2018 IEEE Intelligent Vehicles Symposium (IV)
Changshu, Suzhou, China, June 26-30, 2018

Online Cooperative 3D Mapping for Autonomous Driving

Zhe XuanYuan Boyang Li Xinyi Zhang Long Chen Kai Huang

***Abstract*—Autonomous driving requires 3D representations of the environments as high definition maps. In many cases, it is not efficient for a single vehicle to map the entire large environment. Therefore, a group of vehicles could cooperate to build maps. In this paper, we propose an approach for cooperative 3D mapping by multiple vehicles working simultaneously as a team. Each vehicle uses 3D LIDAR sensor and local mapping algorithms to build local map and the global map can be obtained by merging all the local maps in an consistent manner. The challenges in cooperative mapping lie in both accuracy and efficiency. We show that our cooperative mapping approach can save mapping time as well as reduce the accumulated error often suffered by single vehicle mapping algorithms. Meanwhile, real world experiments results indicate that our mapping algorithm can be implemented online with minimum burden imposed on communication channel and computation resources on each vehicle.**

***Index Terms*—Autonomous Driving, 3D mapping, LIDAR, Cooperative Mapping**

I. Introduction

Autonomous driving requires building 3D representations of the driving environment as high definition maps for localization, trajectory estimation, path planning and vehicle control purposes. The problem of self-localization and mapping in unknown environments is one of the most fundamental problems in mobile robotics and intelligent vehicles, which is mostly referred as the SLAM (simultaneous localization and mapping) problem. SLAM with 3D LIDARs is very popular in autonomous driving because LIDARs provide direct range measurement with relatively low error and high frequency point cloud scans.

A typical 3D mapping task can be carried out by mounting long range multi-channel (from 16 up to 128 channels) LIDARs such as Velodyne LIDARs on a vehicle together with other necessary equipment (e.g. IMU/GNSS, wheel encoder, and/or computer vision systems depending on the specific solution) and drive the vehicle along a designed route covering the mapping region to collect data. The data is processed using online or offline mapping algorithms to create the map. The process can be time consuming and many mapping algorithms are bound to cumulative drift as the mapping distance increases.

In order to enhance the efficiency and accuracy when mapping large areas, such as the driving area of a whole city, it is often necessary for two or more vehicles to participate in this mapping task simultaneously. Vehicles could build maps independently of one another and merge their maps into a global consistent one when possible. This map merging problem becomes challenging if we assume the relative coordinate transformation between the initial poses of the vehicles is unknown. In such cases, cooperating vehicles do not share a common reference frame and have to estimate the transformation between local maps based on the feature information from the map data itself.

In this paper, we contribute by designing an online cooperative 3D mapping system that can be used to create consistent global maps by multiple vehicles working cooperatively. The vehicles equipped with LIDAR scanners use scan matching based mapping algorithm to create local 3D point cloud maps. When cooperating vehicles encounter each other, they will exchange LIDAR scans with overlapping regions to identify the relative transformation of their local map reference frames. A coarse-to-fine point cloud feature extraction and matching algorithms is proposed to align overlapping maps and thus accomplish the map merging.

This paper is organized as follows: Section II presents related works in both 3D mapping and map merging. Section III systematically describes the work flow of the proposed cooperative mapping system and its detailed components. Section IV validates the proposed method with several real world experiments and Section V summarizes the paper.

II. Related Work

The cooperative mapping problem is mostly related to the SLAM problem and map merging problem, both have been discussed quite a lot in the literature. While SLAM focuses on autonomously constructing a map of the environment for each mapping vehicle, map merging helps to combine the local maps into a globally consistent one. Most SLAM techniques are based on probabilistic methods [1], where the vehicles have probabilistic motion and uncertain perception

