PERFORMANCE OF LOCALIZATION BASED ON VISUAL SLAM AND LIDAR POINT CLOUD REGISTRATION

P.R. Lu [1] *, K.W.Chiang [1], Y.S.Lin [2], C.Y.Hsu [2], T.H.Yeh [3], P.L.Li [4]

[1] Dept. of Geomatics, National Cheng Kung University, Taiwan - (F64084010, kwchiang)@geomatics.ncku.edu.tw
[2] Automotive Research & Testing Center, Taiwan - jasonlin@artc.org.tw, evan@artc.org.tw
[3] Dept. of Geomatics, National Cheng Kung University, Taiwan – p66101061@gs.ncku.edu.tw
[4] High Definition Maps Research Center, Taiwan - pointlpl@geomatics.ncku.edu.tw

KEY WORDS: Camera, SLAM, LiDAR, Point Cloud, Registration, NDT.

ABSTRACT:

Self-driving car technology has become increasingly popular in recent years. Traditionally, these cars rely on GNSS and INS, but limitations in urban environments can affect their effectiveness. GNSS technology has developed to address this issue, but it still has limitations in certain environments. LiDAR technology has emerged as a solution to obtain high-precision point cloud maps and SLAM technology is now used to integrate these maps with sensor data, such as cameras, to achieve precise positioning. This enables visual SLAM to be performed in environments where GNSS signals are blocked. In this research, we aim to use SLAM technology to obtain posterior environmental information and match it with prior high-precision point cloud maps. The research will start with hardware configuration, actual measurement and analysis of SLAM algorithms, and 3D point cloud matching methods. Results and benefits will be analysed to compare the advantages and disadvantages of point cloud matching algorithms and applicable hardware.

1. INTRODUCTION

In recent years, the smart industry has experienced remarkable growth, particularly in the field of self-driving cars. As AI technology continues to mature, an increasing number of manufacturers are investing in research and development in this area. According to Intel and the SA International Research Institute, the passenger economy generated by fully automatic driving is predicted to create a $7 trillion market by 2050, indicating enormous market development potential. The Boston Consulting Group (BCG) has estimated that the self-driving car market will have an output value of approximately $42 billion by 2025. Additionally, BCG predicts that sales of new cars with self-driving capabilities will account for roughly 25% of total sales in 2035. With this rapid growth, accurate navigation and positioning technology have become increasingly crucial. Self-driving cars traditionally rely on GNSS and INS for positioning, but signal limitations in urban environments and other obstructions have made multi-spatial and integrated positioning technology increasingly popular. Using Simultaneous Localization and Mapping (SLAM) technology, multiple sensors can be used for real-time positioning by comparing features. This research uses SLAM algorithms to generate visual point clouds and match them with LiDAR point clouds. It is expected to be used in GNSS In case of signal failure, precise positioning can still be maintained.

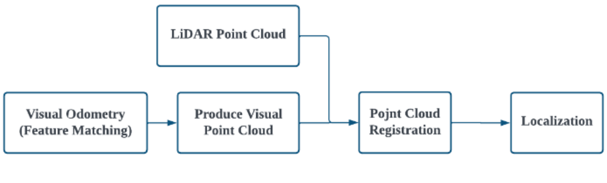

LiDAR Point Cloud
Visual Odometry (Feature Matching)
Produce Visual Point Cloud
Pojnt Cloud Registration
Localization

Figure 1. Process of SLAM and LiDAR point cloud matching

SLAM technology utilizes multiple sensors with image data to achieve real-time positioning and map construction while moving in an unknown environment by comparing features. Currently, there are two main methods of SLAM: Laser SLAM and Visual SLAM. Laser SLAM was the main research method in the early stage, with relatively mature theory, low computational requirements, high accuracy, and immediate response. However, it is expensive and has the problem of passing through object planes due to its long wavelength characteristics. On the other hand, Visual SLAM mainly uses cameras to capture image information, which can extract a large amount of feature information from the environment, and has gradually become the main research focus due to its low cost and wide application range. Based on the main visual sensor, Visual SLAM can be divided into three types: Monocular Camera, Stereo Camera, and Depth Camera (RGB-D Camera).

Visual SLAM (Simultaneous Localization and Mapping) is a technique that uses a camera to obtain images of an unknown environment and construct a map while simultaneously estimating the camera's position and orientation. This is done by comparing observed features in the images to features in the map to determine the camera's location. However, Visual SLAM requires repeated observations of the same location to improve accuracy through loop closure detection. In the context of self-driving cars, repeating the same route multiple times is not practical as it adds unnecessary cost and time. Instead, LiDAR technology has become popular for obtaining 3D point cloud maps of environments by scanning. These maps can be used to achieve high-precision navigation and positioning by matching the real-time point clouds obtained by the vehicle-mounted LiDAR to the pre-built high-precision point cloud map. However, the cost of LiDAR is high, making it impractical to equip each vehicle with one. To overcome this, a lower-cost camera can be used as the sensor instead, and the high-precision point cloud map can be used to replace repeated observations in Visual

* Corresponding author

This contribution has been peer-reviewed.
https://doi.org/10.5194/isprs-archives-XLVIII-1-W1-2023-257-2023 | © Author(s) 2023. CC BY 4.0 License.

