Proceeding of the IEEE
International Conference on Robotics and Biomimetics
Dali, China, December 2019

FC-SLAM: Federated Learning Enhanced Distributed Visual-LiDAR SLAM In Cloud Robotic System*

Zhaoran Li[1,2], Lujia Wang[1,†], Lingxin Jiang[1] and Cheng-Zhong Xu[3]

***Abstract*—SLAM has shown great values in many fields such as self-driving cars, virtual reality and robotic localization, etc. Cloud robot based collaborative SLAM can effectively improve the efficiency of mapping tasks. However, place recognition and matching accuracy among different robots can greatly affect the map fusion performance of the entire SLAM system. Therefore, this paper presents a learning architecture for cooperative SLAM named FC-SLAM, a distributed SLAM in cloud robotic systems by taking advantage of federated learning to enhance the performance of visual-LiDAR SLAM. Additionally, we propose a federated deep learning algorithm for feature extraction and dynamic vocabulary designation which works in real-time on cloud workstation. FC-SLAM can ensure real-time collaborative SLAM by keep the original images on robot side instead of sending them to the cloud server. We test our system on open datasets and in simulated environment. The results show that it has better feature extraction performance than SIFT and ORB under illumination and viewpoint changes. Besides, map fusion is conducted to generate a global map according to place matching relation of distributed robots.**

I. Introduction

Simultaneous Localization and Mapping (SLAM) techniques enable robots to continuously build maps and locate themselves at the same time in an unknown environment. SLAM was first proposed by Smith Self and Cheeseman et al. in 1987. In the early stage, SLAM was mostly implemented by sonar, single-line lidar and other sensors. Since 2000, with the development of computer vision, the use of camera and fusion with other sensors such as IMU has become a research hotspot. In addition, it has shown great values in many fields such as virtual reality, self-driving, robot localization and navigation and so on. C. Cadena [1] and J. Fuentes-Pacheco [2] et al. made a detailed review of the development and research of SLAM.

The front end of visual SLAM is concerned with camera motion between adjacent images, also known as visual odometry (VO). VO mainly includes feature detection, matching and pose estimation. It is worth noting that the quality of feature extraction and the accuracy of matching have important influence on the performance of pose estimation, localization and mapping in the whole SLAM system. Traditional feature extraction methods include ORB [3] and SURF [4], etc. In recent years, with the rise of deep learning, many studies have started to use convolutional neural network (CNN) to extract feature points and descriptors. These methods show better invariance, robustness and distinguish ability under the varying conditions of scale, rotation and illumination.

In order to improve the efficiency of robot mapping, researchers began to use multiple robots to complete the SLAM tasks. Considering of the global optimization of pose graph and map fusion, the recognition accuracy of the same place is important for multi-agent SLAM systems. After the concept of cloud robot was proposed, some researchers have started to apply cloud architecture into multi-agent SLAM system. However, a stable and scalable multi-agent SLAM system which is based on cloud has not been proposed yet.

In this paper, we propose a multi-agent SLAM system based on cloud robotic system which is called FC-SLAM. We implement FC-SLAM with three robots and a central cloud server to complete the task of building global map of an unknown environment. Our major contributions are:

1) A novel scalable federated learning framework is proposed to enhance distributed SLAM in cloud robotic systems.
2) A federated deep learning detector is proposed to enhance the robustness of feature extraction and the accuracy of feature matching.
3) A federated learning based place matching algorithm is proposed, and the dynamic dictionary is adopted to improve the accuracy rate of place matching.

In the rest of the paper, we discuss related work in section II, we describe our system and method in section III, and present the experiment results in section IV and with conclusions in section V.

II. Related Works

A. Laser SLAM and Visual SLAM

Laser-based SLAM has been well studied [5], [6], [7], and so far, there have been many stable and open source works [8], [9], [10], etc. These methods are mainly based on particle filtering, graph optimization or non-linear optimization. Visual SLAM can be divided into two branches: the direct method and feature-based method. The former calculates camera motion and builds maps of the surrounding environment according to the gray-scale value of all pixels [11]. Studies in recent years include DTAM [12], DSO [13] and so on. The latter extracts feature points from images and then matches them [14]. Based on this matching relation, camera motion estimation and map