This paper is based upon work supported by UIC research grant R201708 and NSFC grant 61501528. Zhe XuanYuan and Xinyi Zhang are with the Beijing Normal University - Hong Kong Baptist University United International College, {zhexuanyuan, xinyizhang}@uic.edu.hk, Boyang Li, Long Chen, and Kai Huang are with Sun Yat-sen University, China. {liby3, chenl46, huangk36}@mail.sysu.edu.cn

models. Example of such approaches include the Extended Kalman Filter (EKF) based methods [2], Rao-blackwellized Particle Filter (RBPF) based methods [3] and Sparse Extended Information Filter (SEIF) based ones [4]. Such methods have achieved tremendous success in solving the SLAM problem. However, the necessary computational effort makes it difficult to use them for real time 3D mappings. Scan matching based SLAM techniques [5] [6] [7] are shown to be efficient and accurate for 3D point cloud based mapping tasks. In [8], the authors incorporate an IMU and use loop closure to create large maps. In essence, this method requires batch processing and optimization to correct the IMU biases and develop accurate maps, therefore is unsuitable for applications where maps are needed in real time. On the contrary, the LOAM [9] system does not require addition IMU as odometry and the 3D mapping task can be done online as the vehicle moves around the environment. We adopt the method described in [9] as our primary 3D mapping algorithm for single vehicle.

The maps obtained by scan matching algorithms such as [7] [8] [9] are composed of point clouds registered in the local map reference frame. The map merging algorithms can be divided into different categories depending on whether the relative reference frames of the vehicles are known initially, or acquired during mapping based on some rendezvous strategy [15] [16], or completely unknown [10]. If the initial relative positions of the vehicles are known, the map merging problem are simplified as it is equivalent to the problem that all the vehicles start the mapping in the common reference frame [11] [12]. We do not make this assumption in this work because it is too restrictive in real world application. On the contrary, if the relative position of the cooperating vehicles are completely unknown, the map merging usually requires feature extracting and matching of the whole map acquired to find the structural/geometric similarities of the maps [13] [14]. Such tasks consume more computational power and thus are often harder to implement in real time. Alternatively, our proposed method makes use of the vehicle rendezvous to identify their relative pose during mapping. The algorithm can be implemented efficiently such that map merging can be done online.

III. System for 3D Cooperative Mapping

A. System Overview

The work flow for cooperative 3D mapping by multiple vehicles is illustrated in Fig. 1. Vehicles are assumed to be equipped with computers, local area wireless communication interfaces and high accuracy RTK GPS receivers. Each vehicle starts to explore the environment from its own initial position unaware of the relative poses of the other vehicles. When no vehicles are considered encountering each other, each vehicle simply carries out single vehicle mapping using the scan-matching based algorithm described in Section. III-B. Meanwhile, in order to detect peer rendezvous, the vehicles actively send out and listen to predefined beacon signals through a dedicated wireless communication channel, trying to communicate with each other that are within communication range. The beacon signals contain the current RTK GPS pose $(longitude, latitude, heading)$ of each sending vehicle. The vehicles will use both of their GPS pose information to determine whether they have "encountered" each other, by comparing the physical distance between each other to a predefined threshold. Note the rendezvous detection module in our proposed cooperative mapping framework can also be implemented using other technologies such as computer vision.

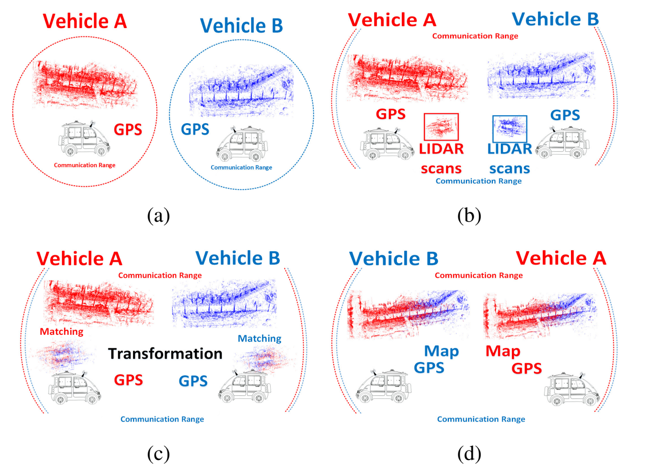


Fig. 1. Cooperative Mapping Illustration. (a) Vehicle A & B have not met each other, they do single vehicle mapping independently. Meanwhile, they broadcast their GPS pose information through wireless channel. (b) Vehicles detect nearby peer by calculating their Euclidean distance using GPS pose information, if the distance is close enough, vehicles are considered to "encounter" each other in this rendezvous point and they will start sending local LIDAR scans to each other. (c) Scan matching are conducted for remote and local scans to obtain the relative transformation of local reference frames between vehicles. (d) When scan matching is done successfully, the maps from different vehicles are merged

Once a pair of vehicles decide that they have "encountered" in some rendezvous point, they will send the recent LIDAR scans to match and merge with each other. The map merging technique will be described in more details in Section III-D. After map merging, the vehicles continue to do single mapping incrementally and share the newly acquired maps to each other whenever possible.

