Integrating V-SLAM and LiDAR-based SLAM for Map Updating

Yu-Cheng Chang
Department of Mechanical and Electro-Mechanical Engineering
National Sun Yat-sen University
Kaohsiung City, Taiwan

Ya-Li Chen
Department of Mechanical and Electro-Mechanical Engineering
National Sun Yat-sen University
Kaohsiung City, Taiwan

Ya-Wen Hsu
Department of Mechanical and Electro-Mechanical Engineering
National Sun Yat-sen University
Kaohsiung City, Taiwan

Jau-Woei Perng
Department of Mechanical and Electro-Mechanical Engineering
National Sun Yat-sen University
Kaohsiung City, Taiwan
jwperng@faculty.nsysu.edu.tw

Jun-Dong Chang*
Smart System Institute
Institute for Information Industry
Taipei City, Taiwan
jundongchang@iii.org.tw

2021 IEEE 4th International Conference on Knowledge Innovation and Invention (ICKII) | 978-1-6654-2307-6/21/$31.00 ©2021 IEEE | DOI: 10.1109/ICKII51822.2021.9574718

***Abstract*—Vehicle positioning generally uses the global navigation satellite system (GNSS), but systems of different levels significantly affect positioning accuracy. Moreover, it is greatly affected by weather that may cause inaccurate positioning due to excessive cloud cover. In the technology of self-driving cars, how to achieve stable and accurate positioning is a critical topic. This research first uses RTK-GPS, LiDAR, and cameras to build high-precision map information and realize vehicle positioning functions. By using the normal distribution transform (NDT), ORB-SLAM, and iterative closest point (ICP) were matched to the feature points of the current map with the pre-built map to complete the vehicle positioning system. If the change of static objects in the map was detected during the positioning process, the map was updated. Finally, the performance of the proposed system was verified on the road on the campus.**

Keywords—Visual SLAM, LiDAR SLAM, HD Map, Point Cloud Map Updating

I. INTRODUCTION

In the research field of mobile robots and autonomous driving, map-based outdoor navigation is one of the critical technologies. The robot or vehicle can complete its positioning by matching a series of sensor signals with a pre-prepared accurate map.

Most of the previous vehicle positioning systems used GPS as a reference signal, and the error was larger than 15 meters. There is also a high-precision and-priced positioning system, RTK-GPS, whose positioning accuracy reaches 2 cm. This indeed completes the precise positioning function of the vehicle. Even though RTK has s high positioning accuracy, the signal is based on GPS. If the weather is bad or the signal is blocked, the error is no different from ordinary GPS.

In Ref. [1], the author installed the Velodyne HDL-64E S2 laser scanner on the top of the vehicle to scan the 360° environment. This view, combined with high data rates, has made it popular among smart vehicle research and development groups. Although there are methods to recalibrate [2,3], the main disadvantage is that the measurement noise is relatively high. The method introduced in Ref. [2] introduces a special treatment of this noise in the context of SLAM. When building the map, the measurement is made in a flat place. This provides positioning accuracy and allows the constructed map for a 3D model in the future.

In terms of vehicle positioning, Ref. [4] uses iterative closest point (ICP) to match two sets of 3D point clouds and calculates the translation matrix and rotation matrix between the two sets of point clouds. Although ICP accurately matches in most cases, it is appropriate for offline point cloud matching as the algorithm calculates each point in the set, and the process takes much time. In Ref. [5], the normal distributions transform (NDT) method was proposed for point cloud matching. NDT divides the space into multiple grids and uses a normal distribution that models the probability of data points in each grid. This method does not need to calculate every point, so the processing speed is much faster. However, the setting of the grid size still affects the matching speed and accuracy.

In Ref. [6], the Velodyne HDL-32E LiDAR sensor was used in the M-City test experiment of the University of Michigan to obtain a large amount of point cloud information. Through ground filtering, point cloud clustering, and principal component analysis, the features in the 3D point cloud effectively propose vehicle positioning. The author proposes a fast and stable method to extract 3D features from sparse point clouds.