[1] The authors are with the Cloud Computing Lab of Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences, Shenzhen 518055, China. zr.li, lj.wang1, lx.jiang@siat.ac.cn

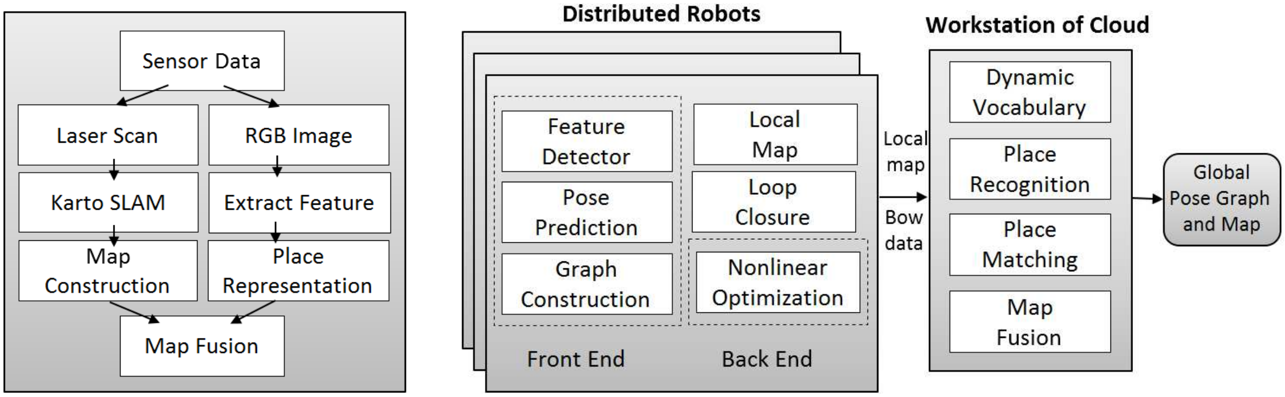
[2] University of Chinese Academy of Sciences, Beijing 100049, China.

[3] University of Macau, China. czxu@um.edu.mo

† Lujia Wang is the corresponding author. lj.wang1@siat.ac.cn

* This research is supported by the Shenzhen Science and Technology Innovation Commission (Grant Number JCYJ20170818153518789), the Guangdong Science and Technology Plan Guangdong-Hong Kong Cooperation Innovation Platform (Grant Number 2018B050502009) and the National Natural Science Foundation of China (Grant Number 61603376) awarded to Dr. Lujia Wang.

978-1-7281-6321-5/19//$31.00 ©2019 IEEE



(a) System Threads.

(b) System Modules

Fig. 1. FC-SLAM is a federated learning based distributed SLAM for cloud robotic systems. LiDAR and monocular camera are adopted for perception. Point cloud is mainly used by distributed robots to construct local maps. Visual image is adopted to extract features and match images from the same place. Finally, The system merges the local maps by using place matching relation.

construction are completed. State of the art studies include ORB-SLAM [15], etc. However, the feature-based method has a poor effect on circumstance where there is no obvious texture and feature missing, while the direct method is easily affected by illumination and requires a large amount of calculation.

B. Fearture Detectors

The traditional hand-craft feature point extraction method has been widely researched. FAST corner detector can detect obvious changes in local grayscale pixels. Last but not least, Oriented FAST and Rotated BRIEF (ORB) method [3] takes improved FAST corners and binary BRIEF descriptors, which is popular in SLAM systems due to its low computational effort but excellent performance in rotation and scaling.

More and more researches have been devoted to training deep neural networks for feature extraction. DeepDesc [16] is a piece of early typical work that uses a twin neural network to extract local feature descriptors. Subsequently, Patch based method is proposed. MatchNet [17] is a deep network based on patch with robust feature comparison which consists of two networks: feature network and metric network. SuperPoint [18] carries out convolution on the whole image, calculates the position of feature points and related descriptors in a forward propagation, and adopts self-supervised learning method to train the network.

Since federated learning was first proposed by Konečnỳ et al. [19] in 2016, researches have begun to apply federated learning framework to improve learning efficiency and protect privacy. There is also some work [20] that apply federated learning in robot system to realize motion planning tasks of multi-agent system. However, so far there has been no work that applies federated learning in a SLAM system.

