Proceedings of the 2014 IEEE
International Conference on Robotics and Biomimetics
December 5-10, 2014, Bali, Indonesia

Pose-graph Based 3D Map Fusion with Distributed Robot System

Haoyao Chen, Jin Zhong, Yanling Fu and Yunjiang Lou

***Abstract*—Map fusion is an essential and challenging issue in multi-robot simultaneous localization and mapping (SLAM), especially for 3D SLAM applications. In this paper, we propose an efficient pose-graph based approach to fuse several 3D maps built by distributed mobile robotic system. Several robots, each equipped with a Kinect sensor, cooperate to map the operated environment; each of them just scans a part of the environment. A target detection algorithm is presented to help the robot detect other robots during the map building. Key information will be transferred from the detected robot to the detecting robot to build a link between these two robots. And at last, G2O optimization library is applied on the connected pose-graphs to build an optimized map. Experiments are finally performed to demonstrate the effectiveness of the proposed approach.**

I. Introduction

SLAM is a challenging problem in robotics, because the robot often needs to build a map of the unknown environment and localize its pose w.r.t. the built map simultaneously. To acquire the map, a robot need to know its pose; while a robot wants to obtain its pose, the map information is required. This is well-known as a chicken-and-egg issue. Despite of the difficulties, great success has been achieved on this topic in the last two decades [1]-[4], and has promoted other research fields like autonomous navigation, path planning and obstacle avoidance, etc. Till now, many SLAM methods based on probabilistic theory [5] have been developed, including extend Kalman Filter SLAM [6], Graph based SLAM [7], [8] and FastSLAM [9], etc.

Developments in single robot SLAM have significantly promoted the research on multi-robot SLAM. Multi-robot SLAM gains superiority in efficiency, accuracy, robustness, etc. For example, multiple robots can concurrently perform SLAM task, and speed up the map construction especially in large-scale environments. Moreover, multiple robots can give us complementary perspective of the environment which is very useful to scale the map size and enrich the map information. In addition, multi-robot system is robust as it can still work even some of the robots is broken.

However, multi-robot SLAM problem is more challenging than single-robot SLAM as it needs the relative poses between the different robots and then fuses the maps built in each robot. Map fusion is critical in multi-robot SLAM because the to-be-fused sub-maps are built in distributed robots.

This paper aims to develop a new map fusion approach with a distributed robot system. Inspirited from RGBD-SLAM [10], the proposed approach is implemented in a similar way to deal with the single-robot SLAM problem. First, the robots independently build their own pose graph as well as store the sensed data; meanwhile, an object detection algorithm is implemented to detect other robots in the field of view (FOV). The detection algorithm does not need to attach any artificial landmarks (i.e. checkerboard) on the robots. Once a robot is detected, it is inquired to send its current sensor data (color and depth images captured by the RGBD camera) together with the current pose index to the detecting robot. The detecting robot matches the received data with the local sensor data to establish nodes' connection. The detection algorithm will provide a primary relative pose between the two robots to reduce the matching time. With the matching results, an accurate transformation between the robots' poses at detected time is calculated, and a new pose node will be added to the pose graph of the detecting robot. When the robots finish SLAM, all the pose graphs and sensor data will be transferred to the server for pose graph optimization [11]; a fused map is finally built by attaching colored point cloud to the pose graph. Experiments with two mobile robots were performed to verify the proposed approach. The proposed approach can be easily scaled to applications with more than two robots.

II. Related Work

For 3D SLAM applications, laser scanners, Time-of-Flight (ToF) cameras, and RGBD cameras are widely-used sensors. Compared to the laser and TOF camera, RGBD camera has the advantages of low cost, easy use, and rich data of both dense depth and color images. In many of the 3D mapping algorithms, the iteratively closest point (ICP) algorithm [12-13] is employed to implement data registration.

F. Endres et al. [10] proposed a novel visual SLAM algorithm, named RGBD-SLAM, which utilized Kinect for 3D mapping. Features extracted from the current color image were matched to previously extracted features. Combined with the synchronized depth image, the features and point clouds are registered to compute the transformations between two camera poses. A pose graph, including the poses along the motion trajectory and poses' uncertainty, can then be constructed. However, as the map keeps growing, the error of the pose graph will be accumulated if without loop closures.

