Multi-robot collaborative SLAM and scene reconstruction based on RGB-D camera

Tianyun Ma
Department of Automation
Tsinghua University
Beijing, China
mty16@mails.tsinghua.edu.cn

Tao Zhang
Department of Automation
Tsinghua University
Beijing, China
taozhang@tsinghua.edu.cn

Shaopeng Li
Department of Automation
Tsinghua University
Beijing, China
sp-li16@mails.tsinghua.edu.cn

***Abstract*—Single-robot visual SLAM has problems such as slow mapping speed. This study proposed a multi-robot collaborative SLAM and scene reconstruction system based on an RGB-D camera. The system adopts a centralized structure. While several client robots collect RGB-D image data respectively and transmit the data to the server, the server runs a stand-alone visual SLAM system based on ORB-SLAM2 for each client, and stores the real-time mapping data in the map manager. At the same time, it detects whether the maps in the map managers meet the fusion conditions at a certain frame rate, and uses the map fusion algorithm to fuse when the conditions are met. In order to solve the problem that ORB-SLAM2 can only do semi-dense mapping, the system uses the 7-DoF poses of each client output by collaborative SLAM to reconstruct the dense map. We evaluated the performance of the system through simulations. Compared with the single-robot system, the collaborative SLAM in this system has a significant improvement in speed and can maintain high accuracy. We verified the effectiveness of the scene reconstruction algorithm through the scene reconstruction simulation and further proved that high-precision camera poses can be obtained in collaborative SLAM through the reconstruction results.**

***Index Terms*—RGB-D camera, collaborative SLAM, scene reconstruction, multi-robot**

I. Introduction

Due to the rich information, low price, simple configuration, and other characteristics of visual sensors, visual SLAM has increasingly drawn the attention and many mature visual SLAM frameworks have been proposed. A classic visual SLAM framework consists of five modules, which are visual sensor, visual odometry, back-end optimization, loop closure detection, and mapping [1]. One drawback of this framework is that in some scenes with long trace and deficient loop closure, the accumulated error is difficult to be eliminated, so it will greatly reduce the accuracy of mapping. It is an effective idea to divide the SLAM task into several parts and complete it by multi clients.

A single visual SLAM platform is difficult to meet the needs of scene data collection and mapping in many application scenarios, while the multi-robot system has more advantages in task efficiency, fault tolerance, reconfigurability, and hardware cost. The collaborative visual SLAM can not only shorten the time of mapping but also improve the mapping accuracy.

This study proposed a multi-robot collaborative SLAM and scene reconstruction system based on an RGB-D camera. The system collects RGB images and depth images through an RGB-D camera in the client and transmits the image data to the server directly. The data transmission between client and server is based on ROS (Robot Operating System). The server creates a visual SLAM sub-system for each client based on ORB-SLAM2 and transmits the image data package to the corresponding sub-system according to the client ID in the package for visual SLAM. At the same time, the server creates a map manager for each client which manages the map data of the client and checks whether the map meets the fusion conditions through additional threads. The maps that meet the conditions are fused and the fusion map is returned to the corresponding map managers respectively. At the end of collaborative SLAM, the server outputs each client's 7-DoF camera poses and integrates the clients' camera poses for dense scene reconstruction. We carried out the simulations of collaborative SLAM and scene reconstruction using the well-known TUM dataset.

The rest of the paper is organized as follows: Section II presents related works. In section III, we introduce the system topology structure, data processing flow, and communication mode and discuss the client SLAM framework, then we introduce the map fusion algorithm and scene reconstruction algorithm respectively. Section IV shows the simulations of collaborative SLAM and scene reconstruction using the sequences of the TUM dataset, the simulation results are evaluated. Finally, we draw a conclusion in Section V.

II. RELATED WORKS

The Related works are introduced from the multi-robot collaborative SLAM and scene reconstruction.