C. Multi-agent SLAM and Cloud Robotic System

Using the SLAM system established by predecessors to construct a multi-agent collaborative SLAM system has been studied in [21], [22], [23]. These work has realized specific SLAM system but without well scalability. In addition, place recognition and matching are much more important for multi-agent SLAM system. Nevertheless, the above work has not come up with a better method to improve the accuracy rate of place matching. As the concept of cloud robot [24] was put forward by Dr.Kuffner of Carnegie Mellon University in 2010, some researches [25], [26] began to apply cloud robot framework into multi-agent SLAM system. Some typical work [27], [28], which offloads part of high resource consumption work of SLAM, such as mapping and place recognition, to the cloud, while the robot only performs tracking and relocation. Besides, work [29] realizes real-time data retrieval of multiple sensors in cloud robotic system.

In contrast to the above methods, our work aims to apply federated deep learning and cloud robot framework to enhance feature extraction and place matching, so as to improve the stability of multi-agent SLAM and the accuracy of map fusion. We train a deep neural network to achieve more robust feature extraction and matching than traditional method. At the same time, we give full play to the advantages of computational resource in cloud to improve the performance of the entire multi-agent system, i.e. we try to maintain a dynamic dictionary in real time as the place changes, thus improving the matching accuracy of images from the same place. Last but not least, We adopt federated learning framework in our system so as to keep original images and private models on robot side instead of sending them to cloud server.

III. FC-SLAM

The main components of FC-SLAM are summarized here for reader convenience. An overall overview of system is shown in the Fig. 1. The system can be divided into two modules: distributed robot and cloud workstation: 1) The front end on distributed robot side is in charge of receiving sensor data, extracting features and constructing the pose map. The back end performs graph optimization and loop detection and correction. Then it outputs bag of words data of each place and local map; 2) On cloud workstation side, we train a dynamic dictionary after receiving the bag of words of images and local maps from distributed robots, and then perform place recognition and matching. Finally the cloud merges local maps and outputs a global pose graph and global map. We apply federated learning architecture in our system by maintaining private models and images on robot side while only sending bag of words data to the parameter server on cloud side. The application of federated learning framework in our system is shown in Fig. 2.

The system uses Karto SLAM [8] as the basic algorithm for robot map construction. Firstly, the single frame scan data is acquired by laser scanning. Then the submap is constructed by accumulating the scans. Pixel-accurate scan matching method is applied to generate the constraint relationship of scan and submap. Finally, Loop closing is applied to eliminate cumulative errors generated by the submaps. In the rest part, we show how we design convolutional neural network to detect features, place matching (dynamic vocabulary) algorithm and map fusion method.

A. CNN Feature Detector

In this section we introduce the design of the convolutional neural network which is used to extract feature points and descriptors.We mainly introduce the network architecture, the loss function and improvements we make to apply it in our systems.

There are many deep neural networks that are proposed to extract image feature points and descriptors. However, some networks have too many layers and even adopt recurrent neural networks. This leads to high computational complexity which is not suitable for real-time SLAM system. Therefore, we adopt a shallow but effective network for our task.

We used a VGG-style architecture that is proposed by SuperPoint [18], which is shown in Fig. 3. This network shares an encoder consisting of convolutional layers and pooling layers and activation functions. Then it splits into two networks to extract feature points and descriptors, respectively. For computational reasons, the feature point extraction network is not implemented with the classical encoding-decoding structure, but the sub-pixel convolution method is applied to implement up-sampling. The feature extraction network outputs a probability map of the same size as the original image. After the descriptors are extracted, in order to have the same format as the ORB descriptor and facilitate the calculation of the bag of words, we employ the method proposed in paper [30] to convert the descriptor into a binary format by adding a binary activation layer. The binary activation layer can be described as follows:

$$b(f(x)) = \begin{cases} +1 & f(x) \geq 0 \\ 0 & otherwise \end{cases} \tag{1}$$

The loss function is formulated as follows:

$$Loss = \frac{1}{N} \sum_{i=0}^{N} \max\left(0, d\left(c_i, p_i\right) - d\left(c_i, n_i\right)\right) \tag{2}$$