B. Zhang et al. presented a method [14] using heterogenous

This work is supported by a grant from National Natural Science Foundation of China (Reference No. 61005063)

J. Zhong, H. Chen, Y. Fu and Y. Lou are with the Department of Mechanical Engineering and Automation and the Shenzhen Key Lab of Advanced Control and Automation at Harbin Institute of Technology Shenzhen Graduate School, Shenzhen, China. (Tel: 86-755-26033788; E-mail: hychen5@hitsz.edu.cn, zhongjin616@foxmail.com, 2644218275@qq.com, louyj@hitsz.edu.cn).

978-1-4799-7397-2/14/$31.00 © 2014 IEEE

sensors distributed in a multi-robot system to perform multi-robot mapping. Two robots build a two-dimensional occupancy map using FastSLAM separately; the robots' motions data and measurements were recorded at the same time. When mapping finished, the AMCL algorithm [15] was used to relocate one robot in the other robot's map based on the recorded information. However, the work only fused two maps with large overlaps.

B. Kim presents an extension to iSAM to facilitate online multi-robot mapping based on multiple pose graphs [4]. Their contribution is a relative formulation of the relationship between multiple pose graphs that avoids the initialization problem and leads to an efficient solution when compared to a completely global formulation. The relative pose graphs are optimized together to provide a globally consistent multi-robot solution.

In this work, we take the advantage of the RGB-D camera's rich information to perform both the target detection and map fusion. By transferring the sensor information between robots' loops can be established to help minimize the accumulated error and avoid mapping degradation.

III. MAP FUSION WITH DISTRIBUTED MOBILE ROBOTS

Figure 1 illustrates the flow chart of the proposed approach. The RGBD-SLAM algorithm is utilized to implement the single-robot SLAM in each robot. During the SLAM process, the target detection, integrated with Harr features and histogram matching, is proposed to detect other robots. Sensor and pose-graph information will be transferred to add a link-node for further map fusion, once a robot is detected. Finally, a powerful server is utilized for pose-graph optimization and map building.

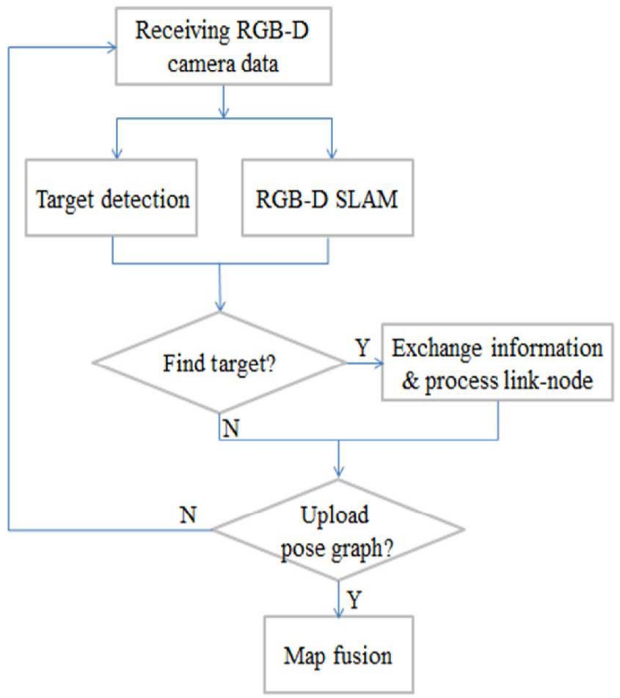

Fig.1.The flow chart of the proposed approach

A. Pose-graph based RGBD-SLAM

RGBD-SLAM utilizes a combined feature and point cloud matching scheme to build a pose-graph, and then applies graph optimization to realize full SLAM. Each or selected RGB image from the RGBD camera is processed to extract apparent features like SIFTs or SURFs. The features are compared with the ones extracted from previous images; once matched, an initial transformation between the two camera poses is calculated by using RANSAC. Based on the transformation, the point clouds obtained at two different poses are registered to obtain a more accurate transformation by using ICP method; it allows us to register dense point clouds in a common coordinate system. Each camera pose associated with its point clouds is called a node. As shown in Fig. 2, the calculated transformations between poses are defined as edges to connect the pose nodes into a pose graph, where the poses are indexed sequentially. The pose graph will be periodically optimized and updated by using g2o to reduce the SLAM uncertainty.

