The observation clustering step deals with observations that are not assigned to the HD map landmarks in Section II-C. If there is no available HD map information, all crowdsourced data becomes unassigned observations. When the HD map information is available, unassigned observations are likely to be the newly created landmarks that was not included in the HD map. Therefore, appropriate map update of adding new landmarks is required when unassigned landmarks appear repeatedly.

The clustering method is identical to the proposed schemes in Section II-C. To cluster observations by the proposed cost functions in Section II-C, a reference and comparison observation must be chosen among the unassigned landmarks. The way to choose the reference and comparison landmarks for clustering is carried out in the same manner as in Algorithm 2. In line 8 of Algorithm 2, $P_{\mathrm{new},i} \in P_{\mathrm{new}}$ and $P_{\mathrm{un},j} \in P_{\mathrm{un}}$ play a role of a reference and comparison landmark for observation clustering, respectively.

## III. LANDMARK UPDATE

### A. Landmark Class Classification

In order to describe the deletion or addition of individual landmarks, four landmark classes are defined as shown in TABLE II. *Normal* and *Deleted* are the landmark classes for HD maps, while *New* and *Outlier* are for crowdsourced data.

TABLE II
DEFINITION OF LANDMARK CLASSES

| | Landmark existence | |
|---|---|---|
| Class | Before Update | After Update |
| *Normal* | existent | existent |
| *Deleted* | existent | non-existent |
| *New* | non-existent | existent |
| *Outlier* | non-existent | non-existent |

If the quality of crowdsourced data is high, the map can be updated accurately with a little observation data. However, a large number of low-quality observations can be found in the crowdsourced data. A significant number of landmarks were not detected due to occlusion by other vehicles. Sometimes the covariance of landmark poses becomes large due to the limitation of visual image processing. Low-quality observations cannot be assigned or clustered successfully by the proposed scheme in many cases. Therefore, class classification should be carefully carried out under the consideration of the quality of acquired data.

$$\frac{N_{\mathrm{assigned},i}}{N} > thr_{\mathrm{assigned}} \tag{2}$$

*i*: reference landmark index

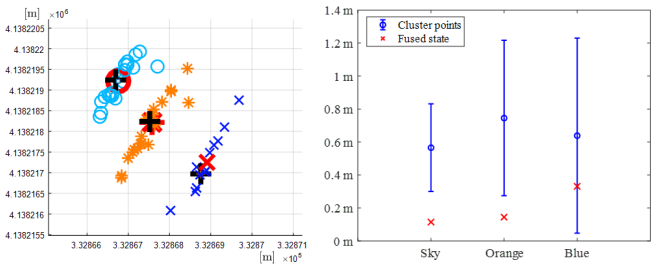
The criteria for class classification are different according to the landmark types. Suppose that the number of crowdsourced data utilized for map update is *N*. For each HD map landmark, *Normal* and *Deleted* are determined by *N* and the number of successful assignments $N_{\mathrm{assigned},i}$ in Section II-C. If the fraction between $N_{\mathrm{assigned},i}$ and $N$ is larger than threshold $thr_{\mathrm{assigned}}$ as shown in equation (2), HD map landmark with index *i* is classified as *Normal* which remains on the HD map. Otherwise, HD map landmark with index *i* is classified as *Deleted*.

$$\frac{n_{\mathrm{new},i}}{N} > thr_{\mathrm{new}}, i \in \{1, \ldots, N_{\mathrm{new}}\} \tag{3}$$

*New* and *Outlier*, which are related to crowdsourced data, are classified by monitoring the ratio between *N* and the number of landmarks in each cluster $n_{\mathrm{new},i}$ that is created in Section II-D. If the ratio between $n_{\mathrm{new},i}$ and $N$ is larger than threshold $thr_{\mathrm{new}}$ as shown in equation (3), the $i_{\mathrm{th}}$ cluster is classified as *New*. Otherwise, the $i_{\mathrm{th}}$ cluster is classified as *Outlier*.