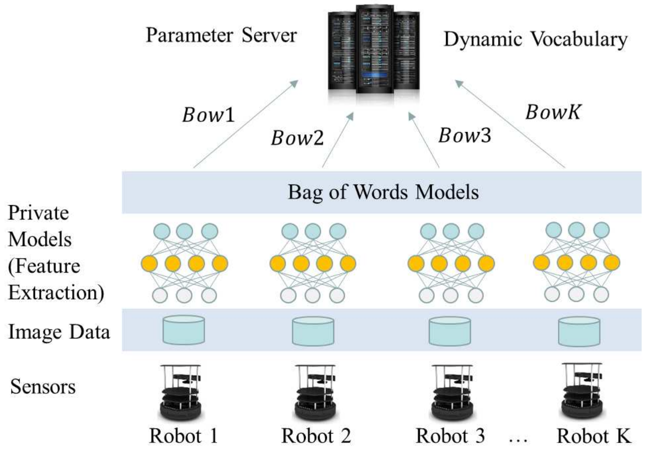


Fig. 2. Application of Federated Learning Framework. We maintain private models and images on robot side while only send bag of words data to the parameter server on cloud side.

$$d\left(c_i, p_i\right) = \left\|b\left(c_i\right) - b\left(p_i\right)\right\|_2 \tag{3}$$

Where c_i is current descriptor, p_i is positive descriptor, n_i is negative descriptor and N is the total number of keypoints.

B. Place Matching

We use the basic method of loop closing which is implemented by DBoW to conduct place matching. By calculating the bag of words model of the places, the dictionary is constructed and the similarity rate between the places is calculated. In other words, the place is represented as a collection of words. The place observed at time k is represented as $Z_k = \{z_1, z_2 \ldots z_i \ldots z_N\}$, where z_i indicates the product of term frequency (TF) and inverse document frequency (IDF) of the $i-th$ word.

The similarity between the palce x and the place y can be formulated as :

$$Similarity(\mathrm{x}, \mathrm{y}) = 2 \sum_{i=1}^{N} \left|Z_{x_i}\right| + \left|Z_{y_i}\right| - \left|Z_{x_i} - Z_{y_i}\right| \tag{4}$$

However, the performance of dictionary method may be affected by the amount and source of environment of training images, such as indoors and outdoors. This leads to the instability of place recognition and matching. On contrary to the traditional method, we make full use of the computational resource of cloud workstation and maintain a dynamic dictionary on the cloud side. Consequently, we train the dictionary through the real time data transmitted by distributed robots to update the dictionary in different environments. In order to ensure logarithmic search efficiency, here we use the k-means tree method to build vocabulary. The dynamic vocabulary algorithm can be described as Algorithm 1.

After obtaining the dictionary, we extract feature points and descriptors from the current place frame, and complete the place matching of multiple robots. The final output $M_{\mathrm{j}}^{\mathrm{i}}$ is a binary variable which denotes whether two different images i, j is from the same place.

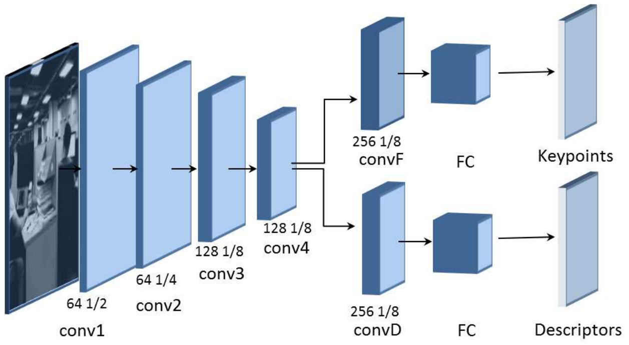


Fig. 3. Architecture of CNN Feature Detector

C. Map Fusion

The places that two robots R_1 and R_2 encounter with are recorded. They are evaluated whether the current place is the one of the recorded places. When two valid matching points are obtained, we can acquire the coordinates of the matching places (x, y). Then map stitching position and direction can be determined. Then we can compute the relation of translation and rotation of two coordinate systems with formula (5) and (6).

$$R = \begin{bmatrix} \cos(\alpha) & \sin(\alpha) & 0 \\ -\sin(\alpha) & \cos(\alpha) & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (5)$$

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = - \begin{bmatrix} 1 & 0 & \Delta x \\ 0 & 1 & \Delta y \\ 0 & 0 & 1 \end{bmatrix} R \begin{bmatrix} 1 & 0 & \Delta x \\ 0 & 1 & \Delta y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \quad (6)$$

Where R is the rotation matrix, α is rotation angle, (x, y) is the matching coordinate of R_1, (x', y') is the current coordinate of R_2, $\triangle x$, $\triangle y$ are coordinate translation values. According to the relation of rotation and translation of the two coordinate systems, we can calculate the target position of original map. Finally, we merge the maps after translation and rotation transformation.