B. Single Vehicle Mapping

When there is only one vehicle in the team, or when the vehicles have not "encountered" each other, each vehicle will work on its own to build its local map based on 3D LIDAR point cloud scan matching algorithm. Here a "scan" refers to the point cloud data obtained when a spinning LIDAR (e.g. Velodyne VLP16) rotate every 360 degrees. To create a correct map of the driving area, the consecutive scans need to be registered into the local coordinate system. The geometric structure of overlapping 3D scans are considered for the scan matching process to provide an accurate and consistent map.

Particularly we incorporate the mapping algorithm proposed in "Laser Odometry and Mapping (Loam)" [9] as our local mapping approach because it does not need back-end opti-

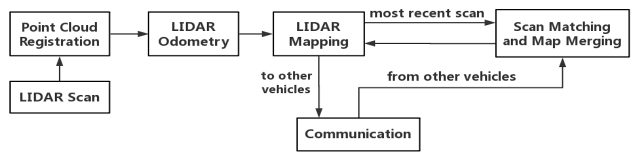


Fig. 2. System for 3D Cooperative Mapping

mization and can better guarantee the real time characteristics of the whole framework. In addition, this algorithm is accurate and does not require additional high cost inertial measurement units as odometers.

As shown in Fig. 2, the local mapping algorithm contains both the LIDAR odometry and the LIDAR mapping modules. Point cloud scans are registered incrementally to build a local map. The original algorithm proposed in [9] works with a 2D laser scanner mounted on pan-tilt. We have adapted it to accommodate multi-channel spinning LIDARs such as Velodyne VPL16 that is popular to the intelligent vehicle community.

From next section on, we will discuss in detail how to conduct cooperative mapping for multiple vehicles.

C. Communication Model and Rendezvous Peer Detection

It is assumed that all the vehicles are equipped with local area wireless communication interfaces (e.g. WIFI, DSRC, etc) so that they can transfer data between each other in a peer to peer manner, as long as the vehicles are within the mutual communication range. In this setting, each vehicle is able to detect a rendzvous peer vehicle by calculating the Euclidean Distance of the two vehicles, based on their position information. If the distance is below a certain threshold, we consider the two vehicles to be rendezvous peers. Then, they start the map merging process as described in Sec. III-D.

Intuitively, we prefer the distance threshold indicating "rendezvous" peers to be as large as possible (upper bounded by the communication range) so that it may increase the chance to detect rendezvous and then trigger cooperative mapping process. However, it may not be set too large because the LIDAR scans from two vehicles at the rendezvous point need to have enough overlapping area for better matching results. This trade-off rationale will be further explained and empirical results will be given in Section IV-A.

D. Robust Map Alignment and Merging

Once the mapping vehicles encounter each other, it means their local maps should contain some overlapping areas to be aligned so as to find out the relative transformation of their local reference frame. After that, the local maps can be further merged into a global one. This is one of the key steps of cooperative mapping.

Since we use 3D point cloud based maps in this work, the map alignment is conducted based on point cloud matching and registration algorithms [17] [18]. Given two 3D point clouds as input, a matching algorithm is to find out the optimal 4x4 relative rotation and translation matrix (RT matrix, see Equation. 1) below of the point clouds so that the distance between corresponding points in two clouds are minimized.

$$\mathcal{T}_{RT} = \begin{bmatrix} R_{3\times3} & T_{3\times1} \\ 0_{1\times3} & 1 \end{bmatrix} \tag{1}$$

One property for most matching algorithms is that the more points are in the clouds, the more time and iterations it requires to converge to optimal result. In our case, we need to match the local point cloud maps from different vehicles efficiently. Therefore, instead of matching the complete local maps obtained, we only match the most recent LIDAR scans at the vehicles rendezvous. The reason is, if the vehicles were mapping non-overlapping areas before they encounter each other, there would be very few common parts in their local maps to help with the matching. Besides, the bandwidth required to interchange data is much lower if we use the recent scans only.

