A Research of Multi-session MAP Reconstruction

Li Cheng Shen B09902019, Chun Sheng Lee R11521608, Jing Heng Lin D11631001

December 26, 2022

1 Introduction

3D reconstruction has been proved useful in various fields such as industrial manufacturing, autonomous driving, archaeology, and augmented reality. Structure-from-Motion (SfM) methods reconstruct 3D point clouds from a collection of images based on feature matching and pose estimation. Several open source implementations of SfM are available, notably COLMAP [SF16, SZPF16] and Meshroom [GGC+21]. A similar task, visual SLAM (Simultaneous Localization and Mapping), aims to estimate the camera pose and 3D structure of the scene in real time, with ORB-SLAM [MAMT15] being one of the most successful SLAM systems.

When performing 3D reconstruction in large-scale dynamic scenes, it is often difficult or inefficient to complete the task in a single session. This yields a new challenge in which 3D reconstruction is done by carrying out multiple mapping sessions and forming a joint map. Several multi-robot SLAM approaches have been proposed to tackle the problem of creating a single map using multiple robots [YFKGH20], but most of them assume the robots are active at the same time and work by extending single-session SLAM methods to operate online in a distributed manner.

In practical applications, it is often desired that large-scale scenes can be reconstructed by merging the results of multiple camera sessions performed at different times, with different sensors, and under different environmental conditions. Therefore, in our final project, we study the multi-session mapping problem. Our general approach is to construct individual point clouds from single-session video frames, and then combine the point clouds from multiple sessions into a single merged map. We collect our own data set, experiment with several methods and frameworks, and provide our results, observations and discussions.

2 Related Work

2.1 Structure from Motion and Visual SLAM

Reconstructing 3D structure from monocular video or images is a long-standing problem. When performed offline, this problem is known as Structure-from-Motion (SfM). The general approach to SfM is to compute 2D point correspondences from extracted features across the input views, and estimate 3D point locations and camera poses that minimize a reprojection error given these observations [TMHF00]. Such modern formulation and approach were proposed in [FCSS10, AFS+11], and refined in [SF16]. Visual SLAM addresses a similar problem, but requires operation in real time. State-of-the-art approaches include similar feature matching methods [MAMT15], while some other approaches are based on dense stereo matching [NLD11].

2.2 Multi-session mapping

When constructing 3D maps in large-scale scenes, in order to enhance the efficiency, accuracy and robustness, it is often necessary to carry out multiple camera sessions or enable the cooperation of multiple robots, and then jointly create a single map. To this end, a simple method for map merging is to leverage the iterative closest point (ICP) algorithm [CM92, BM92] to estimate a transformation for alignment of point clouds. More advanced methods can be employed, and as an example, the multi-session mapping feature provided in the maplab framework [SDF+18] performs loop closure and optimization using heuristics based on vertex distance, orientation, and landmark covisibility.

Moreover, ORB-SLAM3 [CEG+21] can handle map merging in real time during SLAM sessions using their proposed Atlas system. It is also worth noting that multi-session mapping is useful for capturing the environment in differing visual conditions for long-term localization, which has been studied in [CN13] and [BGS+16].

3 Methodology

Our methods come in two phase. The first is point cloud reconstruction. The second is point cloud map merging Figure 1 shows the flowchart of our approaches.

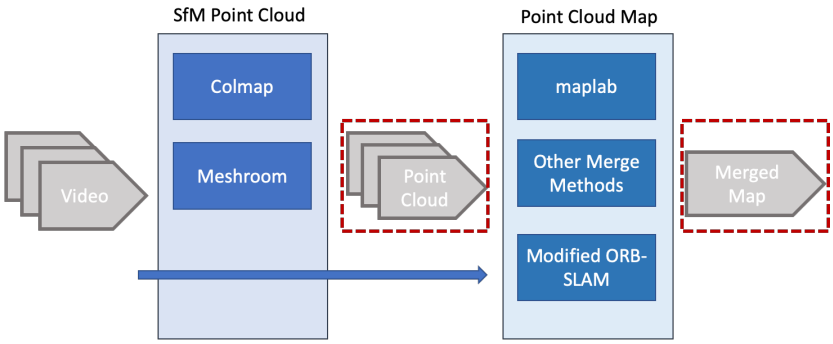

Figure 1: Flowchart of the experiments.

In the first stage of our study, we will extract video frames and use two different reconstruction frameworks, COLMAP and Meshroom, to build a point cloud. We will experiment with different parameters in these frameworks in order to identify the optimal tuning for building the point cloud. This will involve analyzing the resulting point clouds and evaluating their quality and accuracy. In the second stage of our study, we will evaluate several methods for merging point clouds to combine point clouds from different sources or sessions. To achieve this, we will apply three different methods: the multi-session mapping tool from the MapLab framework, direct manual merging of point clouds, and the iterative closest point method provided by CloudCompare. In addition to the above, we additionally tried to use ORB-SLAM as an experimental method.