IV. Experiments and Results

A. Evaluation of Feature Detector

In our experiments, HPatches dataset [31] is used to evaluate feature extraction networks. HPatches includes 116 scenes with 696 images, which can be divided into two types of scenes: illumination and viewpoint scenes. In order to evaluate the feature detector, we compare our method with FAST, Harris, Shi, etc. We apply the method proposed in paper [32] to calculate matching repeatability. In the experiment, non-maximum Suppression (NMS) is initialized to 4 and correctness threshold is initialized to 3 pixels, considering the possible error in homography matrix.

In order to evaluate the performance of the descriptor networks, we apply homography estimation to test the matching capability of the Hpatches data pairs. The feature points and descriptors of the two images are matched by nearest neighbor matching. Then homography matrix is calculated and homography estimation is performed. The results are shown in table I and II. We compare our method with SIFT, ORB, which are implemented by OpenCV. Our method performs better when both illumination and viewpoint changes are taken into account in various correctness thresholds.

Algorithm 1: Dynamic Vocabulary

```
Input :
        Descriptors $D_i$, $D_j$ from different place
        frame $i$, $j$;
        Train batch $L$;
        Vocabulary capacity $C$;
        Robot number $\mathrm{n}(\mathrm{n} \leq \mathrm{N})$;
        Matching threshold $\delta$;
Output:
        Dynamic vocabulary $V_t$ of time $t$;
        Matching flag $M_{\mathrm{j}}^{\mathrm{i}}$;
Initialize $t = 0$;
repeat
    Receive $D_i$, $D_j$ from distributed robots;
    if $i \geq L$ then
        if $i < C$ then
            Compute K-means tree with
             $\mathrm{D} = \{\mathrm{D}_0, \mathrm{D}_1, \ldots \mathrm{D}_i\}$;
        else
            Compute K-means tree with
             $\mathrm{D} = \{\mathrm{D}_{i-C+1}, \mathrm{D}_{i-C+2}, \ldots \mathrm{D}_i\}$;
        end
    end
    Generate Vocabulary $V_t$ with K-means tree;
    Compute the Bow $Z_i$,$Z_j$ according to $D_i$, $D_j$, $V_t$;
    Compute the similarity score $Sim\{\mathrm{i,j}\}$ with (4);
    if $Sim\{\mathrm{i,j}\} \geq \delta$ then
        Set $M_{\mathrm{j}}^{\mathrm{i}} = 1$;
    else
        Set $M_{\mathrm{j}}^{\mathrm{i}} = 0$;
    end
    $t \leftarrow t + 1$;
until $\mathrm{t} \geq \mathrm{T}$;
```

B. Evaluation of Place Recognition and Matching

In order to test the performance of place matching and dynamic dictionary algorithm on the server side, we select 2680 indoor scenes from the TUM data set for the experiment. We send a set of 40 scenes to the server every 10 seconds. Then we use existing place sets and current place set to train the dictionary. In order to evaluate the impact on place matching after performing a dictionary update, we select ten images of indoor places and set one of them as the target image. Among the remaining nine images, one comes from a different perspective of the same place as the target image, which is named as ground truth image. The remaining eight images are from other different places. We compute the significance of similarity score of the target and ground truth images in compared with other different places. The higher the significance score is, the more easily the matching places can be recognized by the system.We calculate the average and variance of the similarity score of target image and non-ground truth images. We also track the mismatching rate, which is the main indicator for evaluating the system. In addition, the number of words in the dictionary is counted as the number of received scenes increases.

TABLE I
REPEATABILITY ON HPATCHES

	Illumination changes	Viewpoint Changes
Our method	**0.659**	0.411
FAST	0.564	0.417
Harris	0.618	**0.468**
Shi	0.577	0.405

[1] Repeatability on HPatches computed with 300 points detected in common between pairs of images and with a NMS of 4.