SLAM. By matching the point cloud and the obtained image information in real-time, the car can perform positioning without the need to pass through the same location multiple times.

2. RELATED WORK

The concept of Simultaneous Localization and Mapping (SLAM) was first proposed in 1986. Professor A. J. Davison later introduced vision-based monocular SLAM, which combines vision with the SLAM algorithm for positioning(Davison et al., 2007). The architecture of visual SLAM is as **Figure 2**.

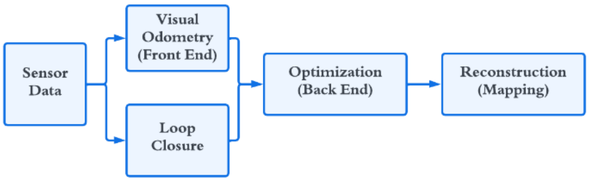


Figure 2 .Visual SLAM flowchart(Davison et al., 2007)

To start, sensor data is used to gather environmental information. The camera is primarily utilized to obtain image data and pre-process it. The pre-processed data is used for estimating camera motion and creating a local map, which is the front-end visual odometry (VO). The loop closure detection is used to determine if the camera has been in the same location before and provides this information to the back end for processing. The back end receives the camera position and attitude observed by the visual odometer at each moment and the information provided by the loop closure detection. The data is optimized to obtain a consistent trajectory and map for the entire area(Optimization), and finally, a map is built based on the processed trajectory data (Mapping).

In 2007, Klein et al. proposed PTAM (Parallel Tracking and Mapping)(Klein & Murray, 2007), which achieved the parallelization of the tracking and mapping process. This research divided SLAM into front and back ends, which is the system design used by most SLAM systems today. This allows the back-end optimization to be performed synchronously in the execution thread. PTAM is also the first SLAM system to use nonlinear optimization. However, the scene size is limited, and tracking can be lost easily.

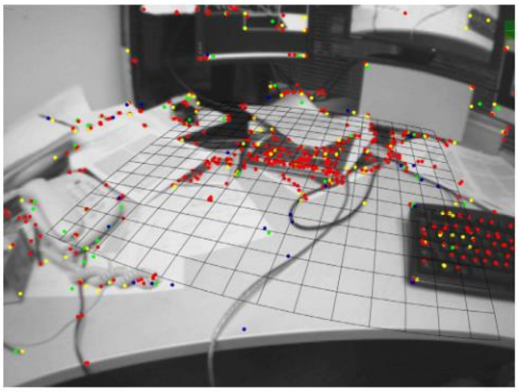


Figure 3. PTAM Demo Result(Klein & Murray, 2007)

In 2014, J. Engel et al. proposed the LSD SLAM (Large Scale Direct monocular SLAM) architecture(Engel et al., 2014), which uses a monocular direct method for feature extraction. The study employs a semi-dense direct method for extracting features from pixels. Specifically, it takes 5 points at equal distances on a line, calculates the sum of squared errors (SSD), and adjusts the scale by standardizing the estimated depth average value. However, the direct method alone is not currently sufficient for loop closure detection, so it needs to be combined with the feature point method.

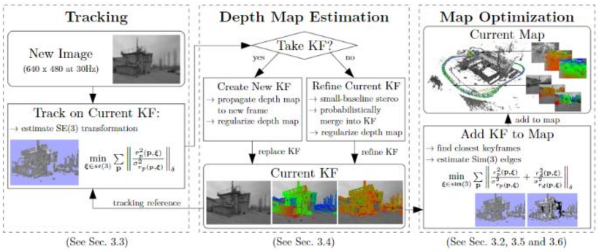

Tracking
New Image
(640 x 480 at 30Hz)
Track on Current KF:
→ estimate SE(3) transformation
$\min_{\xi \in se(3)} \sum_{\mathbf{p}} \left\| \frac{r_p^2(\mathbf{p},\xi)}{\sigma^2_{r_p(\mathbf{p},\xi)}} \right\|_\delta$
tracking reference
(See Sec. 3.3)
Depth Map Estimation
Take KF?
yes
no
Create New KF
→ propagate depth map to new frame
→ regularize depth map
Refine Current KF
→ small-baseline stereo
→ probabilistically merge into KF
→ regularize depth map
replace KF
refine KF
Current KF
(See Sec. 3.4)
Map Optimization
Current Map
add to map
Add KF to Map
→ find closest keyframes
→ estimate Sim(3) edges
$\min_{\xi \in sim(3)} \sum_{\mathbf{p}} \left\| \frac{r_p^2(\mathbf{p},\xi)}{\sigma^2_{r_p(\mathbf{p},\xi)}} + \frac{r_d^2(\mathbf{p},\xi)}{\sigma^2_{r_d(\mathbf{p},\xi)}} \right\|_\delta$
(See Sec. 3.2, 3.5 and 3.6)