The contributions of this research are as follows.

(1) A mechanism for positioning using images and point clouds is proposed. The positioning method of image feature points is used instead of GPS. If the point cloud positioning fails, the image feature points are reused to locate the initial position.

(2) The map update detection and recording method are proposed, which calculates the difference between the map point cloud and the scanned point cloud and updates the area with an excessive difference. The update mechanism improves the usability of the map.

The rest of the study is organized as follows. In section 2, the methodology of the proposed system is described. Section 3 presents the experimental results and evaluates the proposed schemes. Finally, conclusions and discussions are provided in Section 4..

II. PROPOSED METHOD

The proposed system architecture is divided into three parts: 1) building a fusion map of V-SLAM and LiDAR-based SLAM, 2) vehicle localization, and 3) map updating. First, the fusion of LiDAR and image information and 3D mapping are performed in a fixed field. Then the completed map is used to locate the vehicle. The map is matched through the scanned point cloud and image data to find the vehicle's position in the scene and finally locate the vehicle. Simultaneously, the difference between the scanned information and the existing map is detected, and the map is updated when the difference is too large.

A. *Fusion Map of V-SLAM and LiDAR SLAM*

In the map building of point cloud image fusion, the image features in front of the car are extracted with ORB, and the point cloud data from the 3D LiDAR is matched and mapped in the point cloud space with NDT. Finally, the extracted features in the image are combined with the point cloud map.

1) *NDT Mapping*

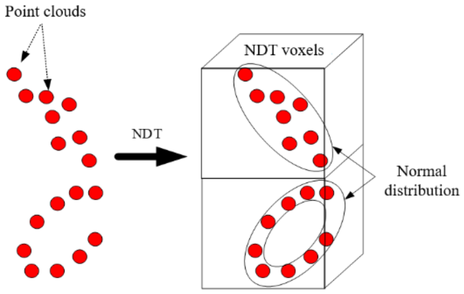


Fig. 1. NDT volume cell segmentation.

NDT is an algorithm for matching two sets of data points. It divides the original data points distributed in 2D or 3D space into cells of the same size and then calculates the distribution of data points in each cell. The initial posture is used to convert the target data points and then matched them, and the Newton method is used to iterate and optimize the objective function. The NDT algorithm is divided into seven steps.

Step 1: Divide the space distributed by the original data points into equal-sized volume cells (Voxel).

Step 2: Allocate the data points in a volume unit, and calculate the mean and variance of the multi-dimensional normal distribution parameters in each voxel.

$$\vec{\mu}_s = \frac{1}{N}\sum_{i=1}^{N}\vec{s}_i \tag{1}$$

$$V_s = \frac{1}{N-1}\sum_{i=1}^{N}(\vec{s}_i - \vec{\mu})(\vec{s}_i - \vec{\mu})^T \tag{2}$$

where $\vec{\mu}_s$ is the mean value of the original data point position $\vec{s}_i$, N is the number of original data points, and V_s is the variance of the original data points, which is the degree of dispersion of the original data point distribution, as shown in Fig. 1.

Step 3: Use the set initial transition to translate each target data point into a volume unit in the original data point space.

$$\begin{pmatrix} x_p{}' \\ y_p{}' \end{pmatrix} = \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix}\begin{pmatrix} x_p \\ y_p \end{pmatrix} + \begin{pmatrix} t_x \\ t_y \end{pmatrix} \tag{3}$$

where x_p and y_p are the coordinate points of the target data point, θ is the angle of the initial transition, t_x and t_y are the translation values of the initial transition, and $x_p{}'$ and $y_p{}'$ are the target data coordinates after the initial transition point.

Step 4: Divide each target data point after conversion into the volume cell of each original data point. To avoid breaking the 3D structure by dividing the space into volume cells, there are overlapping areas in each volume unit, and each target data point in the three-dimensional space is allocated to 8 volume units.