3.1 COLMAP

COLMAP is a Structure-from-Motion (SfM) and Multi-View Stereo (MVS) pipeline with a graphical and command-line interface. We use it to create our point cloud model. RGB frames are input to this construction system, then we use SIFT as a feature extractor which is the default for COLMAP. After geometric verification, it will calculate triangulation and bundle adjustment to reconstruct the model. In the sequential matching part, we use a pre-trained vocabulary tree with 256K visual words for loop detection. Figure 2 shows the flowchart of COLMAP.

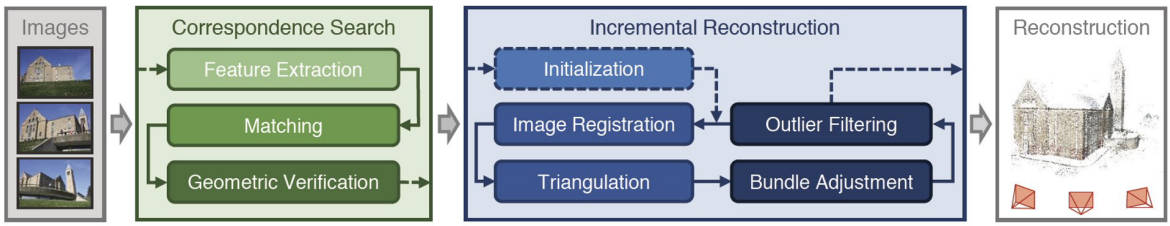

Figure 2: Flowchart of COLMAP.

3.2 Meshroom

Meshroom is a 3D reconstruction software that utilizes a combination of Structure from Motion (SfM) and Multi-View Stereo (MVS) algorithms to generate 3D models from a set of images. It is capable of handling large datasets and can generate high-quality 3D models with a wide range of image sets. To reconstruct the 3D point cloud, Meshroom generates an image pair list to determine the optimal order for aligning the images. The workflow of Meshroom is depicted in Figure 3.

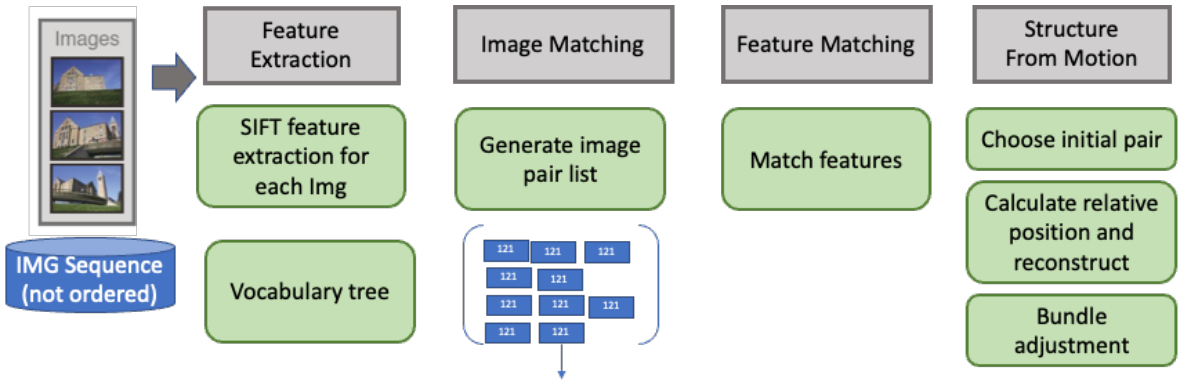

Figure 3: Flowchart of Meshroom.

3.3 Maplab

Maplab consists of two parts. One is the ROVIOLI frontend which is responsible for mapping and localization, and the other is the offline maplab console which can refine or merge the maps. In Figure 4, we can see that using the ROVIOLI system to create maps with VIO or LOC modes, then with maps, we can apply a lot of algorithms in the console part on these maps.

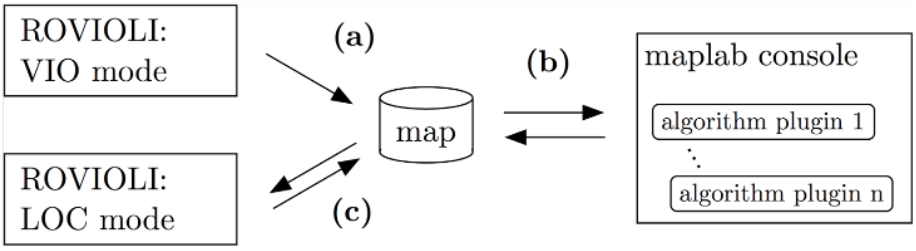

Figure 4: This is the typical workflow with maplab.