Figure 4. Structure of LSD-SLAM(Engel et al., 2014)

In 2015, ORB-SLAM was a feature point method algorithm that used Oriented FAST corner detection and Rotated BRIEF descriptor for feature extraction(Mur-Artal et al., 2015). It employed three threads to complete SLAM: Tracking, Co-visibility Graph, and Essential Graph. The system had good rotation and scaling invariance and reduced calculation time. The Tracking thread calculated feature point positions and roughly estimated camera pose, while the Co-visibility Graph used local bundle adjustment to obtain refined camera pose and feature point spatial position. The Essential Graph performed loop closure detection on the global pose graph, maps, and key frames to eliminate cumulative errors, allowing ORB-SLAM to obtain a globally consistent trajectory map.

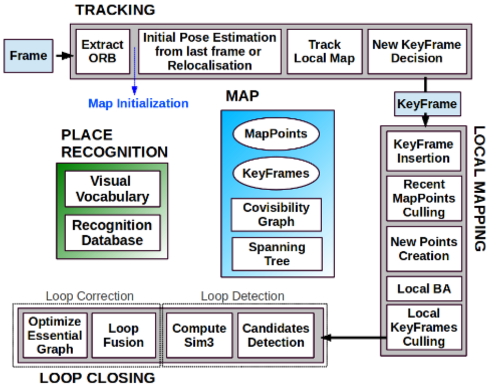


Figure 5. Structure of ORB-SLAM(Mur-Artal et al., 2015)

Visual odometry can be classified into two types: direct method and feature-based method. The direct method estimates the camera motion using the gray value of pixels without the need for feature point operation and descriptors(Chen et al., 2018). It assumes that the gray value of the same pixel in two images is constant, and the camera pose is estimated using the minimum photometric error between two pixels. According to the source of the known point P, it is classified into three categories: Sparse Direct Method (SDM), Semi-dense Direct Method (SDDM), and Dense Direct Method (DDM).

The sparse direct method (SDM) estimates the camera poses using sparse key points, and it tracks the photometric positions of pixels instead of using descriptors. Since it doesn't require descriptor calculation, it saves computing time and is a key point-based pose estimation method. The Semidense Direct Method (SDDM) estimates P using pixels with significant pixel gradients and a large number of voxels, similar to the Sparse Direct Method. However, SDDM is a more sophisticated version of the direct method that uses more pixels to estimate the pose, making it more accurate and stable. LSD-SLAM employs this method to ensure reliable and immediate tracking. The Dense Direct Method (DDM) uses all pixels in the image for estimating camera motion, which means all pixels contribute to the calculation. However, because it involves a large amount of data, this method requires significant computing resources and high-quality hardware equipment to achieve accurate results.

When the gray value of the image is unstable due to factors such as light sources and materials, we need to use feature points in image analysis as reference points to estimate the camera position. Feature point methods find stable points during camera movement, and some well-known algorithms include Scale-invariant feature transform(SIFT), Oriented FAST and Rotated BRIEF(ORB), and Speeded Up Robust Features(SURF).

The Scale-invariant feature transform (SIFT) algorithm uses a multistage process to identify stable feature points in images. First, the image is down-sampled using Laplassian of Gaussian (LOG) to create a pyramid of images at different resolutions. Then, each layer of the pyramid is filtered using Gaussian convolution. In each image, pixels with a large gradient of gray value across the entire image are identified by computing the Difference of Gaussian (DOG) between adjacent images, resulting in a new pyramid. Key points are then selected from these pixels based on extreme values. The descriptors for these key points are used as SIFT feature points. This method considers luminosity, scale, and rotation, but requires a significant amount of computation, making it impractical for use in most SLAM architectures. (Lowe, 2004)

ORB (Oriented FAST and Rotated BRIEF) is a feature detection and description method used in computer vision. It combines the advantages of the FAST corner detection algorithm and the BRIEF descriptor. FAST is known for its fast processing speed and the ability to detect areas where the gray levels of local pixels change significantly. However, it lacks directionality and has scale uncertainty. ORB solves these problems by using image pyramids to detect key points at different scales, and using the gray scale centroid method to calculate the center of gravity of key points within a certain range to define a main direction. The BRIEF descriptor is then used to describe the key points by randomly selecting pixels around the key points and using binary encoding. This method has the advantages of fast speed and convenient storage, making it suitable for feature analysis, especially in areas where the scene changes are not obvious.(Rublee et al., 2011)

The SURF algorithm uses a BOX filter to quickly calculate approximate Laplacian of Gaussian, allowing for real-time applications such as tracking and object recognition. It uses the Integral Image or Summed-Area Table to quickly calculate the average intensity of pixels within a given range and employs the Hessian matrix to select feature point positions based on quadratic differential extrema. The algorithm assigns a main direction to each feature point and uses Haar wavelet to operate on the image, with the calculation amount being independent of the selection range.(Bay et al., 2006)