Step 5: Calculate the probability distribution of each converted target data point in the volume cell.

$$\mathrm{P}(\vec{p}') \sim exp\left(-\frac{(\vec{p}'-\vec{\mu}_s)^T V_s{}^{-1}(\vec{p}'-\vec{\mu}_s)}{2}\right) \tag{4}$$

where $\vec{p}'$ is the initially converted target data point, V_s is the variance in the volume cell, $\vec{\mu}_s$ is the mean value in the volume cell, and P is the function of probability distribution.

Step 6: Sum the probability distributions of all target data points.

$$\mathrm{score}(\vec{p}') = \sum_{i=1}^{N_p} exp\left(-\frac{(\vec{p}'-\vec{\mu}_s)^T V_s{}^{-1}(\vec{p}'-\vec{\mu}_s)}{2}\right) \tag{5}$$

where N_p is the number of target data points.

Step 7: Use Newton's method to optimize the objective function score$(\vec{p}')$, find the best transformation parameters to minimize $-$score$(\vec{p}')$, and perform iterative operations until convergence.

2) *ORB SLAM*

ORB-SLAM mainly includes three parts: tracking, mapping, and loop detection. In the overall ORB-SLAM process, the ORB algorithm detects and matches the feature points of the image, and then the bundle adjustment (BA) nonlinear iterative optimization method obtains the camera position and map data.

The OGB algorithm consists of two parts: the key point oriented FAST (oFAST) and the feature descriptor binary robust independent element feature (BRIEF). The key point is the corner point where the pixel value of the small area in the image has noticeable bright and dark changes. Feature descriptors encode features into a series of numbers to distinguish features. When the object is moving, the surrounding environment image continues to transform (rotation, displacement), and the feature descriptor can make the surrounding environment feature to be found again.

The primary purpose of keyframe loop detection is to establish the constraint relationship between the current moment and the last moment and achieve the optimization and map correction method. Since there are some errors between each frame when building a map, the camera position drift is inevitable during the movement of the carrier, such as vehicles and robots. As the running time increases, the larger the amount of data, the greater the risk of drift and error, eventually leading to divergence.

Therefore, it is necessary to use the keyframe closed-loop method and the first image as a keyframe and add it to the keyframe sequence. During the movement, after the preset frame number detection interval is exceeded, the system calculates whether it is a keyframe and adds it to the keyframe sequence. When the camera comes to the previously visited scene again, it obtains a similar feature description and achieves closed-loop detection because it sees the same scene.

3) *Fusion*

Although ORB-SLAM builds a map outline, there is an unknown scale factor in constructing a 3D map from a 2D view. Therefore, visual-based SLAM is prone to scale drift.

We propose using the ICP algorithm to fit the mapping trajectory of NDT and ORB, making the ORB map get scale correction.

B. Localization

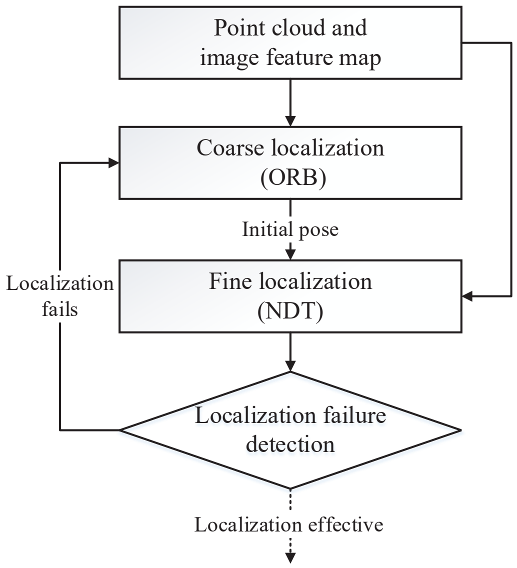


Fig. 2. Vehicle positioning process based on point cloud and image fusion.