3.4 ORB-SLAM

ORB-SLAM uses a combination of feature-based and direct methods to achieve map construction and localization. It first detects distinctive features in the images, called ORB (Oriented FAST and Rotated BRIEF) features, which are used to track the motion of the camera and construct the map. ORB-SLAM also utilizes direct image alignment to optimize the map and camera pose estimates. ORB-SLAM provide a well loop closure method, it can correct any accumulated errors in the map and camera pose estimates by aligning the current image with the image of the revisited location. By utilizing loop closure, we can merge segmented maps by aligning the overlapping regions between the segments.

4 Experiment

4.1 Dataset

In order to collect data from different sections, we split the library into three parts, and will model the perimeter and corridors respectively. The collection route is shown in Figure 5

Figure 5: This is the route of our data collection.

Our data collection process uses a Realsense-D435 camera to record the video. To ensure that the recorded video would be of the equal camera intrinsic parameters, we set the video format at 15fps 720p. This frame rate and resolution combination provided a balance between image clarity and file size, making it an optimal choice for our needs. After the video was recorded, we needed to extract it from its original format and convert it to a series of individual PNG files. By extracting the video in this way, we were able to work with the individual frames at a different frame rate for further SfM reconstruction methods. Images taken from the perimeter and corridors are shown below in Figure 6.

Figure 6: Data collection inside.

The results of our data collection, Figure 7, show that in the corridor section, there were significant changes in the background light and shadow due to the fact that the photos were taken at different times. However, the characteristics of the buildings in the corridor remained relatively stable, as the corridors were well-lit. Compared with the corridor section, our data shows that the images around the library vary significantly, with a high degree of tree coverage and occlusions. In order to minimize the difficulty of constructing a map in this area, we selected similar time periods to capture the images in order to reduce the variability in the data.

4.2 Point Cloud Reconstruction

For the point cloud analysis, we first used the Structure-from-Motion (SfM) framework Meshroom to generate a point cloud. The results are shown below in Figure 9. The point clouds in the figure correspond to three different routes of image capture, denoted as "a", "b", and "c". Route "a" exhibits

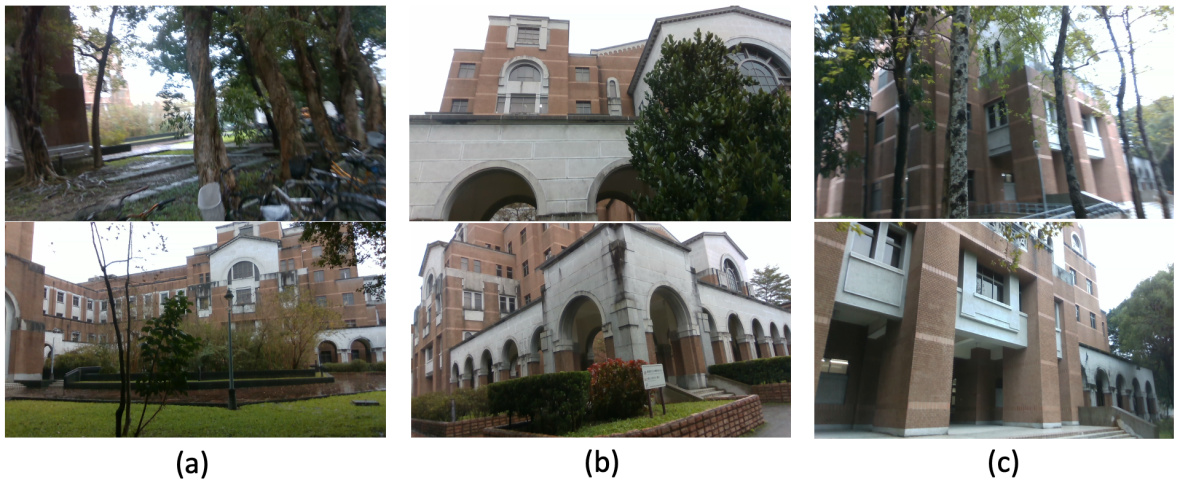

Figure 7: Data collection outside.

the greatest degree of tree coverage, leading to limited identification of building features. Route "b", on the other hand, provides clear identification of the frontal features of the library. Route "c" falls in between routes "a" and "b" in terms of the identifiable features of the building. Figure 9 illustrates the correct drawing of corridor tracks.

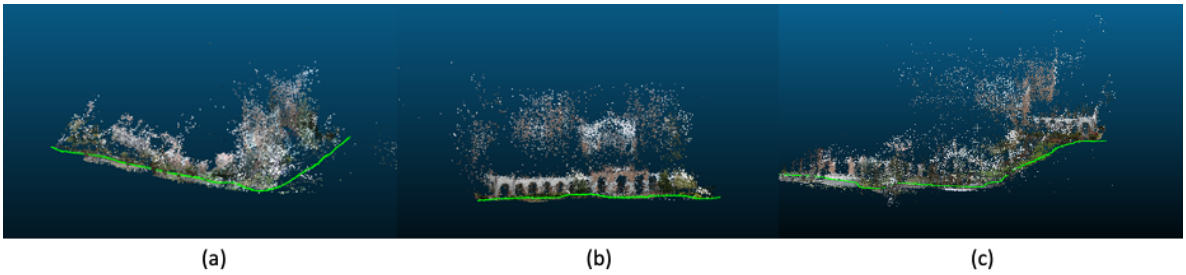

Figure 8: : Outside 7fps with Meshroom.