A. Multi-robot Collaborative SLAM

A few works have achieved collaborative SLAM using ground robots or UAVs, most of which are server-client architectures. [2] proposed a collaborative SLAM system for UAVs based on a monocular camera. Each UAV independently runs the visual odometry, carries out image feature extraction and initial pose estimation of image frames, and transmits the feature and the poses to the ground station. The ground station first optimizes the poses and selects the keyframes, then detects the overlap and loop closure through an independent scene recognition module, and further optimizes the poses by local BA (bundle adjustment) at the same time as detection. Finally, map fusion and loop correction are carried out. Although the visual odometry and scene recognition are modularized in this system, only the initial optimized pose is used in the scene recognition module, which leads to low accuracy of fusion and loop calibration in more complex scenes, and it is easy to occur the map drift. [3] proposed a multi-robot collaborative SLAM framework based on a pose graph, which can use the existing single-robot SLAM algorithm as its client's SLAM system. [4] proposed a centralized multi-robot collaborative SLAM system based on ORB-SLAM2. The system uses an efficient data transmission method. After the image features are extracted, the client encodes them and transmitted to the server, which greatly reduces the bandwidth utilization and improves the real-time performance of SLAM. However, the system only supports the stereo vision.

B. Collaborative Scene Reconstruction

Collaborative scene reconstruction pays more attention to the use of lightweight clients for data acquisition and requires reasonable task allocation to achieve the best utilization of clients. [5] put forward a concept of collaborative scene reconstruction combining multi-robot collaborative visual SLAM and SFM (Structure From Motion). Deploying SLAM client on mobile phones, Google glasses, and other devices to perform collaborative SLAM, and using a remote server to collect data for scene reconstruction, so as to reconstruct the scene quickly, which is of great significance in AR, 3D navigation and other fields. The method of collaborative scene reconstruction using a mobile camera for image acquisition and 3D dense point cloud splicing in the cloud server has been proposed in [6], [7], and [8]. [9] proposed a task assignment method based on OMT (Optima Mass Transport) which can plan the path dynamically for each robot, so that the robot can perform efficient scene reconstruction with the minimum overlap rate.

The accuracy of collaborative scene reconstruction depends on the accuracy of pose estimation. At present, the mature visual SLAM framework uses graph optimization, loop detection, and other methods to improve the accuracy of pose estimation. Therefore, using pose obtained through SLAM for scene reconstruction is an effective method to improve the accuracy of scene reconstruction. [10] proposed a scene reconstruction method for UAV based on visual SLAM. This method effectively improves the accuracy of pose estimation and the quality of scene reconstruction.

III. SYSTEM OVERVIEW

Figure 1 shows the topology structure of the system. Figure 2 shows the flow chart of the system. The server needs to specify the number of clients when initializing and create a sub-system based on ORB-SLAM2 for each client. The RGB-D image sequence is first processed by tracking thread and local mapping thread, and then the system transmits the generated keyframes to loop closure detection thread and map fusion thread respectively. The server performs global BA optimization after loop correction or map fusion. After collaborative SLAM, the system output the camera poses of all clients for scene reconstruction.

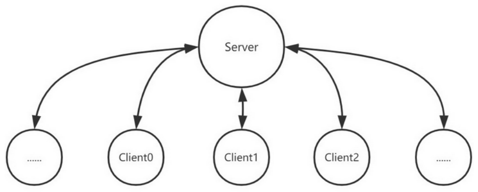

Fig. 1. System topology structure.

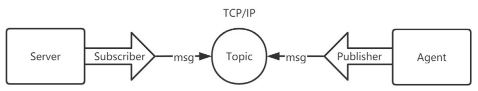

Fig. 2. Communication mode.

The system uses the topic-message mode of ROS for server-client communication, as shown in Figure 3, and the communication network is based on the TCP/IP protocol.

After collecting image data, the client encapsulates the data with the ROS message and then publishes the message to the topic. The server first initializes the map manager for each client, then establishes a message receiver for each client and subscribes messages from corresponding topics. Every time the receiver receives a message packet, it calls a callback function. In the callback function, the server extracts the RGB and depth image data, timestamps, and client ID from the message, and starts the tracking thread. Table I shows the types and meanings of communication messages between server and client.

TABLE I
MESSAGE FORMAT OF COMMUNICATION BETWEEN SERVER AND CLIENT

Message type	**Meaning**
Sensor_msgs/Image rgb	RGB image data
Sensor_msgs/Image depth	Depth image data
int16 clientId	Client ID
float64 timestamps	Timestamp of image data

A. Client SLAM Framework

The client SLAM framework is based on ORB-SLAM2 [11]. For each client, the sub-system runs three threads for visual SLAM, which are tracking, local mapping, and loop closure detection. The three threads implement parallel processing through the mechanism of the mutex lock. Based on the RGB-D sensor, the three threads are introduced as follows:

The tracking thread first preprocesses the RGB image and depth image input to the system. The RGB image is transformed into a gray image, and the image frame is constructed by the gray image and the depth image. In the construction of the image frame, the thread first extracts the

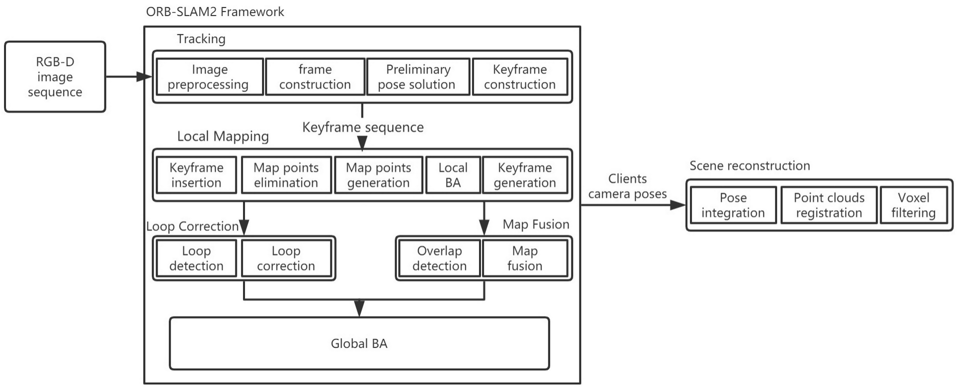

Fig. 3. System flow chart.

ORB features, then solves the camera pose and carries out local tracking. After the pose solution is completed, the thread selects keyframes from the frames and transmits them to the local mapping thread.

The local mapping thread maintains a covisible relationship between keyframes through map points which exists in the keyframes that can observe the same map points, and it optimizes the poses and the associated map points of the covisible keyframes using local BA. Specifically, the thread first places the keyframes received from the tracking thread in the queue, then calculates the BoW vector for each keyframe, after that, it adds constraints to the associated map points of the current keyframe, and finally adds the current keyframe to the existing covisibility graph. The added map points need to be filtered, and the map points that do not meet the conditions will be deleted. Then the thread uses the covisibility relationship between the current keyframe and its covisible keyframes to generate new map points by triangulation and adds them to the map. After that, the thread takes the current keyframe and its covisible keyframes, as well as the observed map points as the overall optimization target and uses the g2o library [12] to carry out local BA. Finally, redundant keyframes are removed from the covisibility graph.

The loop closure detection thread receives the keyframes from the local mapping thread. First, the thread performs loop closure detection, obtains the candidate loop correcting keyframes, then calculates the sim3 poses of all the candidate keyframes and selects the best keyframes for correction. Finally, it runs a global BA thread to optimize the global pose graph.

B. Map Fusion Algorithm

The map fusion thread of the server runs after the local mapping thread. The algorithm flow is divided into three parts: overlap detection, relative sim3 pose transformation, and map fusion, as shown in Figure 4.

Next, we will introduce the algorithm for each step.

1) Overlap Detection: The thread first obtains the client ID of the current keyframe and its covisible keyframes, then it locks the current keyframe to prevent other threads from deleting the keyframe. After that, the thread uses the word-bag technique to calculate the BoW score of the current keyframe and its covisible keyframes, and uses the lowest score as the threshold value to select the candidate matching keyframe when detecting overlap. Then, according to the threshold, the thread selects the candidate keyframes from the keyframe database of other clients. In order to reduce the mismatching rate of detection, it is necessary to detect the continuity of candidate keyframes. The method is to judge whether there is intersection among the sets that consist of candidate keyframe and its covisible keyframes. If there is intersection between current set and the former set, we say that this two candidate keyframes are continuous. The last keyframe of three continuous keyframes is selected and added to the final candidate keyframe queue.

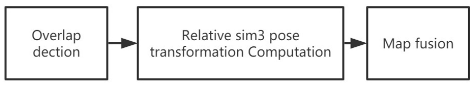

Fig. 4. Fusion algorithm flow.