### B. Landmark State Update

As the result of observation clustering in Section II-D, a cluster for discrete landmarks is represented as a set of points. In the case of a cluster for continuous landmarks, it appears as a set of pointsets. Every element in a cluster contains its 3D position and covariance matrix information. HD maps only store point and pointset information of discrete and continuous landmarks, respectively. For map updates, *New* landmarks should be inserted as a point or a pointset to the HD map.




(a) Clustering result (b) Position errors for each cluster

Fig. 6: State fusion results for *New* discrete landmarks: (a) Clusters and ground truths are illustrated as markers and black crosses, respectively. Each red marker is a fused state for each cluster. (b) Red crosses indicate position errors of fused states with respect to the ground truths. Blue error bars indicate position errors before fusion.

*1) Discrete landmark:* In order to add a set of multiple points as one point with the 3D position and the covariance matrix, Maximum A Posteriori (MAP), which is based on Bayes' theorem, was used [12].

$$\mathbf{x}_{MAP} = \Sigma_{MAP} \sum_{k=1}^{n} {\Sigma_k}^{-1} \mathbf{x}_k$$
$$\Sigma_{MAP} = \left[ \sum_{k=1}^{n} {\Sigma_k}^{-1} \right]^{-1} \tag{4}$$

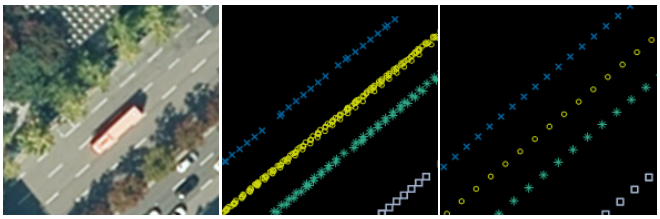
MAP in equation (4) converts a set of points into one 3D position $\mathbf{x}_{MAP}$ and one covariance matrix $\Sigma_{MAP}$. The fused state is the state that most appropriately represents multiple points as a single point.

Fig. 6 shows the result of observation clustering and position errors between fused states and ground truths. Fig. 6a shows that unassigned discrete landmarks are clustered appropriately. It is clear that the observation uncertainty is high to the driving direction. However, the position errors decrease significantly by fusing states.

*2) Continuous landmark:* In order to add a set of clustered continuous landmarks to an updated map, a traffic lane has to be represented by a smooth curve. Representative points for each lane set are computed, and a smooth dotted line is generated by using the representative points. A part of the graph fitting algorithm that was proposed by Pannen *et al.* is used to compute the representative points for each set [5].

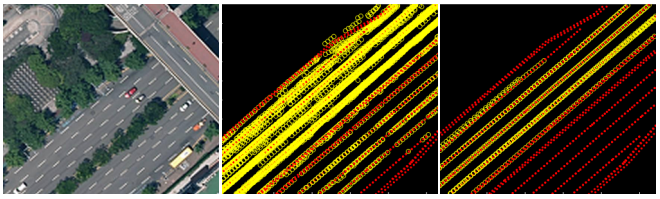
In order to generate smooth curves, there are some possible alternatives including Bézier curve [13], B-spline curve [14] and spline-interpolation [15]. Bézier curve is advantageous to fit smooth curve than spline-interpolation algorithm that fits curves by connecting all control points. Even though there is no obvious difference in the appearance of the fitted line between B-spline curve and Bézier curve algorithm, Bézier curve algorithm is computationally more efficient. Therefore, we exploited Bézier curve fitting algorithm to express lanes.

Fig. 7b shows that clustered sets are represented by blurred lines due to positional errors. Through Bézier curve fitting algorithm, a clustered set is fitted into a smooth dotted line that is similar to an HD map lane. The result of lane fitting is presented in Fig. 7c.



(a) Road environment (b) Clustering (c) Lane fitting

Fig. 7: Result of lane fitting for *New* continuous landmarks: Different markers with different colors stand for individual re-clustered lanes in (b) & (c).



(a) Road environment (b) Raw data (c) Updated map

Fig. 8: Lane data comparison between before and after the update without HD map: Yellow dotted lines show the pointsets before and after updating maps. Red dotted lines represent the ground truth lanes.