As depicted in Fig. 3, after exchanging the most recently registered LIDAR scan frames, in each vehicle, the LIDAR scan from peer vehicle is matched to the corresponding local scan using point cloud matching algorithms. In reality, the two point clouds to be matched may have large translation and rotation value due to the arbitrary difference between two vehicles' local reference frames. The most commonly used ICP [17] or NDT [18] matching algorithm may take very long time to converge or even cannot converge to the global optimal. Therefore we propose to use a coarse-to-fine matching scheme to match the point clouds.

In the coarse matching phase, we try to compute an initial alignment for the pair of LIDAR scans from different vehicles. The initial alignment is obtained by extracting key points and descriptors from the point clouds and finding correspondence in them. We use the Fast point feature histograms (FPFH) [19] algorithm for this purpose. The FPFH descriptor stores the relative orientation of normals and distances between point pairs falling within the spherical neighborhood of a keypoint. Point-to-point correspondences between the keypoints can be estimated using RANSAC approach [21]. After the coarse matching phase, the two LIDAR scans should be roughly aligned. The output from this phase is an initial RT matrix describing the rough relative transformation of the two local reference frames in the vehicles. We use this RT matrix as an initial estimation for the input of the next fine matching phase.

In the fine matching phase, we use an iterative registration algorithm such as ICP [17] to further refine the matching result and increase the accuracy.

This pipeline is robust and efficient and we will show the experimental results in Section IV.

The final output from the fine matching phase would be an RT matrix accurately describing the relative transformation of two local reference frames used in the vehicles local maps. Therefore, we can merge the already built maps and overlay the point clouds into a global map by using the 4x4 RT matrix to transform the coordinate of each points, as shown in Equation 2, where M2 and M1 are point coordinates in

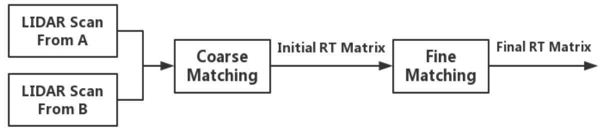


Fig. 3. The coarse-to-fine scan matching process for rendezvous peers

respective reference frames. In one word, we have successfully merged the two local maps into a global one.

$$M2 = \mathcal{T}_{RT} * M1 \tag{2}$$

When any two vehicles encounter, they can use this process to merge their local maps. Furthermore, when they have left the rendezvous point, they can keep sending the new locally registered point cloud scans to each other through the wireless communication channel, as long as they are within the communication range. The new scans can be readily merged into the global maps incrementally by directly applying the RT transform matrix obtained in the matching steps.

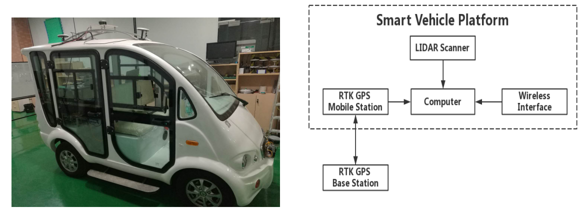
IV. Experimental Results

The online cooperative 3D mapping approached proposed in this paper has been validated through real world experiments.

For the experiments, we use two electric patrol vehicles (see Fig. 4) as the platform, with one Velodyne VLP-16 LIDAR scanner and one UnionStrong UGM512 RTK gps mobile station mounted on each vehicle. The reference frames between LIDAR and gps are calibrated. We also set up a UnionStrong RTK base station in our campus on top of a building within the experiment area. For the computation tasks, we use two Lenovo laptops, one for each vehicle, for running the mapping and communication software. The CPU and memory configuration of the laptops are i7-6500 CPU, 8G memory for one and AMD A10-5700 CPU, 4G memory for the other. Both laptops are capable of handling the required tasks proposed in this work in real time. We use Netgear R8000 WIFI routers and set them in ad-hoc mode for communication between the two vehicles. The maximum transmission rate is 3.2 Gbps and communication range is around 50 meters subject to building occlusions.