If the initial positioning point is not given during NDT positioning, NDT searches the entire map for matching. This process is time-consuming, so the vehicle usually is parked at the initial position set by itself for matching and convergence. However, when the vehicle is moving, if positioning failure or divergence occurs, the vehicle does not reposition itself, so it is usually equipped with another sensor to assist in the initial positioning. At present, GPS is most commonly used to give initial positioning coordinates so that NDT is repositioned in any area of the scene. However, GPS also has its limitations on the field of use, such as indoor places, high-rise buildings around, tunnel sections, or poor weather which seriously affect GPS signal transmission. This research proposes using a camera instead of GPS to match the coordinates of the NDT with the image information to initialize the positioning and reduce the limitation of the positioning by using scenes.

After the fusion map is built, the vehicle positioning based on point cloud and image fusion is performed as shown in Fig. 2. First, the ORB algorithm extracts the image features in front of the vehicle, matches and compares them with the image features in the map, and then obtains the current initial positioning position of the vehicle. The initial position is then provided to the NDT algorithm and the map for matching and positioning. Then, the matching score of the current positioning is calculated. When the matching score is less than the positioning failure threshold, the image feature points is used to initialize the positioning again.

In the vehicle positioning process of point cloud and image fusion, positioning failure is to determine whether the matching score is less than the failure positioning threshold; the positioning failure threshold was set to 1, and the matching score is calculated and compared between the scanned point cloud and the map point cloud, The matching score is calculated as Eq. (5). The matching score is the probability that the scanned point cloud is distributed in the voxel space of the map. The larger the matching score, the more reliable the positioning result, and the smaller the positioning possibility.

C. Map Updating

When the vehicle is driving, the fusion map update performs updating detection and recording according to the location. The system determines whether this location has been updated within a certain period. If it has been updated, it continues to determine the next location. If it has not been updated, it calculates the matching score between the map and the scanned point cloud. If the matching score is less than the update threshold, the location and its scanning point cloud information are recorded. Here, the update threshold is set to 3. The map is updated when the vehicle is in the mission completion or standby phase. First, the location to be updated recorded in the map is removed, and the relative scanning point cloud is filled, and finally, the map information of the location is updated. The update process of the fusion map is shown in Fig. 3.

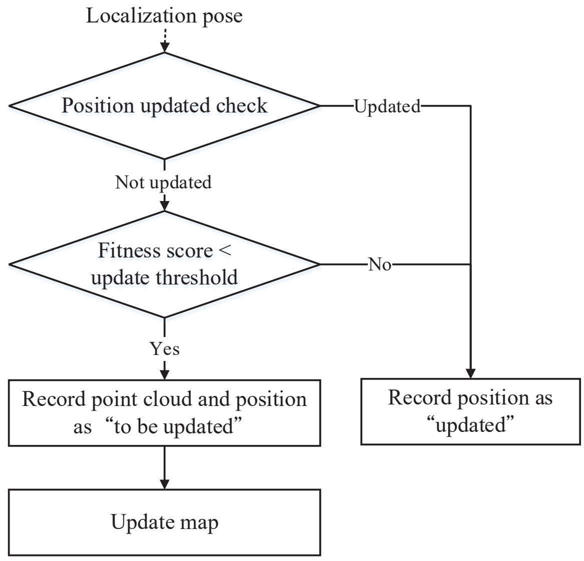


Fig. 3. Updating process of fusion map.

III. Experimental Results

A. Vehicle setting and experimental field

The experimental platform is mainly an autonomous vehicle modified from a five-seater golf cart, as shown in Fig. 4. Two 3D LiDARs (Velodyne Puck) are respectively mounted on the front and rear of the golf cart roof. The camera is installed above the windshield of the vehicle, and its detection range is mainly in front of the vehicle.