LiDAR point cloud matching methods are divided into three types: (1) Point-Based Scan Matching; (2) Feature-Based Scan Matching; and (3) Scan Matching Based on Mathematical Characteristics(Ren et al., 2019). The most common point-based method is the Iterative Closest/Corresponding Point (ICP), which finds the corresponding relationship between points in space using iterative methods(Marani et al., 2016). In the case where the density of the camera and lidar point clouds differs significantly, there may be a situation where the matched scale cannot converge. The feature-based method matches feature points and structural lines or planes, typically using curvature or normal, to achieve high-precision pose estimation, but can suffer from poor estimation accuracy in scenes with inconspicuous feature textures. The most common feature-based method is the Lidar Odometry And Mapping (LOAM)(Zhang & Singh, 2014). The distribution method characterizes scan data and pose changes using mathematical properties, with the Normal Distributions Transform (NDT) being the most famous. NDT subdivides the model's space into regularly sized pixels and can obtain more accurate and stable results compared to point-to-point matching, especially when the overlapping area is small or the initial alignment method is poor(Magnusson et al., 2007). This study mainly uses NDT as the research method because ICP requires a large amount of computation and may match to noise points, causing positioning errors, which is not suitable for ORB-SLAM2 that uses feature extraction to extract point clouds.

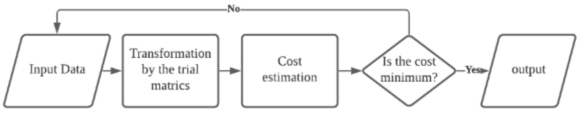


Figure 6. ICP Flow Chart(Marani et al., 2016)

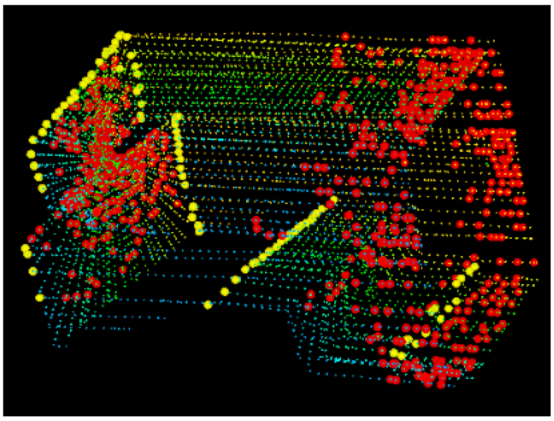


Figure 7. The schematic diagram of the feature points extracted by the LOAM algorithm.(Zhang & Singh, 2014)

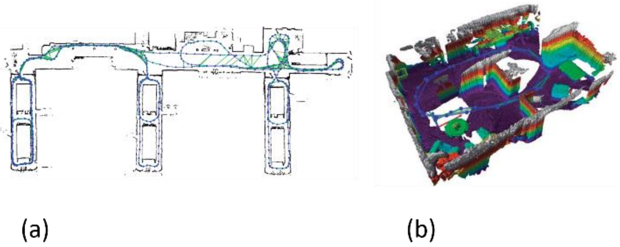


Figure 8. (a) 2D NDT map (b) 3D NDT map

(Magnusson et al., 2007)

3. METHODOLOGY

This study focuses on utilizing visual SLAM and LiDAR point clouds for positioning and navigation research on a vehicle. The sensor is mounted on the vehicle, which moves to obtain spatial information and match it with the established 3D point cloud. The algorithm will be tested first in a small field using a cart and then in a larger field using a vehicle. The research process involves five steps also shown in Figure 9, which are: (1) collecting spatial data using the camera, (2) extracting feature points from camera data, (3) using the prior LiDAR point cloud map to perform matching operations, (4) determining the camera position and updating the trajectory, and (5) integrating and developing future positioning methods.

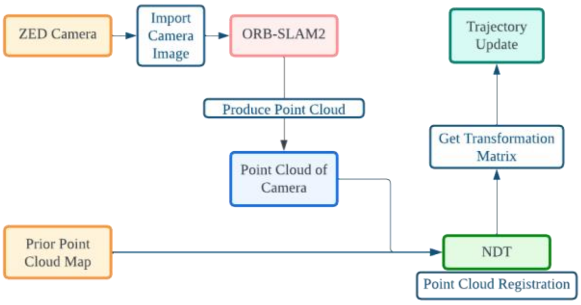


Figure 9. Flowchart of the implementation process

3.1 Hardware

This research mainly uses the ZED stereo camera to collect spatial information, and uses the VLP-16 lidar to create environmental point clouds in the experiment for algorithm testing purposes.