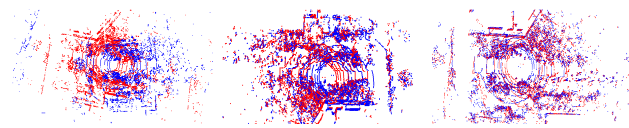
The software are implemented based on ubuntu 16.04 LTS operating system and ROS-kinetic [20]. Fig. 2 illustrates the block diagram of the sotware pipeline and modules implemented for cooperative mapping.

We validate the proposed cooperative mapping method and illustrate some of the experimental results in the following three scenarios. In Scenario A, we look into the detail results of the map alignment and merging step, which is one of the most important step in the whole cooperative mapping process. In Scenario B, we describe a complete cooperative mapping task carried out in a campus area. In Scenario C, we illustrate the advantage of cooperative mapping over single vehicle mapping in terms of time consumption and mapping accuracy.



(a) (b)

Fig. 4. Smart Vehicle Platform



(a) (b) (c)

Fig. 5. Matching result for different distance threshold. (a) 10m. (b) 3m. (c)1.5m. (result for 5m is similar to 10m so we omit it to save space

A. Scenario A

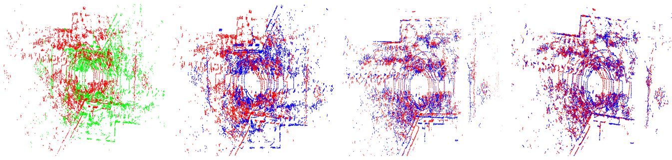
In this experiment, we first explain the rationale behind choosing the distance threshold for deciding whether two vehicles have "encountered" in the rendezvous point. Apparently, larger threshold could increase the chance of vehicle rendezvous .

We use the coarse-to-fine scan matching method proposed in Section III-D to match the LIDAR scans between vehicles for different distance thresholds such as 10m, 5m, 3m and 1.5m. The qualitative illustration is shown in Fig. 5. Clearly the matching result with 10m distance is not satisfactory and the result for 3m and 1.5m are both acceptable, with the alignment of 1.5m more accurate and neat. Table 1 quantitatively shows the matching score as defined in [21], the lower the score is, the better the matching result is. To guarantee a better map merging result, we choose to use the 1.5m distance threshold through our experiments. Of course we believe it is also fine to loose the threshold to 3m if the mapping task setting requires so.

Distance Threshold	Matching Score
10m	8.45119
5m	4.29151
3m	1.57628
1.5m	0.79729

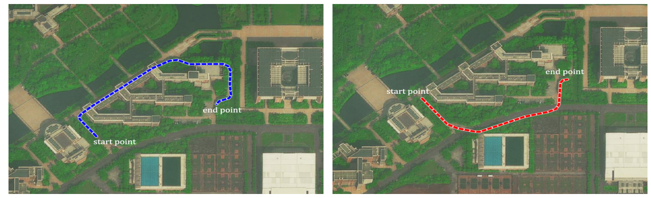
TABLE I
Matching Score for Different Distance Thresholds

When two vehicles' distance is within the threshold, they send their most recently registered LIDAR scans for matching using the coarse-to-fine matching pipeline. Because the scans are already register respectively in their local reference frames which have arbitrarily different initial position and orientations, the rotation and translation between the scans could be arbitrary too. It is hard to provide proper initial guess for fine grained iterative scan matching algorithms such as ICP



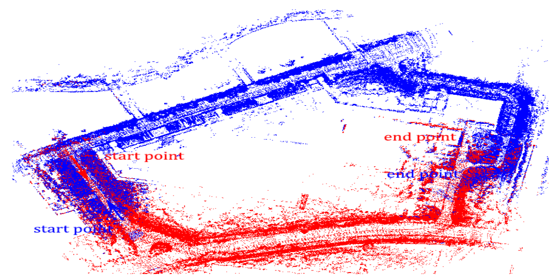
(a) (b) (c) (d)

Fig. 6. Coarse-to-Fine matching result, (a) is the original source and target scans to match. (b) is the direct ICP matching result without proper initial guess. (c) is the coarse matching result (d) is the fine matching result



(a) (b)

Fig. 7. The loop area to be mapped in scenario B, (a) is the mapping route for vehicle A and (b) is the mapping route for vehicle B



(a)

Fig. 8. Map built cooperatively of the loop area

[17]. As shown in Fig. 6(b) , directly applying ICP to the scans would lead to large matching error. Our coarse-to-fine matching approach can help tackle this issue. In Fig. 6(c), we can see the coarse matching algorithm can provide fairly good initial alignment of the scans which is then further improved by fine matching.