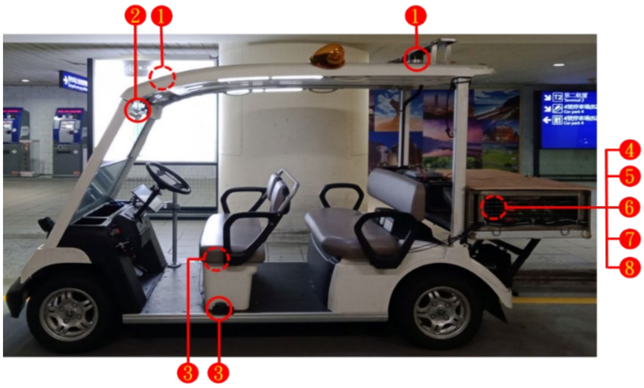


Fig. 4. Golf cart and equipped sensors in the experiment.

The LiDARs equipped on the vehicle scanned the 3D environment around the vehicle, and the pre-built point cloud map of the driving route was used to locate the vehicle. The positioning algorithm matched the currently scanned point cloud around the vehicle with the pre-built point cloud map based on the given initial value. This experiment was conducted around the building of the College of Marine Science on the campus. On a rectangular closed-loop road, the performance of closed-loop detection was tested. Besides, there were roadside parking vehicles in the field to verify the update capability of the map. The satellite image and route of the experimental scene are shown in Fig. 5.

Fig. 5. Vehicle route used for evaluation.

B. *Evaluation of mapping based on sensor fusion*

After scanning and recording the image in front of the vehicle, the ORB-SLAM algorithm built the image map. The feature points of the ORB were extracted from the front image, then matched with the feature points of the image in the continuous picture. Finally, according to the matching result, the continuous image sequence was superimposed into an image feature point map. Figure 6 shows that the black points represent the feature points of the image, the red points the current map location, and the green line is the driving path when recording the front image. Through the front image of the car, a ring road map of the experimental scene was roughly constructed, and the closed-loop detection and connection were completed. However, all 3D coordinates output from ORB-SLAM were not in metric system but prone to proportional distortion.

Fig. 6. Mapping result of ORB-SLAM.

After scanning and recording the point cloud data of the scene, the NDT-SLAM algorithm built the point cloud map. By calculating the point cloud distribution state and parameters of each voxel space, the scanned point cloud was superimposed into a complete point cloud map based on the matching result and the relative position, as shown in Fig. 7. The point cloud map in the figure is presented in a bird's view and shows the closed-loop route of the scene is built entirely.

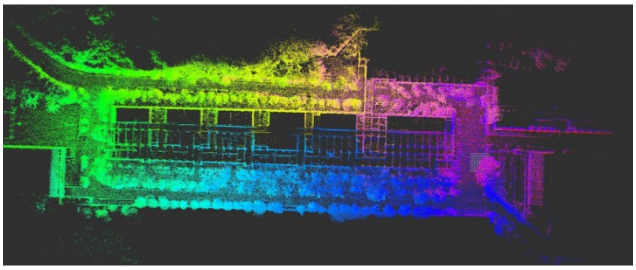


Fig. 7. Result of NDT mapping.

The ICP algorithm matched the routes of ORB-SLAM and NDT-SLAM so that the image feature point map and the point cloud map were superimposed as shown in Fig. 8. The colored lines in the figure are based on NDT, and the white lines are based on ORB-SLAM. After the path is superimposed, the map has both 3D point cloud and image feature information for the subsequent use of LiDAR or camera for vehicle positioning.

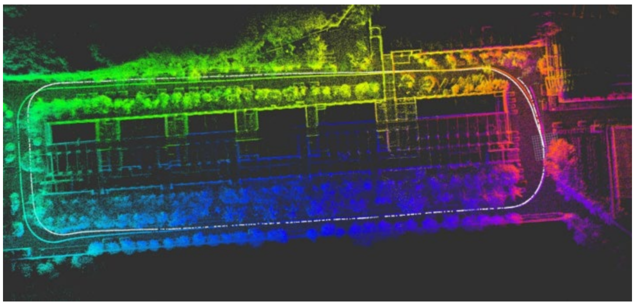


Fig. 8. Fusion map after closed-loop detection.