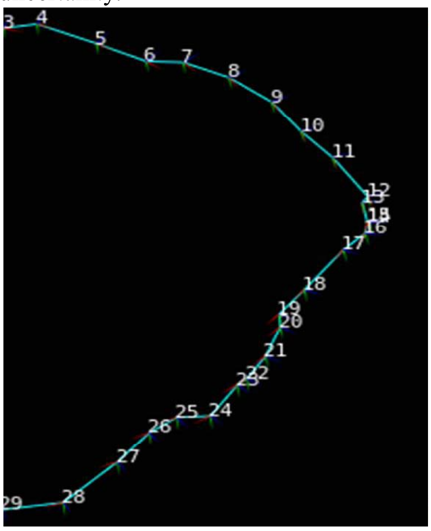

Fig.2.Illustration of a pose graph of RGBD-SLAM

B. Target detection

The relative poses between different robots are critical for multi-robot SLAM. Generally, it is preferred to calculate the relative pose when two robots rendezvous at some place. That means one robot needs to detect other robots when they appear in its FOV.

The existing target detection methods in multi-robot SLAM, always equipped artificial landmarks to improve the detection rate. In this paper, a new method without needing any landmarks is developed. As shown in Fig. 3, the Haar-like features of samples are first extracted to train a cascade AdaBoost classifier [16], [17]; the classifier is then used to judge whether there is a target in the FOV. In this paper, 319 positive samples and 1300 negative samples were used. However, sometimes the classifier can generate detection outlets; therefore the gray histogram matching method is integrated. The targets' histograms, denoted as His1, His2, …, Hisn, are matched to the template gray histogram, to obtain the similarities denoted as S1, S2, …,Sn. Finally, the maximum value in the set $\{S1, S2, \cdots, Sn\}$ is compared with a predefined threshold δ. If $MAX\{S1, S2, ..., Sn\} > \delta$, the target with the maximum similarity is the final target. If $MAX\{S1, S2, ..., Sn\} < \delta$ or the classifier detects no target, there is no target in the current FOV.

Figure 4 illustrates a detection example in our office. The method has also been verified in different illuminations and places. Besides providing target detection, a crude transformation between the detecting and detected robots is also obtained by using the depth information from the RGBD camera.