2) Relative Sim3 Pose Transformation Computation: The thread calculates the sim3 pose of the candidate keyframes in this step. First, the thread matches each candidate keyframe with the target keyframe using the word-bag technique. If the map points matching the candidate keyframe and the target keyframe meet the requirements, the thread initializes the sim3 pose solver according to the map points, otherwise, it removes the candidate keyframe. Then, the thread uses RANSAC to solve the sim3 pose transformation matrix. If the sim3 pose transformation matrix is not found after the maximum number of iterations, it removes the keyframes. For the candidate keyframes solving the sim3 pose transformation matrix, more matching map points are searched again and the pose is optimized by map points. Finally, the thread takes out all the covisible frames and their map points of the candidate keyframes for the sim3 pose transformation matrix and projects with the target keyframes for matching. If the matching conditions are met, the final confirmation can be made for map fusion.

3) Map Fusion: In the map fusion step, the thread first obtains the client ID of the keyframes to be fused and extracts the map data from the corresponding client map managers. Then it uses the propagation method to calibrate the poses of all keyframes in the map managers and recalculates the coordinates of map points associated with keyframes. After that, it connects the common views of both sides according to the covisible relationship, and fuses the map points. After fusion, the system uses a global BA thread to optimize the whole poses of the graph. At this time, the two maps have been fused, and the map managers of both sides are updated with the fused map data.

C. Scene Reconstruction Algorithm

After the collaborative vision SLAM is completed, the system outputs each client's 7-DoF poses in the SLAM process to files respectively. Each line is the pose of a certain image frame. Its format is shown as

$$[ID\ of\ Frame]\ x\ y\ z\ qx\ qy\ qz\ qw$$

Where **[ID of frame]** is the timestamp of each frame, **(x y z)** and **(qx qy qz qw)** are the 3-D coordinates and quaternion digital positions of the camera in the current pose.

Because the poses used for scene reconstruction can be disordered, the system integrates the client poses, then converts the corresponding RGB and depth images into dense points according to the timestamp of the pose in each line and splices the point cloud. In order to avoid excessively massive points in the point cloud, the system uses voxel filter to filter the point cloud, and finally outputs the result of scene reconstruction.

IV. EXPERIMENTS

A. Collaborative SLAM Simulation

In order to verify the performance of collaborative SLAM, we carried out two groups of collaborative SLAM simulations respectively using two clients and three clients. For each group of simulations, we discussed the impact of client map overlap rate on the speed and accuracy of SLAM and compared the performance of the single-robot SLAM Based on ORB-SLAM2 and the collaborative SLAM.

We used the *rgbd dataset freiburg2 pioneer* sequence in the TUM dataset published by Munich University of Technology as a simulation sequence, which includes 2544 RGB images and 2531 depth images. In the simulation, we primarily intercepted the first 1800 RGB images and their matched depth images.

The client map overlap rate is defined as the ratio of the number of RGB images overlapped by two clients to the total number of RGB images of one client in collaborative SLAM. In the two clients simulation, the image sequences of SLAM performed by the two clients are $[\mathbf{1}, \mathbf{900}+\mathbf{1}/\mathbf{2}\boldsymbol{\delta}]$ and $[\mathbf{900}-\mathbf{1}/\mathbf{2}\boldsymbol{\delta}, \mathbf{1800}]$, where δ is the number of overlapped images, the map overlap rate is

$$\eta = \frac{\delta}{900 + \frac{1}{2}\delta} \times 100\% \tag{1}$$

In the three clients simulation, the image sequences of SLAM are $[\mathbf{1}, \mathbf{600}+\mathbf{2}/\mathbf{3}\boldsymbol{\delta}]$, $[\mathbf{601}-\mathbf{1}/\mathbf{3}\boldsymbol{\delta}, \mathbf{1200}+\mathbf{1}/\mathbf{3}\boldsymbol{\delta}]$ and $[\mathbf{1201}-\mathbf{2}/\mathbf{3}\boldsymbol{\delta}, \mathbf{1800}]$ respectively, and the map overlap rate is

$$\eta = \frac{\delta}{600 + \frac{2}{3}\delta} \times 100\% \tag{2}$$

Figure 5 shows the APE (absolute trajectory error) of global poses relative to the ground truth. The upper is three clients collaborative SLAM with a 15% overlap rate and the lower is single-robot SLAM based on ORB-SLAM2. It shows that the trajectory error of collaborative SLAM at fusion point A and fusion point B is lower than that of single-robot SLAM.