The LIDAR scans are down-sampled using voxel filters with leaf size of 20cm so as to reduce the transmission bandwidth and computation intensity. During rendezvous, the scans transmitted between vehicles has a typical size of 100k Bytes per frame, with up to 10Hz transmission rate, which is not a heavy burden to the wireless channel. We have done tests in different rendezvous points with different environment structures such as trees, buildings and roads and the matching results are robust and can be obtained in real time. The time requirement for the matching thread in each vehicle to carry out one successful coarse-to-fine scan matching is typically 1-3 seconds.

B. Scenario B

In this scenario, the task is to map a loop route in the campus more than 500 meters long with cooperative mapping. We use two vehicles to accomplish the task. Note that one can easily increase the number of cooperative vehicles with the proposed approach as vehicles cooperate in a pair-wise manner. As illustrated in Fig. 7, the starting and ending point of each mapping route for vehicle A & B are marked on the picture. The two vehicles starts from different initial positions to do mapping task individually, unaware of each others' locations. The route is designed such that A & B will encounter each other in the middle of their routes. The mapping is done cooperatively because each vehicle only travels a portion of the entire route to be mapped. The map merging is done when two vehicles encounter each other. The maps acquired locally by each vehicle before and after the encounter can be merged according to the scan matching results that reveal the relative transformation between two local map reference frames.

In Fig. 8, the blue point cloud is the map built by vehicle A and red by B. The overlapping area of red and blue in the left part of the figure is the rendezvous point two vehicles first encounter. The map merging technique we proposed determines the quality of overall map constructed. As shown in the figure, the rendezvous point that triggered scan matching for both vehicles is aligned very well. The loop is also closed very consistently at the end point of both route even without explicit loop closing algorithm involved in our approach. Overall speaking, the cooperative mapping approach works successfully as expected.

C. Scenario C

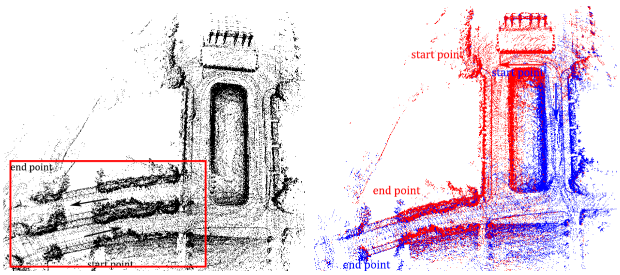
In this scenario, the task is to map a more challenging route than in scenario B. The main purpose is to demonstrate the advantage of using cooperative mapping. This scenario is more challenging because the environment along the loop is more tedious and has repeated patterns, which makes the scan registration more difficult. Scan matching based mapping algorithm are prone to accumulate errors in such environment. Cooperative mapping reduces the mapping distance of each single vehicle proportionally therefore can reduce the accumulated error.

As shown in Fig. 9, we use both single (left) and cooperative (right) mapping to accomplish the task. The overall length of the route is around 400 meters. The mapping result is shown in Fig. 10, the left one is from single mapping, clearly the map is not correctly built since in ground truth, the road in the red circle is a road without isolation belt. The error is due to the accumulated drift in the single mapping algorithm we choose. Of course an algorithm with loop closure may mitigate this problem. However, in this work, we focus on simplicity and real-time requirement of the cooperative mapping. We will discuss the loop closure problem in cooperative mapping in our future work.

The time consumed and total mapping length deviation from ground truth can be found in Table II. We can see for the two scenario shown, cooperative mapping could save one thirds to half of the time needed by single vehicle mapping. This time saving can be further increased by using more vehicles in one task. For the mapping quality, apart from

(a) (b)

Fig. 9. The loop area to be mapped in scenario C, (a) is the mapping route for single vehicle and (b) is the cooperative mapping route



(a) (b)

Fig. 10. Map built for loop area C, (a) Single mapping result(b) Cooperative mapping result

the visual validation, we use the total length of the trajectory obtained from RTK gps as the ground truth, and compare the trajectory obtained from the mapping algorithm with it. For both scenario, the deviation for cooperative mapping is smaller than single mapping because more vehicles have shared and reduced the total accumulative drift.