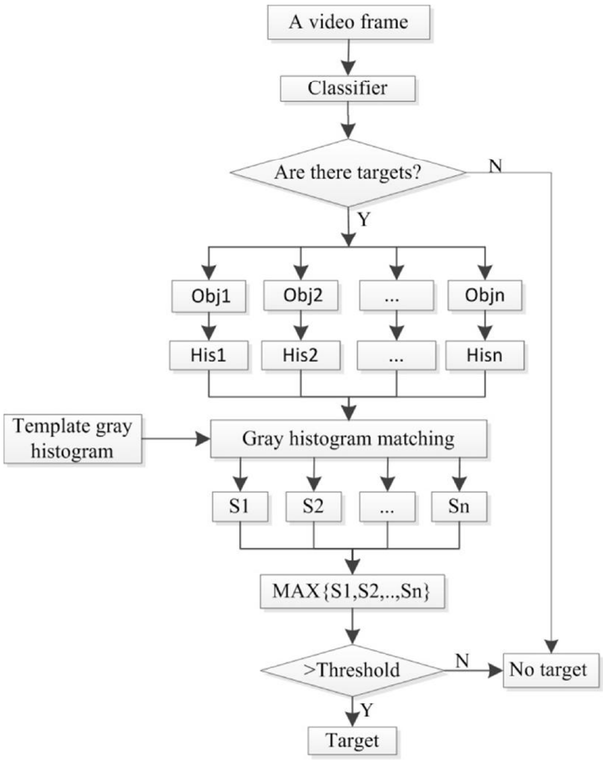

Fig.3.The flow chart of the target detection algorithm

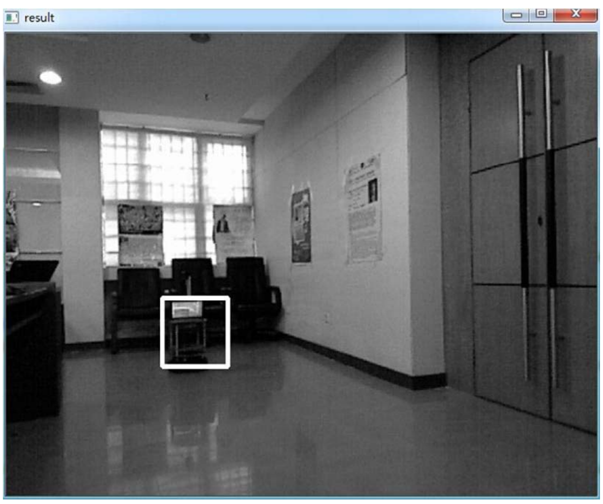

Fig.4.A robot is detected in one robot's field of view

C. Data Transfer

To compute the relative transformation between the two robots, key sensor data should be transferred from the detected robot to the detecting one. First, the detecting robot (*A*) sends a message to the detected robot (*B*); when receiving the signal, *B* sends back its latest few nodes' sensor data (color and depth images acquired by the RGB-D camera) together with the node indexes to *A*. Robot *A* compares the received data with that at its own nodes until matching one of them. With the matching result, a link-node will be constructed and added into the pose graph of robot *A*, which corresponds to the node in *B*'s pose graph. The link-node will be used as a bridge to fuse two robots' sub-maps. As shown in Fig. 5, robot1 detects robot2 at the pose of node 3. Robot 1 receives and compares the data sensed at node 3 of robot 2 with that of itself. Finally a new node, named link-node, is added into robot1's pose graph.

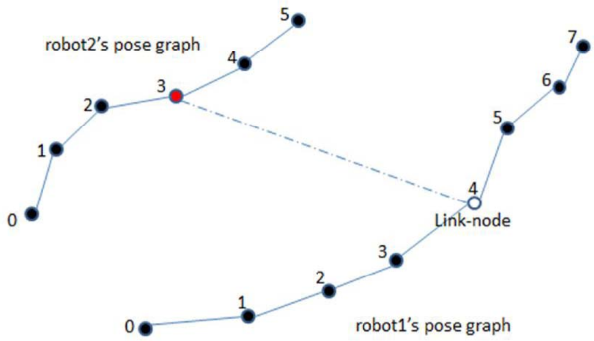

Fig.5. link-node in pose graph

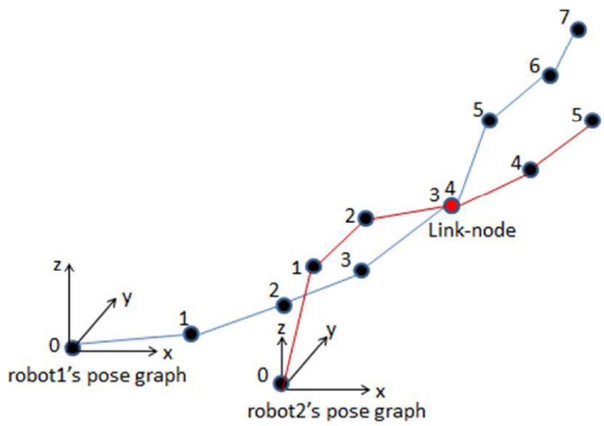

Fig. 6.Convert pose graph to another coordinate system

D. Map fusion

When every robot finishes the SLAM procedure, it uploads its pose-graph and map information to the server. Because point cloud transfer is bandwidth-consuming, the original depth images are transferred instead of the point clouds; the server will use these images to regenerate the point clouds. In the proposed approach, the robot with link-nodes will be transferred first, and then the server will signal the second robot to upload the map. The node in the second robot corresponding to the link-node should be sent first. As explained in section III. C, the special node is the same as the link-node in the pose graph of the detecting robot.

To fuse different robots' pose graphs, the second robot's pose graph needs transforming to the coordinate of the first robot, as shown in Fig.6. Initially, the node corresponding to the link-node is fixed as the coordinate origin of robot 2's pose graph, and all the others' pose coordinates of robot 2 are recalculated consequently. Further, with the link-node's coordinate w.r.t. robot 1, the recalculated pose graph of robot 2 can easily transformed to the coordinate system of robot 1. Denote the link-node's coordinates in graph1 as $(x_l, y_l, z_l, r_l, p_l, y_l)$ and the corresponding node in graph2 as $(x_c, y_c, z_c, r_c, p_c, y_c)$, where (x, y, z) is the translations and (r, p, y) are the Euler angles. Defining the transformation matrix from link-node based frame to the robot1 frame as *M*, then we have