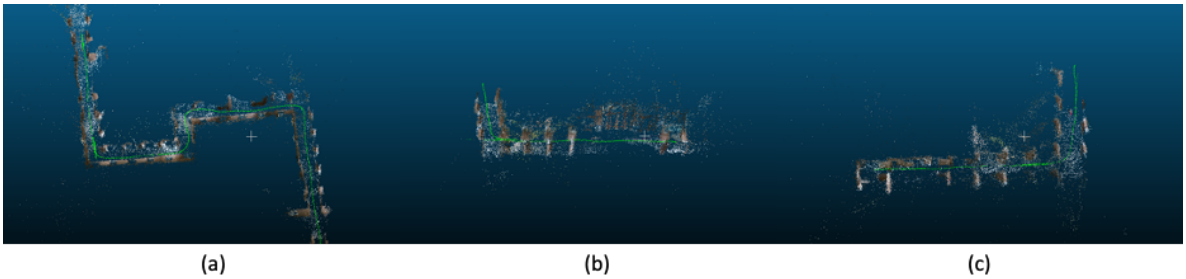

Figure 9: Inside 7fps with Meshroom.

We tried to build the continuous images through Meshroom, and the result is shown in Figure 10. In the non-segmented peripheral map, the whole library building point cloud is clearly reconstructed, but the details of the building's edges are not sharp due to tree occlusion. Figure 10 (b) shows the result of our attempt to construct a continuous map for the corridor section. The resulting trajectory does not match the actual map of the library, indicating a failure in building the map. On the right side of the figure, we can observe overlapping trajectories, which suggests the combination of different corridors.

In the COLMAP part, we tried to build the inside part of the library. First, we set 3 fps for

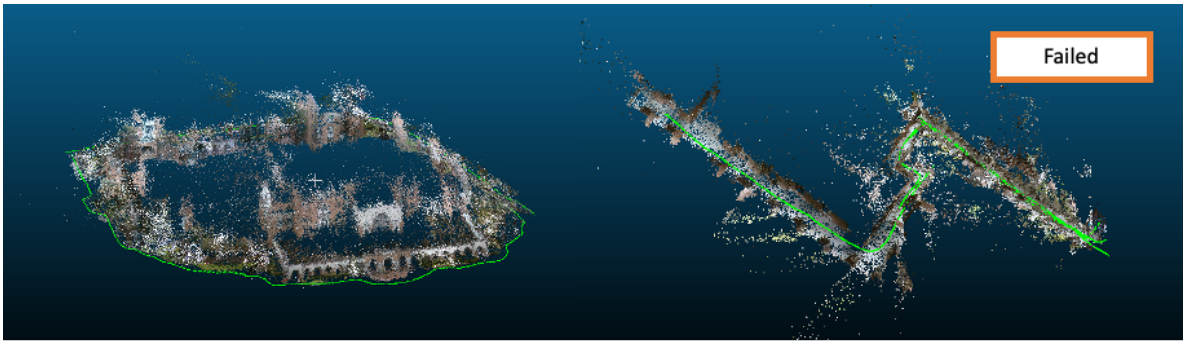

Figure 10: Non-segmented point cloud with Meshroom:(a) Outside, (b) Inside.

extracting RGB frames from video streams. Figure 11 shows the point clouds of each segment. And the result of the merged map is shown in Figure 12. We can observe that there is a branch on the lower right corner in Figure 12, but it should be reconstructed on the other side. the columns at the inside corridor is clear while we zoom in to see the local part.

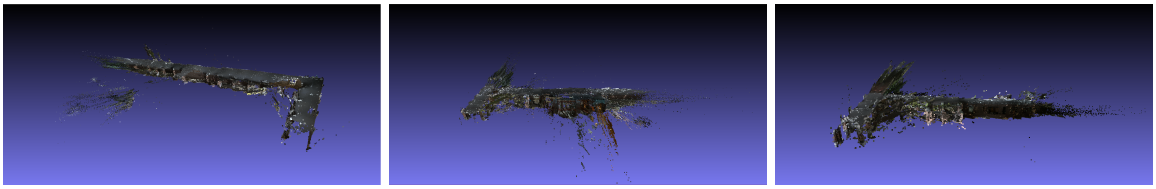

Figure 11: Point clouds of each segment on COLMAP with 3fps.