TABLE II
HOMOGRAPHY ESTIMATION ON HPATCHES

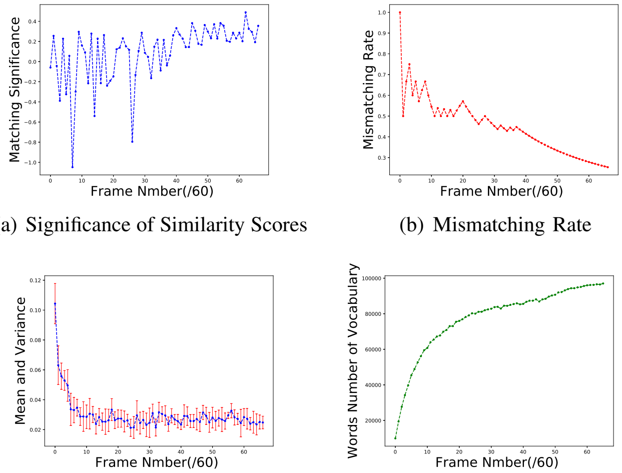
	Illumination	Viewpoint	$e = 1$	$e = 2$	$e = 3$
Our method	**0.942**	0.247	**0.329**	**0.576**	**0.637**
SIFT	0.876	**0.256**	0.311	0.532	0.548
ORB	0.542	0.133	0.123	0.297	0.369

[1] The left part of the table is homography estimation on HPatches computed with a maximum of 1000 points detected in common between pairs of images, with a threshold of correctness of 3 and a NMS of 8. The right one considers both illumination and viewpoint changes with a various threshold of correctness and NMS of 8.

As shown in Fig. 4 (a), experiment result shows the significance of similarity scores fluctuate on a large scale before the $39th$ set of images. Some negative values indicate miss matching. However, the variance of matching significance decreases and no mismatching occurs again after the cloud receives the $40th$ set of images. The overall significance value shows an upward trend during the whole training process. It indicates that we can identify matching places more easily when we increase the dictionary scale. The mismatching rate fluctuates before the 37 th set of images and it tends to go down in general, which is shown in Fig. 4 (b). It means that the increase of the dictionary scale can benefit the matching accuracy. Fig. 4 (c) presents the mean and variance of similarity scores between the target image and all non-ground truth images. We find that the mean of similarity score decreases rapidly and tends to be stable when the number of images increases, and the variance decreases gradually. This indicates that the score between different places can be maintained at a low level with the expansion of the dictionary size. It also means that the degree of interference to identify the matching place from unrelated places reduces. In addition, Fig. 4 (d) presents the dictionary size variation throughout the process.

C. *Performance of Map Fusion*

We use three laptops and one workstation for the simulation environment test. The laptops use ubuntu16.04 and ROS Kinetic system. In Gazabo, turtlebot2 carries lidar and RGB camera to provide data for SLAM nodes. The experiment setting is shown in the Fig. 5 (a).



(a) Significance of Similarity Scores

(b) Mismatching Rate

(c) Mean and variance of the Similarity Scores

(d) Number of Words of Dynamic Vocabulary

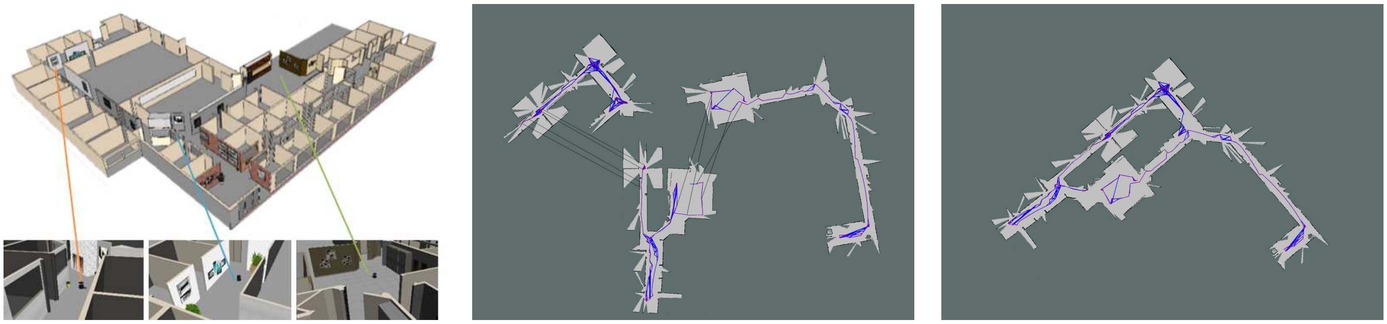
Fig. 4. Results of Place Recognition and Matching