$$M = \begin{bmatrix} R_l & T_l \\ 0 & 1 \end{bmatrix}, \tag{1}$$

where

$$R_l = \begin{bmatrix} \cos r_l & -\sin r_l & 0 \\ \sin r_l & \cos r_l & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos p_l & -\sin p_l \\ 0 & \sin p_l & \cos p_l \end{bmatrix} \cdot \begin{bmatrix} \cos y_l & 0 & -\sin y_l \\ 0 & 1 & 0 \\ \sin y_l & 0 & \cos y_l \end{bmatrix}, \tag{2}$$

$$T_l = (x_l, y_l, z_l)^T . \tag{3}$$

Denote the coordinates of the graph2's node *i* w.r.t. the robot2 frame as $(x_i^2, y_i^2, z_i^2, r_i^2, p_i^2, y_i^2)$, which can be reformed into a matrix as $\begin{bmatrix} R_i^2 & T_i^2 \\ 0 & 1 \end{bmatrix}$. R_i^2 and T_i^2 are similar to R_l and T_l respectively, but with the value of $(x_i^2, y_i^2, z_i^2, r_i^2, p_i^2, y_i^2)$ instead.

Then the matrix-formed coordinates of the node *i* w.r.t. the frame of the corresponding node can be calculated as

$$\begin{bmatrix} R_i^c & T_i^c \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} R_c & T_c \\ 0 & 1 \end{bmatrix}^{-1} \cdot \begin{bmatrix} R_i^2 & T_i^2 \\ 0 & 1 \end{bmatrix} \tag{4}$$

Finally, the matrix-formed coordinates of robot2's node *i* w.r.t. robot1's coordinate frame is given as

$$\begin{aligned} \begin{bmatrix} R_i^1 & T_i^1 \\ 0 & 1 \end{bmatrix} &= M \cdot \begin{bmatrix} R_i^c & T_i^c \\ 0 & 1 \end{bmatrix} \\ &= \begin{bmatrix} R_l & T_l \\ 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} R_c & T_c \\ 0 & 1 \end{bmatrix}^{-1} \cdot \begin{bmatrix} R_i^2 & T_i^2 \\ 0 & 1 \end{bmatrix} \end{aligned} \tag{5}$$

From Eq. (5), the Euler angles of robot2's nodes w.r.t. robot1's coordinate frame can be obtained by resolving the rotation matrix R_i^1. By transforming the coordinates of all nodes into one common frame, the g2o optimization algorithm then can be used to optimize the fused pose-graph.

With the link-nodes, it becomes easy to merge together the pose graphs of different robots. The edges, that connect the special node with other nodes in the detected robot's graph, generate connections to the ones that are connected to the link-node in the detecting robot; thus the relation between two distributed graphs is established. Furthermore, if link-nodes are established at both the beginning and end of the graph, loop closure can be performed to improve the quality of SLAM.

IV. EXPERIMENT

An experiment was performed in a room about 8m*15m to verify the effectiveness of the proposed approach. The robots communicate with each other through a Wifi network. The algorithm was executed on the Robot Operating System (ROS) Groovy version, which provided a convenient interprocess communication framework. As shown in Fig.7, two turtlebots were used in the experiment.

Fig.7.Robots used in this experiment

To make the fused map to form a loop, each robot is controlled manually to scan a complementary part of this room. Figures 8 to 12 illustrates the individual map and pose graph of each robot. As seen in Figs. 8 and 9, the map and pose graph built by the single-robot RGBD-SLAM didn't perform well due to lack of large loops.