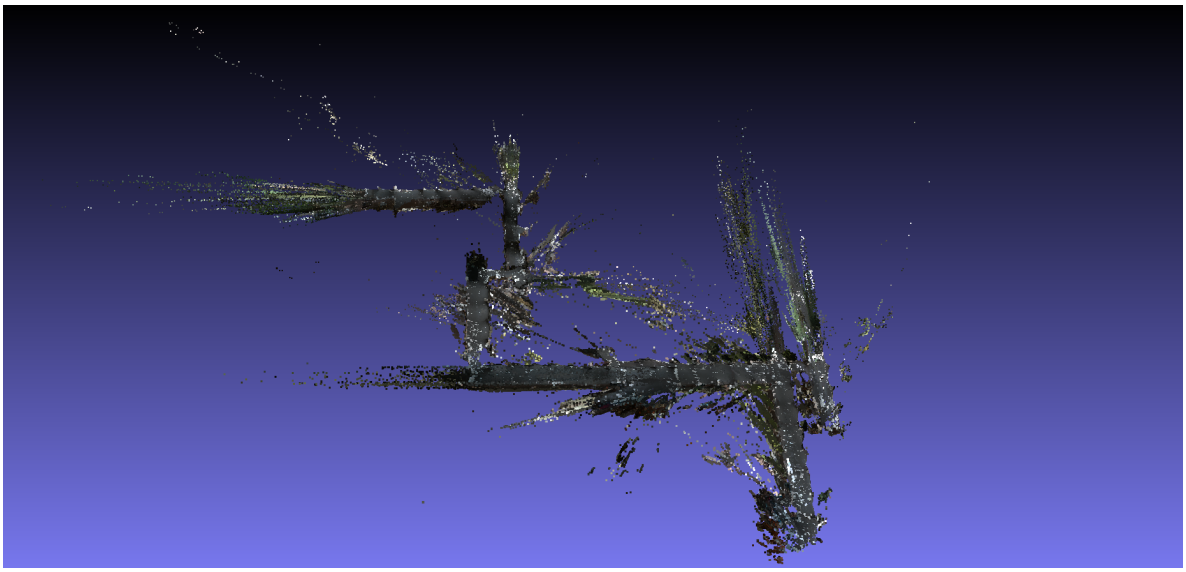

Figure 12: Point clouds of the merged map on COLMAP with 3fps.

The process of modifying ORB-SLAM to generate dense point clouds was only partially successful. We were able to input our segmented MP4 images into ORB-SLAM through modification. The visual comparison results, shown in the Figure 14, indicate that ORB-SLAM can create a map with a correct trajectory using local feature matching optimization.

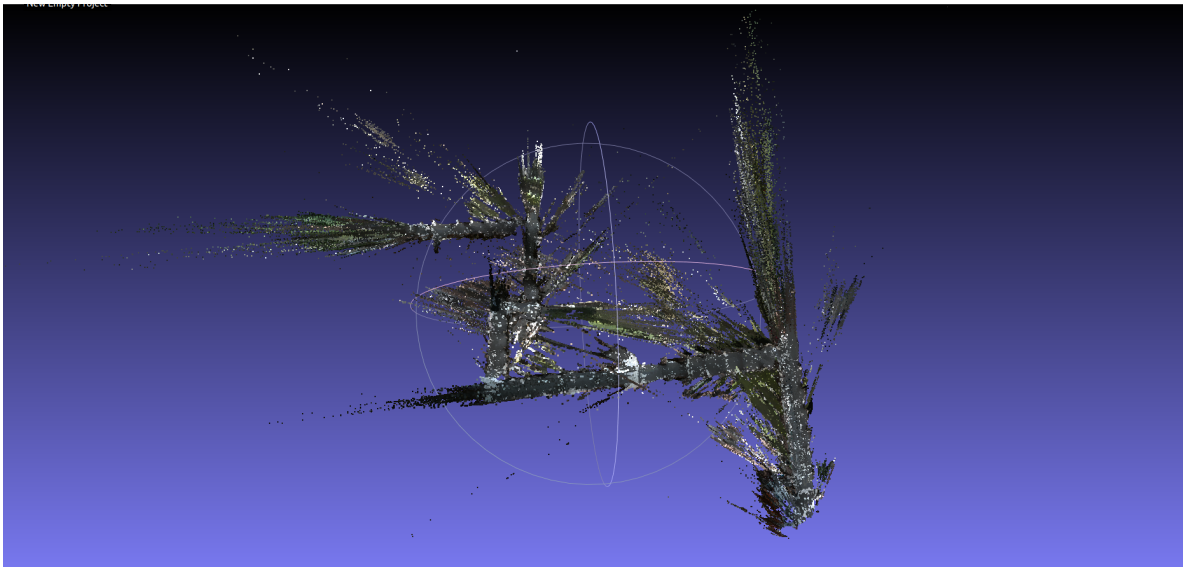

Figure 13: Point clouds of the merged map on COLMAP with 7fps.