When constructing NDT and ORB maps, high-precision RTK-GPS positioning system data was collected as the ground truth to calculate the mapping error and accuracy. Figure 9 shows the path of NDT, ORB, and RTK-GPS, where the blue line is the RTK-GPS path, the red line is the path during NDT mapping, and the green line is the path during ORB mapping. In the lower-left corner of the figure, the RTK-GPS path is relatively unstable. The reason is that there is a connecting bridge between two buildings, which makes the RTK-GPS signal more unstable when the vehicle passes here. The ground truth information of this road section is not referenced.

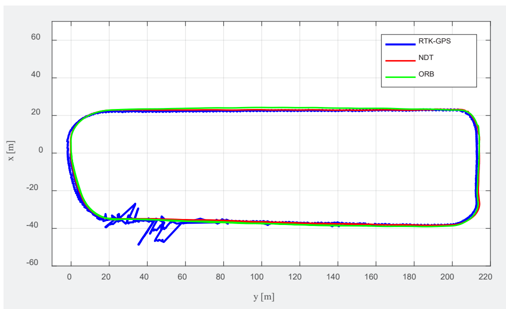


Fig. 9. Trajectories of mapping.

Since the point cloud and image fusion map is built, the ORB image mapping information is superimposed on the point cloud map built by NDT. This experiment included RTK-GPS positioning information as a reference to calculate the NDT mapping information error. As shown in Table 1, the longitudinal and lateral errors are calculated separately in the table, and the average error of the mapping is between 0.4m and 0.7m.

TABLE I. NDT MAP CONSTRUCTION ERROR

Error	Longitudinal error	Lateral error
Average	0.680 m	0.463 m
Max	2.331 m	1.379 m
Standard deviation	0.508 m	0.462 m

C. Evaluation of vehicle localization

This project used the vehicle's front image to extract the features of the ORB, find the key points in the image, perform the feature description, and match the key points. After the key points of the image are successfully matched, they are located in the image feature point map, as shown in Fig. 10. The black points in the figure are the key points in the image feature map, and the red points are the key points for the current successful matching. The blue box is the keyframe in the map, and the green box is the current location and direction of the vehicle on the map.

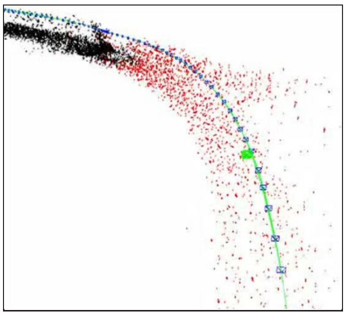


Fig. 10. ORB matching and positioning.

NDT matches the point cloud distribution state of each voxel space of the point cloud map and finally locates the position of the vehicle on the point cloud map, as shown in Fig. 11. The white point cloud in the figure is the point cloud map. The coordinate position in the middle is the positioning position of the vehicle.

After the feature extraction and matching through the image, the vehicle's initial location is obtained, and then the NDT point cloud matching algorithm locates the initial location. Through this algorithm, point cloud and image positioning is performed in the field at the same time. However, as the error of using image feature point positioning is larger than using point cloud positioning, this algorithm is mainly based on point cloud positioning. The image feature points are used for global positioning only when the initial positioning or positioning fails. This method not only enhances the practicability of the positioning algorithm but also makes the positioning more stable.

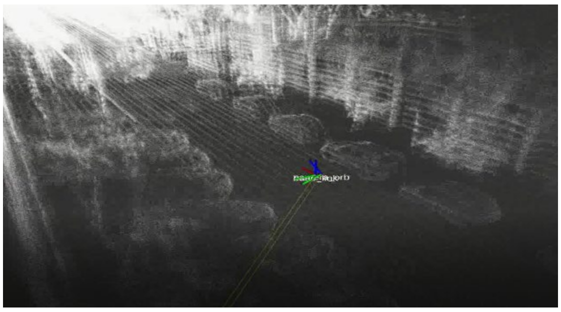


Fig. 11. NDT matching and positioning.