Three robots are controlled to construct maps of the environment from their respective positions. During this process, the system analyzes the current place of one robot and matches with the places stored by other robots. When the similarity score reaches 0.85, it is judged to be a successful matching. When two matching points are obtained, the system connects the matching positions with lines so as to judge the correctness of the system matching intuitively. At the same time, the system automatically converts the map coordinates of according matching relation. Then it merges the transformed maps. The three local maps before fusion and the global map after fusion are shown in the Fig. 5 (b) and Fig. 5 (c).

V. CONCLUSIONS

In this paper, we presents a cloud based distributed SLAM architecture FC-SLAM. To improve the efficiency of collaborative SLAM, we propose a federated deep learning based feature extraction algorithm. Besides, we propose a dynamic dictionary algorithm on the cloud side to improve the place matching accuracy when the environment is changing. The experiments are implemented both in local and global map fusion. We realize more stable feature extraction and matching under viewpoint and illumination changes than traditional methods without the need of sending original images to cloud server.

In the future, we will work on 3D map fusion for cloud robotic systems with federated learning architecture, and verify the effectiveness of our methods and algorithms in public datasets and real environments.



(a) Simulated Environment Settings

(b) Local Maps and Matching Relation

(c) Global Map After Map Fusion

Fig. 5. Performance of Map Fusion

REFERENCES

[1] Cesar Cadena, Luca Carlone, Henry Carrillo, Yasir Latif, Davide Scaramuzza, José Neira, Ian Reid, and John J Leonard. Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age. *IEEE Transactions on robotics*, 32(6):1309–1332, 2016.

[2] Jorge Fuentes-Pacheco, José Ruiz-Ascencio, and Juan Manuel Rendón-Mancha. Visual simultaneous localization and mapping: a survey. *Artificial Intelligence Review*, 43(1):55–81, 2015.

[3] Ethan Rublee, Vincent Rabaud, Kurt Konolige, and Gary R. Bradski. Orb: an efficient alternative to sift or surf. In *International Conference on Computer Vision*, 2012.

[4] Herbert Bay, Andreas Ess, Tinne Tuytelaars, and Luc Van Gool. Speeded-up robust features (surf). *Computer Vision Image Understanding*, 110(3):346–359, 2008.

[5] Ming Liu, Francois Pomerleau, Francis Colas, and Roland Siegwart. Normal Estimation for Pointcloud using GPU based Sparse Tensor Voting. In *IEEE Int. Conf. on Robotics and Biomimetics (ROBIO)*, 2012.

[6] Zhe Wang, Yang Liu, Qinghai Liao, Haoyang Ye, Ming Liu, and Lujia Wang. Characterization of a rs-lidar for 3d perception. In *2018 IEEE 8th Annual International Conference on CYBER Technology in Automation, Control, and Intelligent Systems (CYBER)*, pages 564–569. IEEE, 2018.

[7] Haoyang Ye, Yuying Chen, and Ming Liu. Tightly coupled 3d lidar inertial odometry and mapping.

[8] Kurt Konolige, Giorgio Grisetti, Rainer Kümmerle, Wolfram Burgard, Benson Limketkai, and Regis Vincent. Efficient sparse pose adjustment for 2d mapping. In *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 22–29. IEEE, 2010.

[9] M. Usman Maqbool Bhutta and Ming Liu. Pcr-pro: 3d sparse and different scale point clouds registration and robust estimation of information matrix for pose graph slam.

[10] Yuxiang Sun, Ming Liu, and Max Q-H Meng. Improving rgb-d slam in dynamic environments: A motion removal approach. *Robotics and Autonomous Systems*, 89:110–122, 2017.

[11] M. Liu, C. Pradalier, F. Pomerleau, and R. Siegwart. Scale-only Visual Homing from an Omnidirectional Camera. In *Proc. of the IEEE International Conference on Robotics and Automation (ICRA)*, 2012.