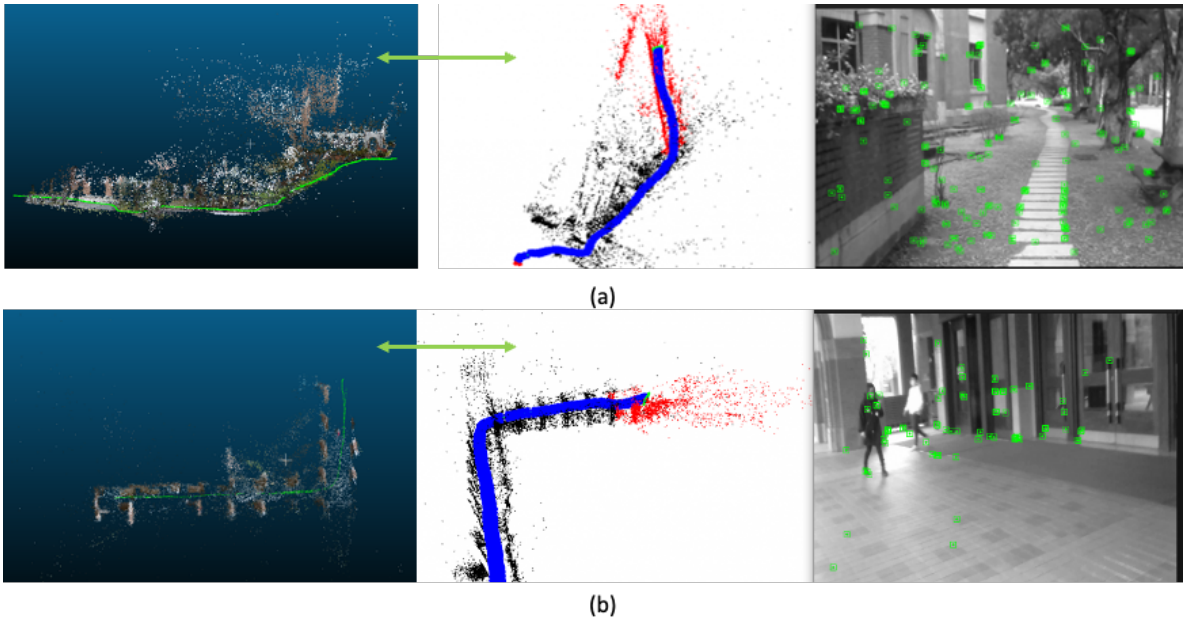

Figure 14: Visualized comparison with ORB-SLAM3 output:(a) Outside session, (b) Inside session.

4.3 Map Merging

We attempted to use the iterative closest point (ICP) method provided by the CloudCompare framework to calculate and merge similar points through iterative transformation of the point cloud. However, this method alone was not effective in merging the point clouds, as it directly merged the densest part of the point cloud, but however, in the wrong direction.

To improve the merging of the point clouds, we manually selected four corner points as references and applied a rigid body transformation to the point cloud. The results are shown in Figure 16 (a). After applying the transformation, we used the ICP method again to merge the point clouds, as shown in Figure 16 (b). It can be observed that the ICP method performs better in combining the point clouds within the restricted area.

Also, we merged the point cloud on COLMAP with 3fps and 7fps, and the result is shown as Figure 12 and Figure 13 respectively. Figure 13 shows that the result of 7fps is not as good as 3fps. Although