Fig.8.The top-view of the map built by robot1

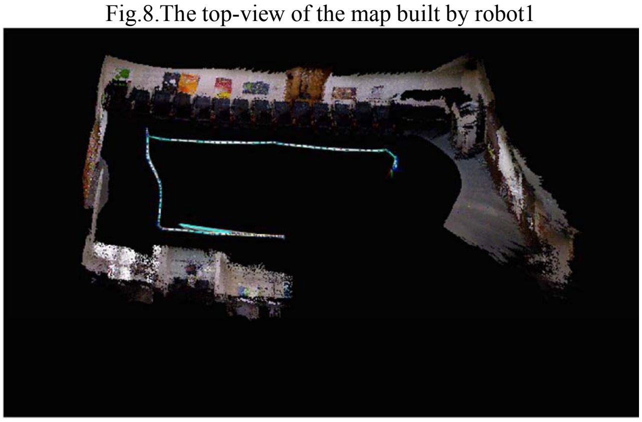

Fig.9.Another view of the map built by robot 1

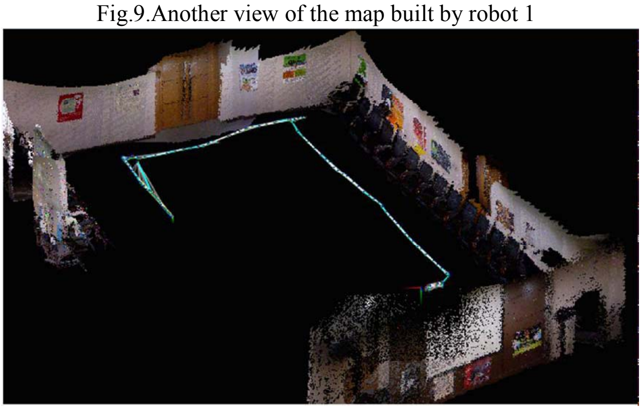

Fig.10.Zooming view of the map of robot1

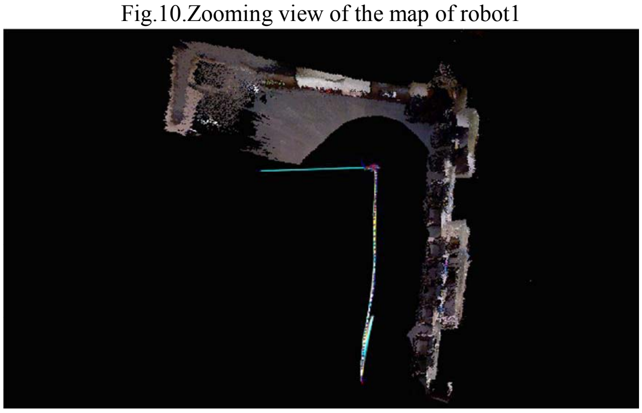

Fig.11 The top view of the map built by robot2

Fig.12Another view of map built by robot2

While the robots finished SLAM process, they uploaded their sub-maps and pose-graphs to the server in sequence as presented in Section III. C and D. The server converted one pose graph to the other's coordinate frame. The g2o method was used to optimize the fused pose-graph. And finally, 3D colored map was built by attached colored point clouds to the optimized pose graph. It is seen from Figs. 13 and 14 that the proposed approach performed well.