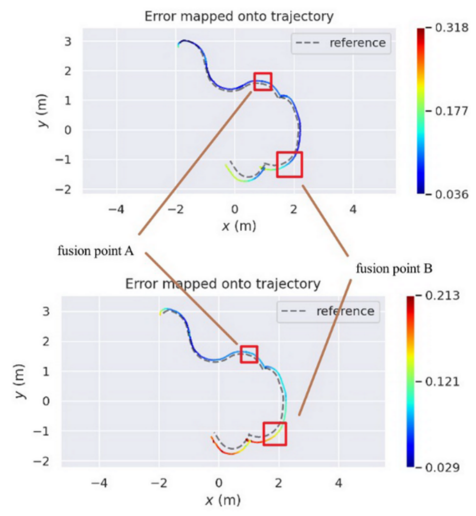

Fig. 5. APE of global poses relative to the ground truth. The upper is three clients collaborative SLAM with a 15% overlap rate and the lower is single-robot SLAM based on ORB-SLAM2.

Figure 6 (a), (b), (c) respectively shows the motion trajectory of the camera in the X-Y coordinate system, the translation motion in the X, Y and Z directions and the rotation motion of the camera of three clients' collaborative SLAM under 5% overlap rate. We can see that at the fusion point, the clients' camera trajectories keep a smooth joint.

Table II shows the performance comparison of single-robot SLAM based on ORB-SLAM2 and collaborative SLAM with $\eta = 0.5\%$, $\eta = 2.5\%$, $\eta = 5\%$ and $\eta = 10\%$ map overlap rate using two clients and three clients respectively. We compare the APE and the SLAM time. As can be seen from the table, collaborative SLAM is significantly faster than single-robot SLAM. For collaborative SLAM with the same number of clients, the SLAM time increases gradually with the increase of overlap rate, the reason is that with the increase of map overlap rate, the total length of map track increases, and the number of fusion candidate keyframes increases during overlap detection.

In the accuracy comparison, it is shown that the APE of collaborative SLAM using two or three clients compared with

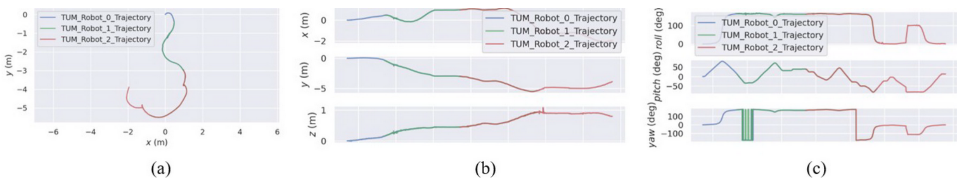

Fig. 6. Three clients' collaborative SLAM under 5% overlap rate (a) the motion trajectory of the camera in the X-Y coordinate system (b) the translation motion of the camera in the X, Y and Z directions (c) the rotation motion of the camera.

TABLE II
PERFORMANCE COMPARISON OF SINGLE-ROBOT SLAM BASED ON ORB-SLAM2 AND COLLABORATIVE SLAM WITH $\eta = 0.5\%$, $\eta = 2.5\%$, $\eta = 5\%$ AND $\eta = 10\%$ MAP OVERLAP RATE USING TWO CLIENTS AND THREE CLIENTS RESPECTIVELY

Simulation types	**Map overlap rate**	**SLAM time(s)**	**APE compared with ground truth**				
			MAX(m)	***MIN(m)***	***MEAN(m)***	***RMSE(m)***	***STD(m)***
two clients	0.5%	**182.991**	**0.272155**	0.025486	0.107329	0.115215	**0.041892**
	2.5%	183.954	0.287261	**0.015851**	0.104600	0.113892	0.045058
	5%	208.346	0.419079	0.023694	0.099872	0.108340	0.046197
	10%	216.230	0.430514	0.018811	**0.093263**	**0.106102**	0.050592
three clients	0.5%	192.963	0.280589	0.023632	0.111820	0.117270	0.042136
	2.5%	196.819	0.282969	0.016776	0.109107	0.116530	0.043377
	5%	210.618	0.302456	0.021313	0.107726	0.117106	0.044203
	10%	226.272	0.359229	0.024495	0.104230	0.113112	0.044530
single-robot	-	254.004	0.437231	0.027386	0.106989	0.117485	0.048540

ground truth is slightly lower than that of single-robot SLAM, which proves that collaborative SLAM has good accuracy compared with single-robot SLAM. For collaborative SLAM with the same number of clients, the increase of overlap rate leads to the increase of APE maximum error and standard deviation, but the decrease of average error and mean square error, which shows that the increase of map overlap rate increases the local error of trajectory and the overall error fluctuation, but the trajectory fits the real value better. This is because the increase of map overlap rate enlarges the solution space of the global BA optimization at the end of map fusion, so as to get a better combination of optimization, but the increase of map overlap rate is easy to cause over-optimization, resulting in the increase of local error and the increase of global error fluctuation.