TABLE III
QUANTITATIVE RESULTS FOR *New* LANDMARKS

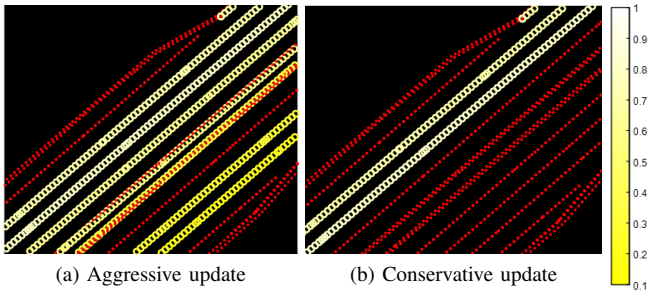
| Metric | Discrete | Continuous |
|---|---|---|
| Number of *New* landmarks | 23 | 3,957 (points) |
| Mean of position error | 0.31 *m* | 0.19 *m* |
| Std. Dev. of position error | 0.16 *m* | 0.17 *m* |

Fig. 8 presents the result of the map update by only using a large amount of crowdsourced data without HD map in a road environment, shown in Fig. 8a. Utilizing only raw data with various position errors in Fig. 8b, *New* lanes are created as in Fig. 8c. From Fig. 8c, it is evident that newly learned lanes precisely match the ground truth lanes.

TABLE III summarizes position errors between ground truths and *New* landmarks generated by state fusion and lane fitting in nine arbitrary areas. Discrete landmarks showed mean errors of approximately 0.31 *m* with a standard deviation of 0.16 *m*. In the case of *New* lanes, the error is defined as the pointset-to-pointset Euclidean distance between *New* lanes and ground truths. This fact implies that longitudinal errors to the driving direction are included. Therefore, lateral errors are less than mean error of 0.19 *m*.

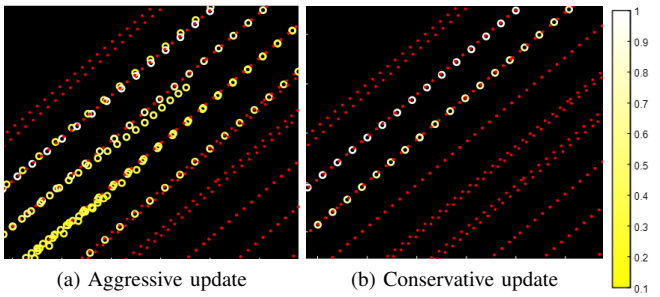
### C. Map Update Mode

Maps are used for various purposes such as environmental representation, localization, and path planning. The strategy for map updates should be carefully designed under the consideration of target applications, data qualities, data size, available sensors, environmental characteristics, and so forth. For example, in order to achieve accurate localization performance, deletion criteria play a significant role. Localization performance decreases when unknown landmarks are repeatedly detected. However, missing landmarks is not a serious



(a) Aggressive update (b) Conservative update

Fig. 9: Two different update results for *Normal*. Depending on *Normal* update mode, *Normal*s (yellow circles) and *Deleted*s (red dots) are diversely defined. The color bar shows the observation frequency. If the color is close to white, the frequency reaches 1.



(a) Aggressive update (b) Conservative update

Fig. 10: Two different update results for *New* (yellow circles). Depending on *New* update mode, *New*s are generated differently. Ground truth lanes (red dots) are plotted to compare with *New*s.

problem because occlusions frequently take place. Therefore, deletions of non-existent landmarks should be aggressively carried out, while additions of new landmarks are rather conservative.

"Aggressiveness" of map updates is determined by two parameters as explained in Section III-A. When corresponding HD map landmarks are found, the degree of change is adjusted by equation (2). When the addition of newly observed landmarks is considered, the aggressiveness is determined by equation (3). By adjusting the two parameters, flexible and systematic map updates are possible.