[12] Richard A Newcombe, Steven J Lovegrove, and Andrew J Davison. Dtam: Dense tracking and mapping in real-time. In *2011 international conference on computer vision*, pages 2320–2327. IEEE, 2011.

[13] Jakob Engel, Vladlen Koltun, and Daniel Cremers. Direct sparse odometry. *IEEE transactions on pattern analysis and machine intelligence*, 40(3):611–625, 2017.

[14] M. Liu, C. Pradalier, Q. Chen, and R. Siegwart. A bearing-only 2d/3d-homing method under a visual servoing framework. In *2010 IEEE International Conference on Robotics and Automation*, pages 4062–4067, May 2010.

[15] Raul Mur-Artal and Juan D Tardós. Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras. *IEEE Transactions on Robotics*, 33(5):1255–1262, 2017.

[16] Edgar Simo-Serra, Eduard Trulls, Luis Ferraz, Iasonas Kokkinos, Pascal Fua, and Francesc Moreno-Noguer. Discriminative learning of deep convolutional feature point descriptors. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 118–126, 2015.

[17] Xufeng Han, Thomas Leung, Yangqing Jia, Rahul Sukthankar, and Alexander C Berg. Matchnet: Unifying feature and metric learning for patch-based matching. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3279–3286, 2015.

[18] Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Superpoint: Self-supervised interest point detection and description. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 224–236, 2018.

[19] Jakub Konečnỳ, H Brendan McMahan, Felix X Yu, Peter Richtárik, Ananda Theertha Suresh, and Dave Bacon. Federated learning: Strategies for improving communication efficiency. *arXiv preprint arXiv:1610.05492*, 2016.

[20] B. Liu, L. Wang, and M. Liu. Lifelong federated reinforcement learning: A learning architecture for navigation in cloud robotic systems. *IEEE Robotics and Automation Letters*, 4(4):4555–4562, Oct 2019.

[21] Isaac Deutsch, Ming Liu, and Roland Siegwart. A framework for multi-robot pose graph slam. In *Real-time Computing and Robotics (RCAR) 2016 IEEE International Conference on*, Angkor Wat, Cambodia, June 2016.

[22] L. Wang, M. Liu, and M. Q. H. Meng. Towards cloud robotic system: A case study of online co-localization for fair resource competence. In *2012 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, pages 2132–2137, Dec 2012.

[23] L. Wang, M. Liu, and M. Q. H. Meng. A hierarchical auction-based mechanism for real-time resource allocation in cloud robotic systems. *IEEE Transactions on Cybernetics*, 47(2):473–484, Feb 2017.

[24] J Kuffner. Cloud-enabled robots in: Ieee-ras international conference on humanoid robots. *Piscataway, NJ: IEEE*, 2010.

[25] Lujia Wang, Ming Liu, and Max Q-H Meng. A pricing mechanism for task oriented resource allocation in cloud robotics. In *Robots and Sensor Clouds*, pages 3–31. Springer, 2016.

[26] Lujia Wang, Ming Liu, Max Q.-H. Meng, and Roland Siegwart. Towards real-time multi-sensor information retrieval in cloud robotic system. In *Proceedings of the IEEE Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI)*, 2012.

[27] Ming Liu, Lujia Wang, and Roland Siegwart. DP-Fusion: A generic framework for online multi sensor recognition. In *IEEE Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI)*. IEEE, 2012.

[28] Luis Riazuelo, Javier Civera, and JM Martınez Montiel. C2tam: A cloud framework for cooperative tracking and mapping. *Robotics and Autonomous Systems*, 62(4):401–413, 2014.

[29] L. Wang, M. Liu, and M. Q. H. Meng. Real-time multisensor data retrieval for cloud robotic systems. *IEEE Transactions on Automation Science and Engineering*, 12(2):507–518, April 2015.

[30] Jiexiong Tang, Ludvig Ericson, John Folkesson, and Patric Jensfelt. Gcnv2: Efficient correspondence prediction for real-time slam. 2019.

[31] Vassileios Balntas, Karel Lenc, Andrea Vedaldi, and Krystian Mikolajczyk. Hpatches: A benchmark and evaluation of handcrafted and learned local descriptors. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5173–5182, 2017.

[32] Krystian Mikolajczyk and Cordelia Schmid. A performance evaluation of local descriptors. 2005.