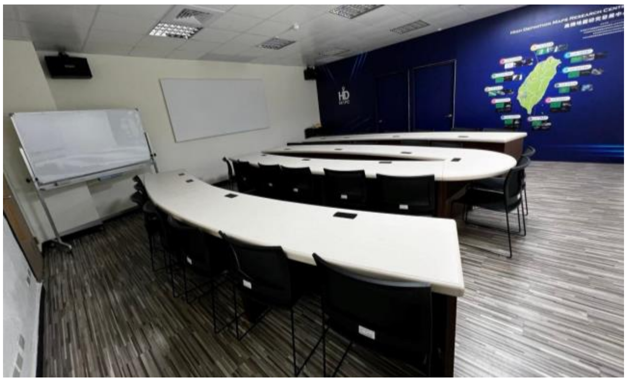
Figure 10. Zed Camera

Figure 11. VLP-16 LiDAR

3.2 Experimental field

In this study, we conducted small-scale tests in various fields to evaluate the performance of the algorithms in a larger setting. Field 1 was a conference room environment, which provided a relatively monotonous testing ground for the ORB-SLAM2 algorithm. However, we encountered some issues with lost tracking. Field 2 was a laboratory and corridor area, where irregularly shaped objects such as desks and chairs, as well as debris, offered more features to track than Field 1. In addition, the algorithms were tested in a linear field such as the corridor to assess their performance in that domain. The third field was the underground parking lot of the library, where sensors were mainly installed on the vehicle to simulate a real-world environment and improve the algorithm's performance.

3.2.1 **Field 1:** Conference room



3.2.2 **Field 2:** Laboratory and Corridor

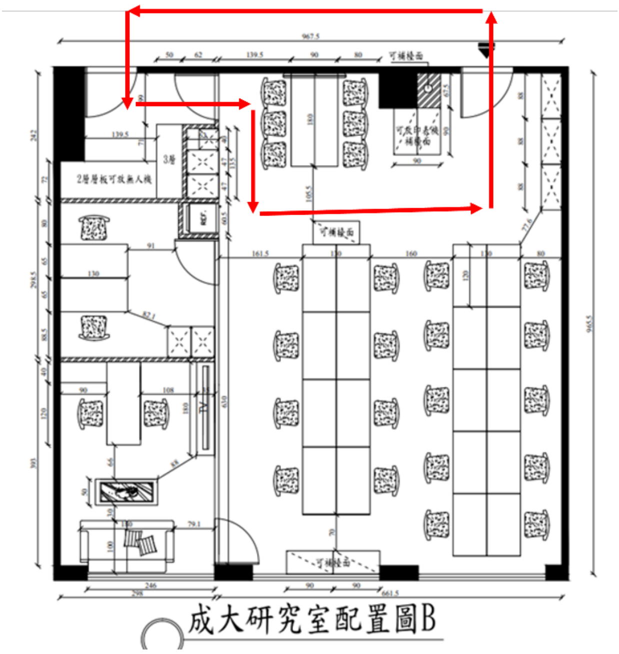


Figure 12. Laboratory and Corridor Route map

3.2.3 **Field 3:** Underground parking lot of National Cheng Kung University Library

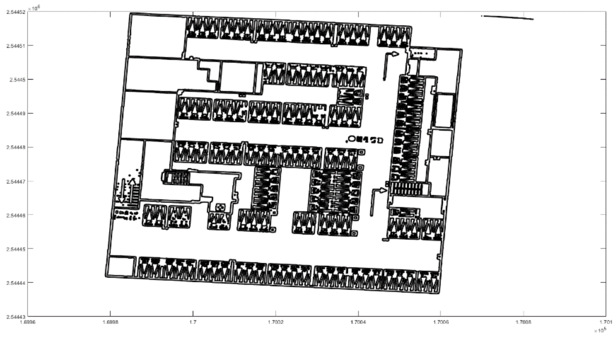


Figure 13. Underground parking lot floor plan

3.3 Point cloud data

3.3.1 Laboratory and Corridor Point Cloud Map: In the small field research, the VLP-16 LiDAR was used to collect environmental information, and then the LOAM algorithm(Zhang & Singh, 2014) was used to generate a laboratory point cloud map, which was used as a base map for rough point cloud matching.

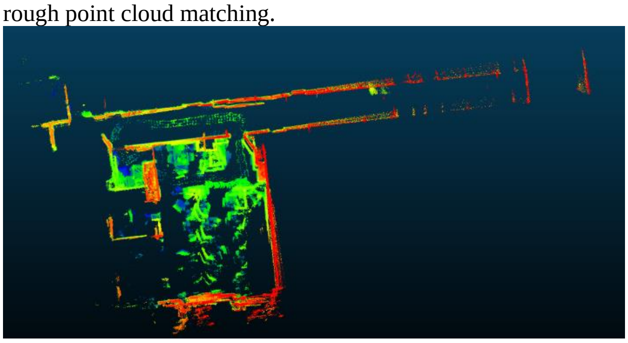


Figure 14. Point Cloud Map of Laboratory and Corridor

3.3.2 Underground parking Point Cloud Map: In the experimental field of the underground parking lot, the HDL-64 lidar is utilized to generate denser point cloud maps in the general map. To produce these point clouds, the Lili-om algorithm(Kailai Li, 2021) is employed, which is an improved version of the LOAM algorithm. **Figure 15** shows the results obtained.