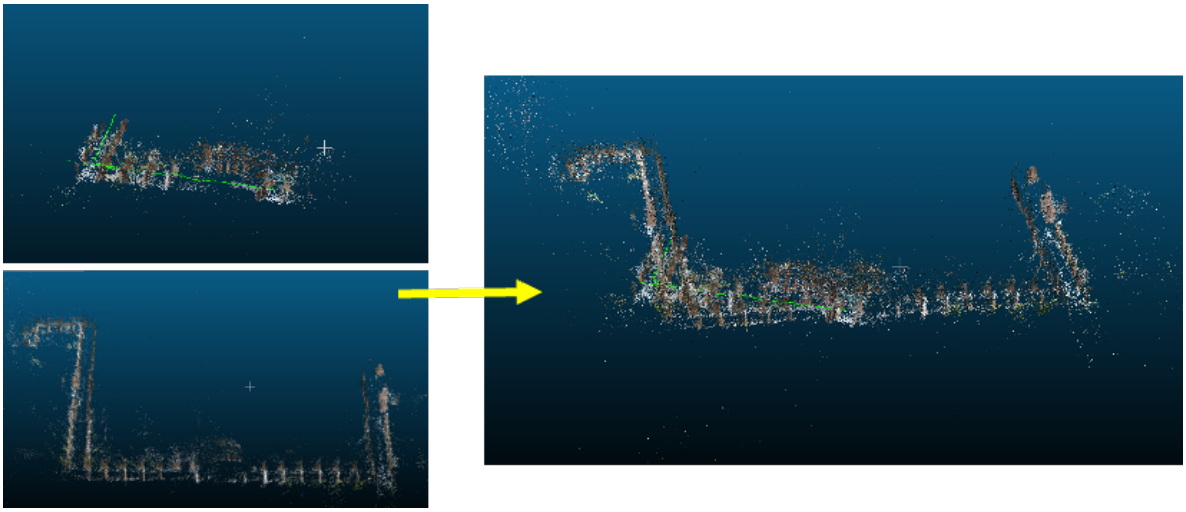

Figure 15: Visualized merge result.

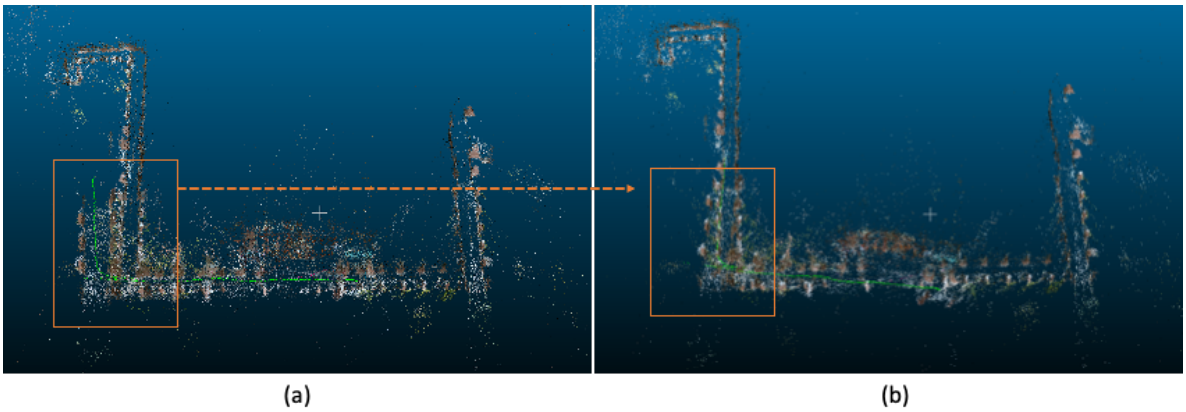

Figure 16: Merge result:(a) Manually select four points, (b) ICP after four point transform.

the two frames are closer to each other, there are more branches in Figure 13 instead.

For the maplab part, we tried to build it in Ubuntu 18.04, but we encountered some problems while we made this project. The specific problem will be discussed in part 5.

When inputting segmented images into ORB-SLAM, we were able to run with individual images successfully, but we encountered difficulties when merging multiple segments. The issue appeared to be related to the failure to call the global bundle adjustment (BA) during execution, leading to the restart of trajectory calculation when switching segments. Local matching was successfully called, but the global BA was not utilized.

5 Encountered Problem

We have encountered several challenges in our work, including the convergence of topic selection, the transformation of datasets, and the integration of different frameworks. The latter challenge has resulted in numerous environmental and system issues. In the COLMAP part, we built the model on the server, and then our memory size was full so we could not build the whole outside part of the library as there were a lot of RGB frames, so we only built the inside part. Figure 12 and Figure 13 both show that the lower right side has a branch.

When we tried to build the maplab environment in Ubuntu 18.04, we encountered that we could not run the ROVIOLI part at the beginning. It says that there are no flags about IMU parameters and camera calibration, and this problem arises from the version we built being the master branch, but the

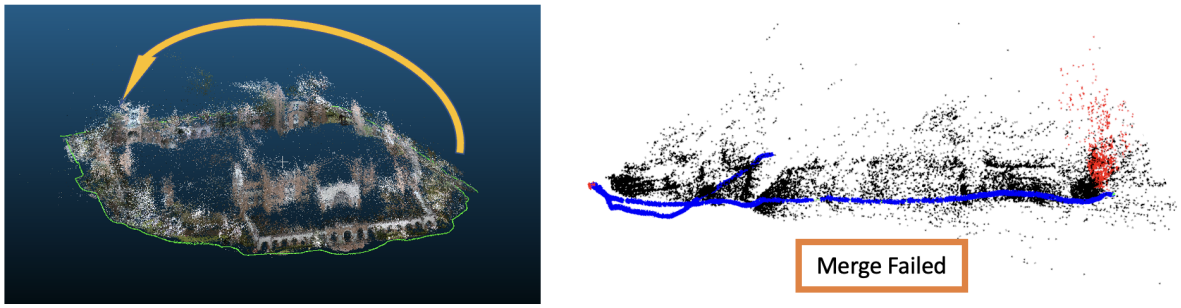

Figure 17: Using ORB-SLAM3 as a map-merging method.

official release of this version cannot build the map. We also tried other branches they released before but we got the same error as the beginning. Finally, we excluded building the map with maplab.