It can be seen from the above simulations that the collaborative SLAM system greatly improves the speed of SLAM. And it keeps the smooth connection of the trajectory at the fusion point and can reduce the global trajectory error, thus improve the accuracy of SLAM. Moreover, the performance of collaborative SLAM is related to the overlap rate. The higher the overlap rate is, the slower the speed of SLAM is. Although increasing the overlap rate can reduce the root mean square error of the trajectory and make the trajectory fit the real value better, it will lead to over-optimization and increase the local error and the global error fluctuation.

B. Scene Reconstruction Simulation

In the scene reconstruction simulation, we choose the *rgbd dataset freiburg1 desk* sequence of the TUM dataset, which contains 613 RGB images and 595 depth images. Figure 7 shows the camera keyframe trajectory with a 5% overlap rate using two clients. Figure 8 shows the comparison between the point cloud model of collaborative SLAM and scene reconstruction with 5% map overlap rate using three clients, and the point cloud model of scene reconstruction using ground truth. From the qualitative analysis, the model of collaborative SLAM is more complete with smoother edge and more abundant details. We use the software *Cloudcompare* to calculate the *Hausdorff distances* [13] for quantitative comparison, which is shown as

$$d_H(X,Y) = max\left\{ \underset{x\in X}{sup}\,\underset{y\in Y}{inf}\, d(x,y), \underset{y\in Y}{sup}\,\underset{x\in X}{inf}\, d(x,y) \right\} \quad (3)$$

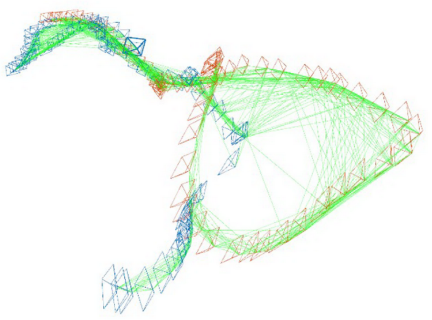

Fig. 7. Camera keyframe trace with a 5% map overlap rate using two clients.

Because *Hausdorff distance* is very sensitive to outlier points, we first match the bounding-box centers and match the scales using *Cloudcompare*. Then we compute the cloud-to-cloud *Hausdorff distance*, which we take the point cloud

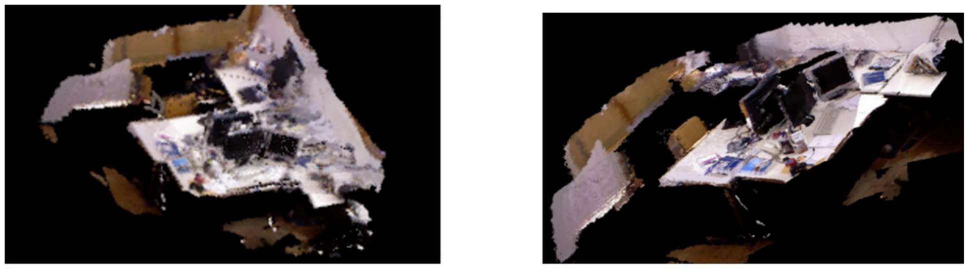
(a) (b)

Fig. 8. Comparison between the point cloud model of collaborative SLAM and scene reconstruction with 5% map overlap rate using three clients (a), and the point cloud model of scene reconstruction using ground truth (b).

TABLE III
CLOUD-TO-CLOUD HAUSDORFF DISTANCE WITH THE POINT CLOUD RECONSTRUCTED BY THE GROUND TRUTH AS A REFERENCE

Simulation types	Map overlap rate	Hausdorff distance(m)
two clients	0.5%	0.538863
	2.5%	0.534888
	5%	**0.531970**
	10%	0.535378
three clients	0.5%	0.536692
	2.5%	0.539653
	5%	**0.442427**
	10%	0.539545
single-robot	-	0.552168

reconstructed by the ground truth as a reference. The result is shown in Table III. It can be seen that the point cloud model obtained by scene reconstruction using the poses obtained by the collaborative SLAM is more accurate than that obtained by scene reconstruction using the poses obtained by single-robot SLAM based on ORB-SLAM2.