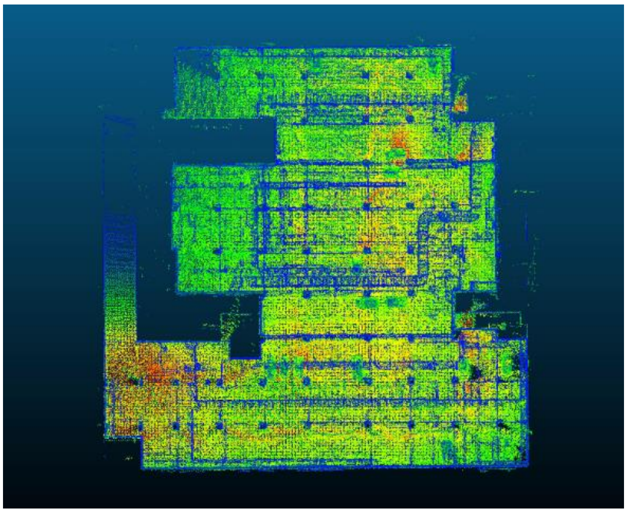


Figure 15. Point Cloud Map of Underground parking

4. RESEARCH RESULTS

4.1 Field 1: Conference room

In this experiment, we aimed to collect environmental data by using a camera loaded on a cart to patrol the interior of a meeting room. We modified the original code of ORB-SLAM2, specifically the Tracking thread, and used the PCL library to convert the feature points in each current frame into a pcd file. The result is presented in the **Figure 16**. However, using the point cloud of the entire data for matching is not practical for actual application. Therefore, we attempted to modify the code to output the point cloud of each current frame into a pcd file, and simultaneously change the field. This modification is expected to enhance the SLAM effect in a field with more features.

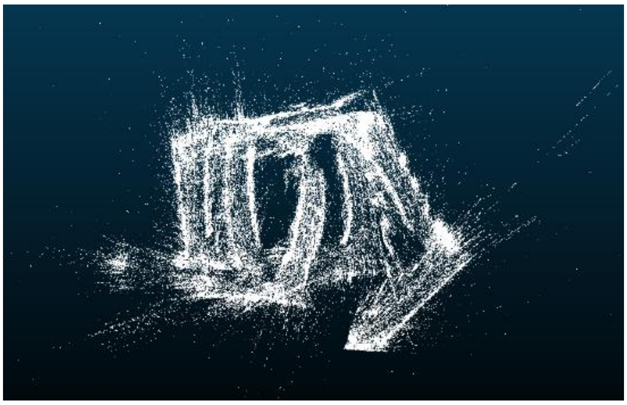


Figure 16. Point Cloud of Entire Dataset

4.2 Field 2: Laboratory and Corridor

For the second field, we moved on to the laboratory and corridor areas, which had more distinctive features. We loaded the camera onto the cart and travelled around the field twice. Using ORB-SLAM2, we generated a point cloud for each frame, as shown in the **Figure 17**. Then we down-sampled the point cloud image produced by LOAM and attempted to match the visual point cloud of each frame with the LiDAR point cloud in the form of NDT, in order to obtain the transformation matrix between the two systems. **Figure 18** shows the point clouds of the two systems before matching. The white point cloud is the LiDAR point cloud, and the red point cloud is the visual point cloud. After matching, the visual point cloud overlaps with the LiDAR point cloud like **Figure 19**. After this test, we used the ROS system to visualize the entire matching process, and a video of the entire matching update process was captured. By viewing the trajectory shown in **Figure 20**, it was found that the updated position is closer to the lidar trajectory. This shows that matching the visual point cloud with the lidar point cloud can improve the positioning accuracy of low-cost sensors.