D. Evaluation of Map Updating

This experiment built and updated the map at different times of the same scene by comparing the changes of the scene at different times and recording the location and road section where the scene changes. As shown in Fig. 12, the black line in the figure is the path of the scene. The area marked in red is the area where the matching score of the scanned point cloud. The map point cloud is lower than 3 after the map update detection, that is, the area where the environment changes. This experiment detected a total of 3 areas that need to be updated.

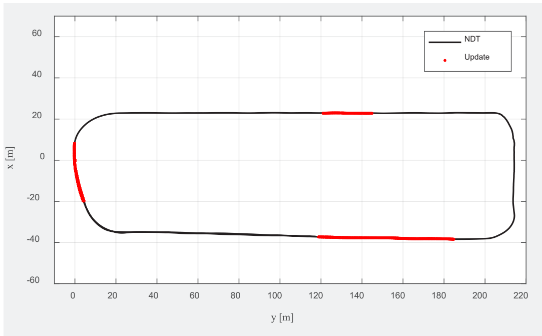


Fig. 12. Trajectories of road segments to be updated.

When the vehicle is in the map update stage, the location to be updated recorded in the map is removed, and the corresponding scanned point cloud is filled. In Fig. 13, the white points are the original map point clouds, and the removed part in the figure is the area where the matching score is less than 3. Figure 14 shows the location to be updated and recorded in the area and its scanning point cloud. The orange point cloud in the figure is the scanning point cloud to be updated.

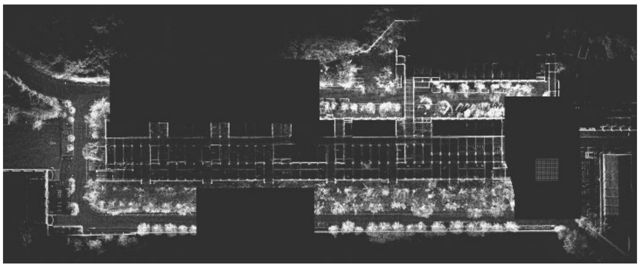


Fig. 13. Remove the area to be updated.

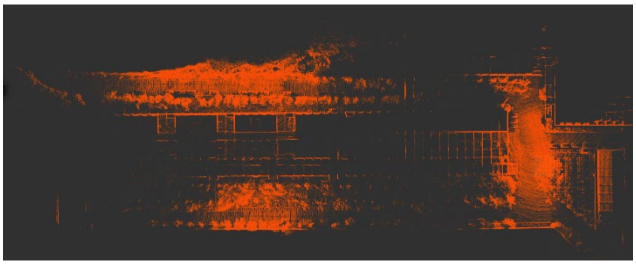


Fig. 14. The area to be updated.

Through the map update mechanism, the vehicle is accurately positioned in the scene. As shown in Fig. 15, there are three update areas in the figure, and the updated point cloud is displayed in orange, and the un-updated point cloud of the original map is displayed in white.

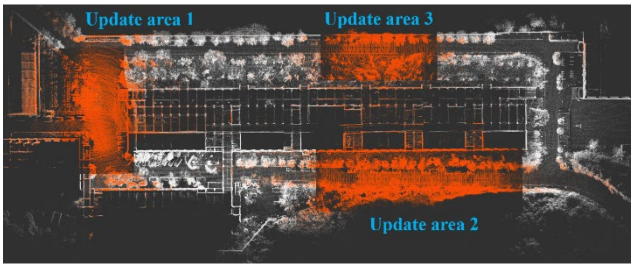
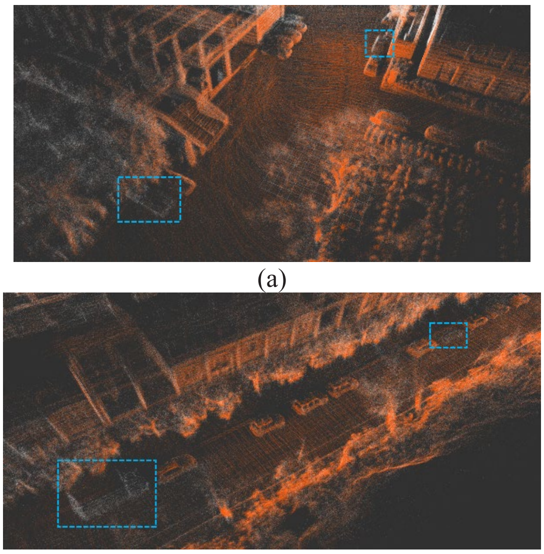


Fig. 15. Map updating and updated area.