While we were able to successfully run ORB-SLAM in MP4 format, we encountered a problem when attempting to output the point cloud in this format. Specifically, we encountered a "core dump" issue, which may be caused by a memory leak in certain procedures. Besides, at multi-session part, the global BA was not utilized may due to the error from calling API.

6 Discussion

Through our experiment, we found that in scenarios where features are repeated, such as corridors or buildings with similar architectural styles, it can be difficult to accurately splice together maps. This is because the algorithms used to merge the maps may not be able to distinguish between similar features and may end up incorrectly combining them. In contrast, maps that have a greater variety of features and a clear sequence of images are generally easier to stitch together. However, even in scenarios where map splicing is more challenging, ORB-SLAM can still effectively reconstruct the path trajectory using individual images. This is because ORB-SLAM uses local feature matching and bundle adjustment to optimize the trajectory, enabling it to handle a wide range of conditions and scenarios.

To continue improving our mapping techniques, we are working on using the multi-session mapping feature in maplab. We are also exploring the possibility of using ORB-SLAM to generate dense point clouds as output. In addition, we are experimenting with applying deep learning methods to merge point clouds.

References

[AFS+11] Sameer Agarwal, Yasutaka Furukawa, Noah Snavely, Ian Simon, Brian Curless, Steven M. Seitz, and Rick Szeliski. Building rome in a day. *Communications of the ACM*, 54:105–112, 2011.

[BGS+16] Mathias Bürki, Igor Gilitschenski, Elena Stumm, Roland Siegwart, and Juan Nieto. Appearance-based landmark selection for efficient long-term visual localization. In *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, page 4137–4143. IEEE Press, 2016.

[BM92] Paul Besl and H.D. McKay. A method for registration of 3-d shapes. ieee trans pattern anal mach intell. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 14:239–256, 03 1992.

[CEG+21] Carlos Campos, Richard Elvira, Juan J. Gómez, José M. M. Montiel, and Juan D. Tardós. ORB-SLAM3: An accurate open-source library for visual, visual-inertial and multi-map SLAM. *IEEE Transactions on Robotics*, 37(6):1874–1890, 2021.

[CM92] Yang Chen and Gérard Medioni. Object modeling by registration of multiple range images. *Image Vision Comput.*, 10:145–155, 01 1992.

[CN13] Winston Churchill and Paul Newman. Experience-based navigation for long-term localisation. *International Journal of Robotics Research*, 32:1645–1661, 12 2013.

[FCSS10] Yasutaka Furukawa, Brian Curless, Steven Seitz, and Richard Szeliski. Towards internet-scale multi-view stereo. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 2010, pages 1434–1441, 06 2010.

[GGC+21] Carsten Griwodz, Simone Gasparini, Lilian Calvet, Pierre Gurdjos, Fabien Castan, Benoit Maujean, Gregoire De Lillo, and Yann Lanthony. Alicevision Meshroom: An open-source 3D reconstruction pipeline. In *Proceedings of the 12th ACM Multimedia Systems Conference - MMSys '21*. ACM Press, 2021.

[MAMT15] Raúl Mur-Artal, J. M. M. Montiel, and Juan D. Tardós. ORB-SLAM: a versatile and accurate monocular SLAM system. *IEEE Transactions on Robotics*, 31(5):1147–1163, 2015.

[NLD11] Richard Newcombe, Steven Lovegrove, and Andrew Davison. Dtam: Dense tracking and mapping in real-time. In *IEEE International Conference on Computer Vision*, pages 2320–2327, 11 2011.

[SDF+18] T. Schneider, M. T. Dymczyk, M. Fehr, K. Egger, S. Lynen, I. Gilitschenski, and R. Siegwart. maplab: An open framework for research in visual-inertial mapping and localization. *IEEE Robotics and Automation Letters*, 2018.

[SF16] Johannes Lutz Schönberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[SZPF16] Johannes Lutz Schönberger, Enliang Zheng, Marc Pollefeys, and Jan-Michael Frahm. Pixelwise view selection for unstructured multi-view stereo. In *European Conference on Computer Vision (ECCV)*, 2016.

[TMHF00] B. Triggs, Philip Mclauchlan, R. Hartley, and Andrew Fitzgibbon. Bundle adjustment - a modern synthesis. *ICCV '99 Proceedings of the International Workshop on Vision Algorithms: Theory and Practice*, pages 198–372, 01 2000.

[YFKGH20] Shuien Yu, Chunyun Fu, Amirali K Gostar, and Minghui Hu. A review on map-merging methods for typical map types in multiple-ground-robot slam solutions. *Sensors*, 20:6988, 12 2020.