V. CONCLUSION

Aiming at the problems of slow mapping speed in single-robot SLAM and only semi-dense mapping in ORB-SLAM2, this study proposed a collaborative SLAM and scene reconstruction system based on RGB-D camera. The system adopts a centralized architecture, transplants the SLAM process to a server with stronger computing power, while only performs image acquisition in the clients, which makes the system more suitable for deployment in the lightweight collaborative SLAM system. We used the TUM dataset to simulate the collaborative SLAM and scene reconstruction respectively and proved that the collaborative SLAM has better speed and precision of mapping than the single-robot SLAM. Through the research on the overlap rate of the map, we found that the increase of overlap rate can improve the accuracy of mapping, but it will increase the fluctuation of the global trajectory. Finally, we verified the performance of the scene reconstruction algorithm through the scene reconstruction simulation and further proved that the camera poses with high accuracy can be obtained through collaborative SLAM. The next step is to optimize the data transmission between the clients and the server. Compression of the data transmitted by the server and the client can reduce the network bandwidth utilization and improve the speed of collaborative SLAM. At the same time, the scene reconstruction algorithm needs to be further optimized to improve the reconstruction accuracy, so that the reconstruction model can be further used for navigation and other functions.

REFERENCES

[1] A. Li, X. Ruan, J. Huang, X. Zhu, and F. Wang, "Review of vision-based simultaneous localization and mapping," in *2019 IEEE 3rd Information Technology, Networking, Electronic and Automation Control Conference (ITNEC)*, 2019, pp. 117–123.

[2] C. Forster, S. Lynen, L. Kneip, and D. Scaramuzza, "Collaborative monocular slam with multiple micro aerial vehicles," in *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2013, pp. 3962–3970.

[3] I. Deutsch, M. Liu, and R. Siegwart, "A framework for multi-robot pose graph slam," in *2016 IEEE International Conference on Real-time Computing and Robotics (RCAR)*, 2016, pp. 567–572.

[4] D. Van Opdenbosch and E. Steinbach, "Collaborative visual slam using compressed feature exchange," *IEEE Robotics and Automation Letters*, vol. 4, no. 1, pp. 57–64, 2019.

[5] P. Fleck, D. Schmalstieg, and C. Arth, "Visionary collaborative outdoor reconstruction using slam and sfm," in *2016 IEEE 9th Workshop on Software Engineering and Architectures for Realtime Interactive Systems (SEARIS)*, 2016, pp. 1–2.

[6] E. Nocerino, F. Poiesi, A. Locher, Y. T. Tefera, F. Remondino, P. Chippendale, and L. Van Gool, "3d reconstruction with a collaborative approach based on smartphones and a cloud-based server," *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. 42, no. W8, pp. 187–194, 2017.

[7] S. Franz, R. Irmler, and U. Rüppel, "Real-time collaborative reconstruction of digital building models with mobile devices," *Advanced Engineering Informatics*, vol. 38, pp. 569–580, 2018.

[8] F. Poiesi, A. Locher, P. Chippendale, E. Nocerino, F. Remondino, and L. Van Gool, "Cloud-based collaborative 3d reconstruction using smartphones," in *Proceedings of the 14th European Conference on Visual Media Production (CVMP 2017)*, 2017, pp. 1–9.

[9] S. Dong, K. Xu, Q. Zhou, A. Tagliasacchi, S. Xin, M. Nießner, and B. Chen, "Multi-robot collaborative dense scene reconstruction," *ACM Transactions on Graphics (TOG)*, vol. 38, no. 4, pp. 1–16, 2019.

[10] Z. Shang and Z. Shen, "Real-time 3d reconstruction on construction site using visual SLAM and UAV," *CoRR*, vol. abs/1712.07122, 2017. [Online]. Available: http://arxiv.org/abs/1712.07122

[11] R. Mur-Artal and J. D. Tardós, "Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras," *IEEE Transactions on Robotics*, vol. 33, no. 5, pp. 1255–1262, 2017.

[12] G. Grisetti, H. Strasdat, and K. Konolige, "g 2 o: A general framework for (hyper) graph optimization," 2011.

[13] R. T. Rockafellar and R. J.-B. Wets, *Variational analysis*. Springer Science & Business Media, 2009, vol. 317.