Figure 16 shows the update areas 1 and 2. As parked vehicles left the scene, the system detects that the map in this area needs to be updated. The white point cloud in the figure is the original map point cloud that has not been updated, while the orange point cloud is the updated scanned point cloud. The blue dashed box in the figure shows that vehicles have left.



(a)

(b)

Fig. 16. Side view of updated area, (a) area 1 and (b) area 2.

In Fig. 15, there is no vehicle change in the update area 3. However, as the trees beside the road have some differences in the point cloud information when the scan is performed, the matching score in this area is low. Especially taller trees affect more the detection range of an optical radar. As shown in Fig. 17, the blue dashed box in the figure is the tree with a large difference in point cloud information.

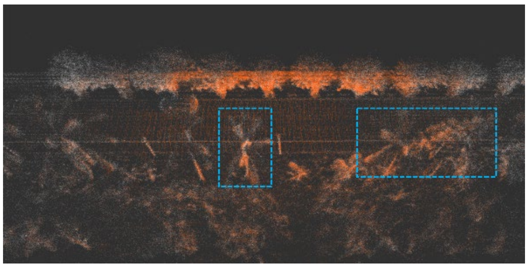


Fig. 17. Side view of updated area 3.

The error value of the three update areas calculated in this experiment is shown in Table 2. The average error of the update area is within the range of 0.3 to 0.5 m, and the standard deviation is between 0.2 and 0.6 m, which is acceptable for self-driving car positioning on the map.

TABLE II. ACCURACY OF MAP UPDATING

Updated area / Error	Area 1	Area 2	Area 3
Average	0.3265m	0.4963m	0.3318m
Max	0.7454m	1.1139m	1.3343m
Standard deviation	0.2024m	0.5524m	0.2526m

IV. CONCLUSIONS

This study proposes a system that performs high-precision vehicle positioning without relying on high-precision GPS. Only relying on LiDAR for positioning may cause irreversible positioning of the autonomous vehicle. Once the positioning has slightly deviated, it causes the positioning to diverge and the vehicle to lose its positioning direction. It is necessary to use other sensors such as ORB-SLAM to assist in positioning correction so that the positioning point of the self-driving car is restored. This study uses a camera to assist LiDAR in positioning and extracts image feature points through ORB-SLAM and map them to anchor points. When the point cloud positioning fails, ORB-SLAM re-sets the initial position and induces the NDT to converge. Therefore, this algorithm allows a greater error tolerance for point cloud positioning.

ACKNOWLEDGMENT

This paper was supported by project of Institute for Information Industry by MOEA, Taiwan, R.O.C.

REFERENCES

[1] Intelligent Vehicles Synposium (IV), Baden-Baden, Germany, Jun 2011, pp. 393-398.

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

[3] C. Glennie and D. D. Lichti, "Static calibration and analysis of the velodyne HDL-64E S2 for high accuracy mobile scanning," Remote Sensing, vol. 2, no. 6, pp. 1610-1624, 2010.

[4] P. J. Besl and N. D. McKay, "A method for registration of 3-D shapes," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 14, no. 2, pp. 239-256, 1992.

[5] P. Biber and W. Strasser, "The normal distributions transform: a new approach to laser scan matching," in Proceedings IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2003, vol. 3, pp. 2743-2748.

[6] J. Serafin et al., "Fast and robust 3D feature extraction form sparse point clouds," in IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Daejeon, South Korea, Dec 2016, pp. 4105-4112.