Fig. 9 describes two updated maps determined by the parameters being applied to the classes related to the HD map. If a user wants to include landmarks as many as possible on the map regardless of precision, relatively large number of landmarks can be registered on the updated map, as illustrated in Fig. 9a. If the HD map is deemed unreliable, only frequently observed HD map landmarks that are described by yellow circles in Fig. 9b are added to the updated map.

Fig. 10 shows two updated maps with different parameter values for the crowdsourced landmark classes. In two maps, *New* landmarks are generated without an HD map. In Fig. 10a, *New* landmarks with relatively large position errors were also added into the updated map by tuning parameters aggressively. In contrast, if a user wants to obtain a highly precise map even though the number of updated landmarks is small, only frequently observed landmarks can be newly updated to the map, as shown in Fig. 10b.

## IV. CONCLUSION

In this paper, an HD map update algorithm is proposed, taking into account the geometric characteristics of landmarks in a number of crowdsourced data. The proposed approach could utilize only reliable information for the map update by using the uncertainty information. With the novel lane observation learner method which uses shell structures, various kinds of crowdsourced lanes were assigned with HD maps and clustered effectively.

In conclusion, the proposed method proved its usefulness of the map update by using low-quality crowdsourced data collected in real road environments. By defining various update modes, the updated map could be expressed diversely, depending on the quality of crowdsourced data and the purpose of HD map usage.

## REFERENCES


[1] G. Petrie, "An introduction to the technology: mobile mapping systems," Geoinformatics, vol. 13, no. 1, p. 32–43, 2020.

[2] D. Pannen, M. Liebner, W. Hempel, and W. Burgard, "How to keep HD maps for automated driving up to date," in 2020 IEEE/RSJ International Conference on Robotics and Automation (ICRA). IEEE, 2020, pp. 2288–2294.

[3] K. Jo, C. Kim, and M. Sunwoo, "Simultaneous localization and map change update for the high definition map-based autonomous driving car," Sensors, vol. 18, no. 9, p. 3145, 2018.

[4] M. Ester, H. Kriegel, J. Sander, and X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise," Kdd., vol. 96, no. 34, p. 226–231, 1996.

[5] D. Pannen, M. Liebner, and W. Burgard, "Lane marking learning based on crowd-sourced data," in 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE, 2019. pp. 7040–7046.

[6] S. Lloyd, "Least squares quantization in PCM," IEEE transactions on information theory vol. 28, no. 2 pp. 129–137, 1982.

[7] A. Gupta, and S. N. Merchant. "Automated lane detection by k-means clustering: a machine learning approach," Electronic Imaging vol. 2016, no. 14, pp. 1–6, 2016.

[8] J. Lee, J. Han, and K. Whang, "Trajectory clustering: a partition-and-group framework," Proceedings of the 2007 ACM SIGMOD international conference on Management of data.

[9] *HD map dataset of South Korea:* Concatenated, National Geographic Information Institute in the Republic of Korea, February 2021. [Online]. Available: https://www.data.go.kr/en/data/15059912/fileData.do

[10] J. L. Bentley, "Multidimensional binary search trees used for associative searching," Communications of the ACM 18.9, pp. 509–517, 1975.

[11] A. Segal, D. Haehnel, and S. Thrun, "Generalized-icp," Robotics: science and systems, vol. 2, no. 4, p. 435, 2009.

[12] M. Kumar, D. P. Garg, and R. A. Zachery, "A generalized approach for inconsistency detection in data fusion from multiple sensors," 2006 American Control Conference. IEEE, 2006.

[13] G. Teofilo, J. Diaz-Herrera, A. Tucker, Computing Handbook, Third Edition: Computer Science and Software Engineering, CRC Press, pp. 14–32. ISBN 978-1-4398-9852-9.

[14] W. J. Gordon, and R. F. Riesenfeld, "B-spline curves and surfaces," Computer aided geometric design. Academic Press, 1974, pp. 95–126.

[15] S. McKinley, and M. Levine, "Cubic spline interpolation," College of the Redwoods vol. 45, no. 1, pp. 1049–1060, 1998.