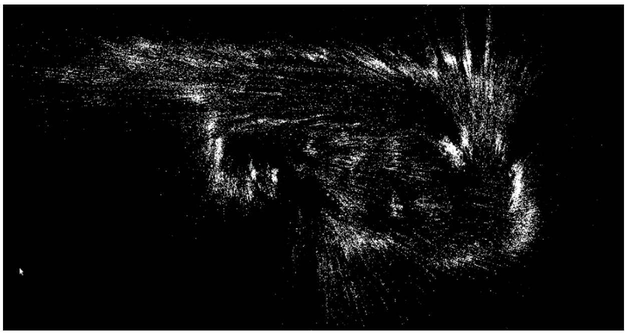


Figure 17. Point Cloud of Laboratory and Corridor

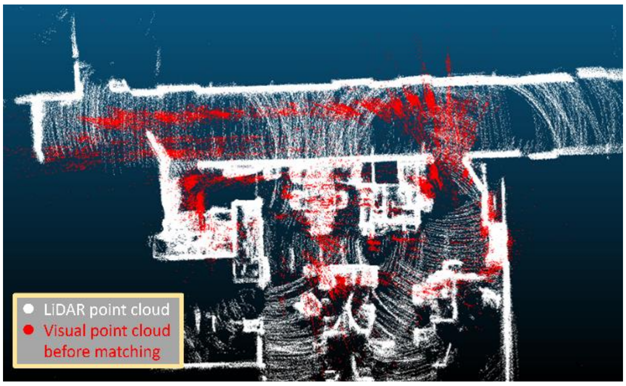


Figure 18. Before NDT Matching

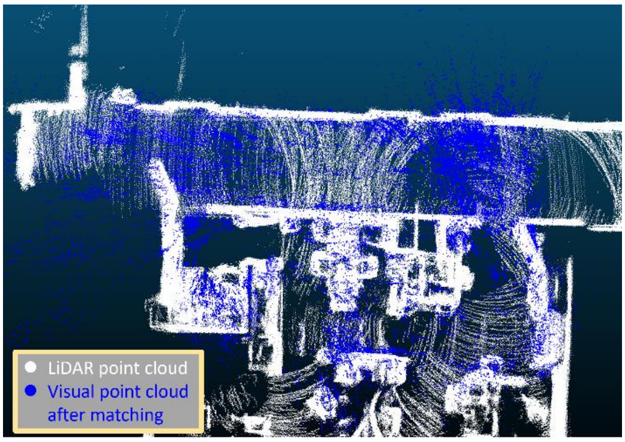


Figure 19. After NDT Matching

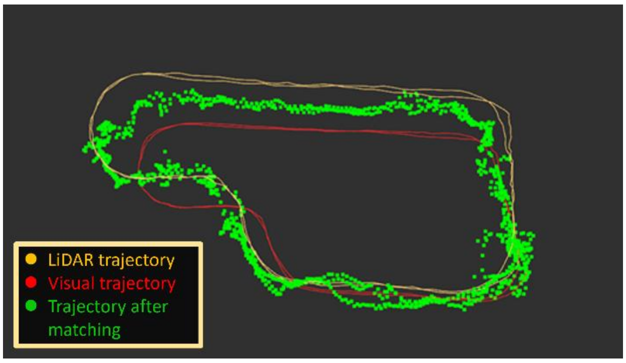


Figure 20. The trajectory of the entire process in conference room

4.3 Field 3: Underground parking lot of National Cheng Kung University Library

After implementing and testing the algorithm in a small field, the environment data in a large field is collected in a vehicular manner, and the underground parking lot of the main library is chosen as the experimental field.

The same steps were taken to import both visual and lidar point clouds into the integrated program, and the resulting trajectory is shown in **Figure 22**. The shape of the green matched trajectory is similar to that of the original camera trajectory, because in the trajectory updating settings, this study uses the camera trajectory as the target for transformation. Therefore, if the correction amount is not significant, the trajectory will be dominated by the camera trajectory, which results in the phenomenon of similar trajectories. This indicates that the point clouds were not matched, and the trajectory was almost not updated. Further adjustments are needed in the matching algorithm in the future.