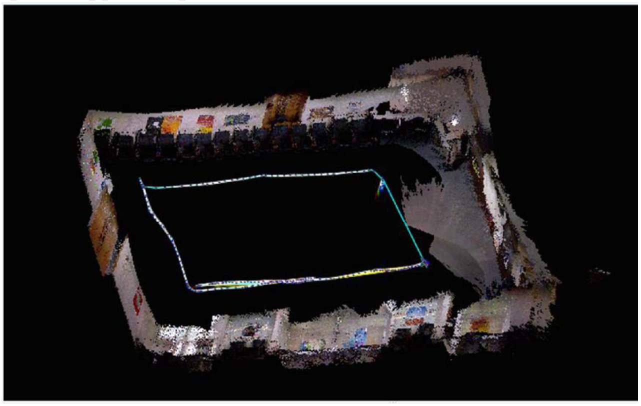

Fig.13 Map after fusion

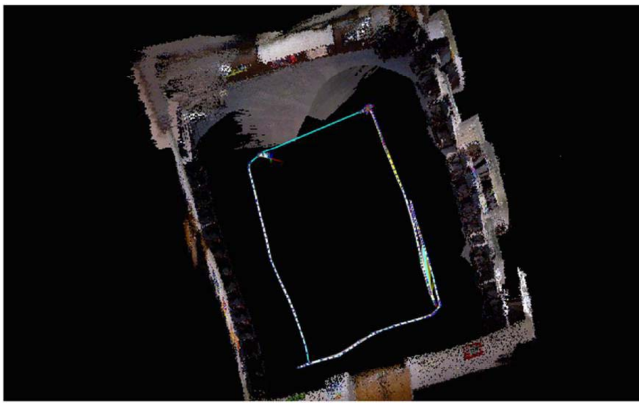

Fig.14 Another view of fusion map

V. CONCLUSION

In this paper, a novel approach is developed to address the problem of multi-robot 3D map fusion. Two robots independently execute single-robot SLAM algorithm to obtain their own pose graph and sub-maps of the environment. A new target detection algorithm is proposed to help the robots to detect each other and calculate the relative poses. Information will be transferred to the detecting robot from the detected one to construct a link-node in the detecting robot's pose graph. After finishing the single-robot SLAM, the data obtained during the SLAM will be transferred to the server for map fusion. Experimental results illustrate that the proposed approach performs well. Our future work includes the reduction of bandwidth burden, and the improvement of computation effectiveness of the SLAM and map fusion algorithms.

REFERENCES

[1] U. Frese, P. Larsson, and T. Duckett."A multilevel relaxation algorithmfor simultaneous localisation and mapping," *IEEE Trans. on Robotics*, 21(2):1-12, 2005.

[2] A. Nüchter, K. Lingemann, J. Hertzberg, and H. Surmann. "6DSLAM with approximate data association." In *Proc. of the 12th Intl. Conference on Advanced Robotics* (ICAR), pages 242–249, 2005.

[3] E. Olson, J. Leonard, and S. Teller. "Fast iterative optimization of posegraphs with poor initial estimates," In *Proc. of the IEEE Intl. Conf. onRobotics and Automation* (ICRA), pages 2262–2269, 2006.

[4] Kim, B., Kaess, M., Fletcher, L., Leonard, J., Bachrach, A., Roy, N., & Teller, S. Multiple relative pose graphs for robust cooperative mapping. *In Robotics and Automation (ICRA), 2010 IEEE International Conference on (pp. 3185-3192). IEEE.*

[5] S. Thrun, W. Burgard, and D. Fox, Probabilistic robotics, Cambridge MIT press, 2005.

[6] A. J. Davison, I. D. Reid, N. Molton, and O. Stasse, "MonoSLAM: Realtime single camera SLAM," *IEEE Trans. Pattern Analysis and Machine Intelligence*, 29(6), pp. 1052-1067, 2007.

[7] J. Folkesson, and H. Christensen, "Graphical SLAM-a self-correctingmap," in *IEEE Int. Conf. Robotics and Automation*. pp. 383-390,2004.

[8] S. Thrun, and M. Montemerlo, "The graph SLAM algorithm with applications to large-scale mapping of urban structures," *International Journal of Robotics Research*, vol. 25, no. 5-6, pp. 403-429,2006.

[9] G. Grisetti, C. Stachniss, and W. Burgard, "Improved techniques for grid mapping with rao-blackwellized particle filters," *IEEE Trans. Robotics and Automation*, 23(1):34-46, 2007.

[10] F. Endres, J. Hess, and W. Burgard, "An Evaluation of the RGB-D SLAM System," in *IEEE Int. Conf. Robotics and Automation*. pp. 1691-1696,2012.

[11] R. Kümmerle, G. Grisetti, H. Strasdat, K. Konolige, and W. Burgard."g2o: A general framework for graph optimization."In *Proc. of the IEEE Intl. Conf. on Robotics and Automation* (ICRA), 2011.

[12] A. Segal, D. Haehnel, and S. Thrun, "Generalized-icp," in *Proc. of Robotics: Science and Systems (RSS)*. pp. 26-27,2009.

[13] S. Rusinkiewicz and M. Levoy. "Efficient variants of the ICP algorithm,"In *Proc. of the Intl. Conf. on 3-D Digital Imaging and Modeling*, 2001.

[14] B. Zhang, J. Liu and H. Chen, "AMCL based Map Fusion For Multi-robot SLAM with Heterogenous Sensors," The *Proc. of the IEEE Inter. Conf. on Information and Automation*, pp. 820-825, 2013.

[15] D.Fox, "Adapting the sample size in particle filters through KLD-sampling," *The international Journal of robotics research,* 22(12):985-1003, 2003.

[16] R. Lienhart, J. Maydt, "An extened set of Haar-like features for rapid objection detection," *IEEE ICIP*, pp. 900~903, 2002.

[17] J. Friedman, T. Hastie, and R. Tibshirani, "Additive Logistic Regression: a Statistical View of Boosting," *The Annals of Statistics*, 38(2), 2000.