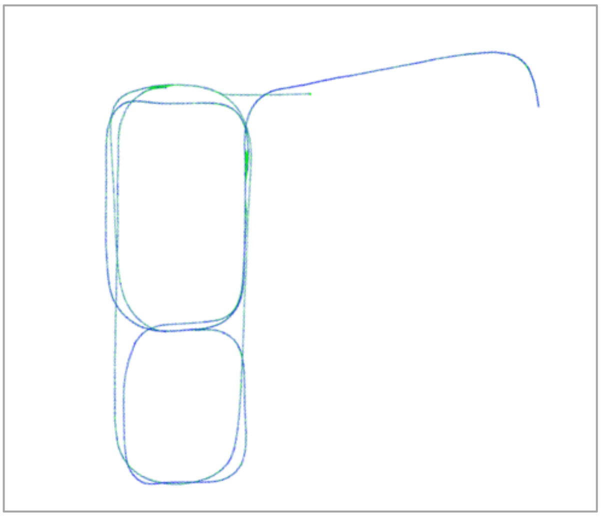


Figure 21. Camera Trajectory of Underground Parking lot

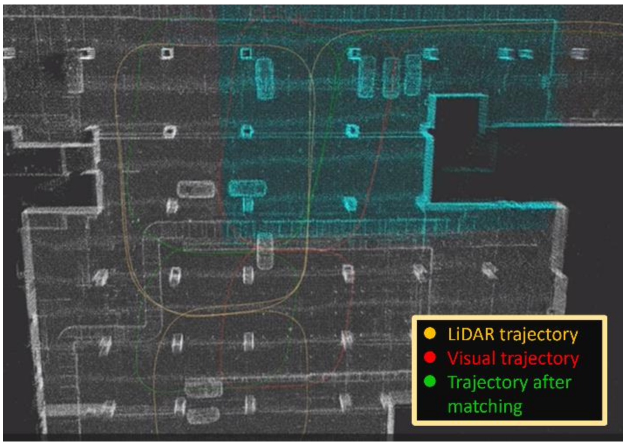


Figure 22. Trajectory of Underground parking lot

5. CONCLUSIONS

In small-scale field tests, the use of low-cost sensors such as cameras, coupled with point cloud matching with prior knowledge of the point cloud map, can improve the accuracy of the sensor's raw trajectory, but there are still many problems to be overcome in the process. Although running ORB-SLAM2 does not suffer from the scale problem that monocular cameras have, it still experiences drift at turning points due to the number of feature points and initialization position issues. Therefore, it requires high-precision maps for localization. In large-scale field tests, errors occurred in the time synchronization of visual point clouds and lidar point clouds, resulting in lower-than-expected results. Therefore, the entire algorithm needs improvement to synchronize the camera and lidar data in time to improve matching accuracy.

Currently, point clouds are only extracted and matched with high-precision maps in stages. In the future, it is expected to integrate this process into ORB-SLAM2. If ORB-SLAM2 is executed, it outputs the position and point cloud of the current frame, then simultaneously matches that data to obtain a new position using point cloud matching, and uses EKF to integrate point cloud matching with ORB-SLAM2 for prediction and matching for updating.

We are also attempting to incorporate IMU to develop Visual Inertial Odometry, which improves the shortcomings of monocular cameras in terms of scale and compensates for situations where the camera cannot capture sufficient environmental information due to high dynamics and limited frame rate, providing a high-frequency position output.

REFERENCES

Bay, H., Tuytelaars, T., & Van Gool, L. (2006, 2006//). SURF: Speeded Up Robust Features. Computer Vision – ECCV 2006, Berlin, Heidelberg.

Chen, Z., Sheng, W., Yang, G., Su, Z., & Liang, B. (2018, 4-8 July 2018). Comparison and Analysis of Feature Method and Direct Method in Visual SLAM Technology for Social Robots<sup>*</sup>. 2018 13th World Congress on Intelligent Control and Automation (WCICA),

Davison, A. J., Reid, I. D., Molton, N. D., & Stasse, O. (2007). MonoSLAM: real-time single camera SLAM. *IEEE Trans Pattern Anal Mach Intell*, *29*(6), 1052-1067. https://doi.org/10.1109/TPAMI.2007.1049

Engel, J., Schöps, T., & Cremers, D. (2014, 2014//). LSD-SLAM: Large-Scale Direct Monocular SLAM. Computer Vision – ECCV 2014, Cham.

Kailai Li, M. L., Uwe D. Hanebeck. (2021). Towards High-Performance Solid-State-LiDAR-Inertial Odometry and Mapping. *Ieee Robotics and Automation Letters*. https://doi.org/https://doi.org/10.1109/LRA.2021.3070251

Klein, G., & Murray, D. (2007, 13-16 Nov. 2007). Parallel Tracking and Mapping for Small AR Workspaces. 2007 6th IEEE and ACM International Symposium on Mixed and Augmented Reality,

Lowe, D. G. (2004). Distinctive Image Features from Scale-Invariant Keypoints. *International Journal of Computer Vision International Journal of Computer Vision*, 91–110.

Magnusson, M., Lilienthal, A., & Duckett, T. (2007). Scan Registration for Autonomous Mining Vehicles Using 3D-NDT. *Journal of Field Robotics*, *24*, 803-827. https://doi.org/10.1002/rob.20204

Marani, R., Renò, V., Nitti, M., D'Orazio, T., & Stella, E. (2016). A Modified Iterative Closest Point Algorithm for 3D Point Cloud Registration. *Computer-Aided Civil and Infrastructure Engineering*, *31*(7), 515-534. https://doi.org/10.1111/mice.12184

Mur-Artal, R., Montiel, J. M. M., & Tardos, J. D. (2015). ORB-SLAM: A Versatile and Accurate Monocular SLAM System. *IEEE Transactions on Robotics*, *31*(5), 1147-1163. https://doi.org/10.1109/tro.2015.2463671

Ren, Z., Wang, L., & Bi, L. (2019). Robust GICP-Based 3D LiDAR SLAM for Underground Mining Environment. *Sensors*, *19*(13). https://doi.org/10.3390/s19132915

Rublee, E., Rabaud, V., Konolige, K., & Bradski, G. (2011, 6-13 Nov. 2011). ORB: An efficient alternative to SIFT or SURF. 2011 International Conference on Computer Vision,

Zhang, J., & Singh, S. (2014). LOAM : Lidar Odometry and Mapping in real-time. *Robotics: Science and Systems Conference (RSS)*, 109-111.