	Scenario	RTK Traj.(m)	Mapping Traj.(m)	Dev.(%)	Time(s)
B	single	583.429	584.859	0.25	170.4
	cooperative	638.422	639.426	0.157	120.02
C	single	394.7	384.433	2.6	121.05
	cooperative	383.207	378.542	1.21	62.31

TABLE II
TIME CONSUMPTION AND MAPPING LENGTH DEVIATION

V. CONCLUSION

This work proposed an online 3D cooperative mapping method that can be used in autonomous driving applications or other robot mapping scenarios. No knowledge is required for the vehicles' initial pose for map merging. Instead, vehicles try to interchange local maps and perform map merging during rendezvous. An efficient coarse-to-fine scan matching algorithm is proposed for online and robust scan matching. Real world experiments are done to validate the proposed method. We make our data set publicly available to ease future evaluation [1]. For future work, we plan to improve the cooperative mapping so that map merging can be carried out when two vehicles travel through common areas at different time. And multiple rendezvous will be leveraged for loop closure to refine the map built.

[1]https://pan.baidu.com/s/1dHhdxFr

REFERENCES

[1] Thrun, Sebastian. "Probabilistic robotics." Communications of the ACM 45.3 (2002): 52-57.
[2] Dissanayake, M. G., Newman, P., Clark, S., Durrant-Whyte, H. F., Csorba, M. A solution to the simultaneous localization and map building (SLAM) problem. IEEE Transactions on robotics and automation, 17(3), 229-241, 2001.
[3] Grisetti, G., Stachniss, C., Burgard, W. Improved techniques for grid mapping with rao-blackwellized particle filters. IEEE transactions on Robotics, 23(1), 34-46, 2007.
[4] Thrun, S., Liu, Y., Koller, D., Ng, A. Y., Ghahramani, Z., Durrant-Whyte, H. Simultaneous localization and mapping with sparse extended information filters. The International Journal of Robotics Research, 23(7-8), 693-716,2004.
[5] Holz, D., Behnke, S. (2010, May). Sancta simplicitas-on the efficiency and achievable results of SLAM using ICP-based incremental registration. In Robotics and Automation (ICRA), IEEE International Conference on (pp. 1380-1387), 2004.
[6] Nchter, A., Lingemann, K., Hertzberg, J., Surmann, H. 6D SLAM3D mapping outdoor environments. Journal of Field Robotics, 24(89), 699-722, 2007.
[7] Moosmann, F., Stiller, C. Velodyne slam. In Intelligent Vehicles Symposium (IV), 2011 IEEE (pp. 393-398). IEEE, 2011.
[8] Zlot, R., Bosse, M. Efficient large-scale 3D mobile mapping and surface reconstruction of an underground mine. In Field and service robotics (pp. 479-493). Springer, Berlin, Heidelberg, 2014.
[9] Zhang, J., Singh, S. LOAM: Lidar Odometry and Mapping in Real-time. In Robotics: Science and Systems (Vol. 2), 2014.
[10] Andersone, I. The Characteristics of the Map Merging Methods: A Survey. Scientific Journal of Riga Technical University. Computer Sciences, 41(1), 113-121, 2010.
[11] Burgard, W., Moors, M., Schneider, F. Collaborative exploration of unknown environments with teams of mobile robots. In Advances in plan-based control of robotic agents (pp. 52-70). Springer, Berlin, Heidelberg, 2002.
[12] Simmons, R., Apfelbaum, D., Burgard, W., Fox, D., Moors, M., Thrun, S., Younes, H. Coordination for multi-robot exploration and mapping. In Aaai/Iaai (pp. 852-858), 2000.
[13] Konolige, K., Fox, D., Limketkai, B., Ko, J., Stewart, B. Map merging for distributed robot navigation. In Intelligent Robots and Systems,(IROS 2003). Proceedings. 2003 IEEE/RSJ International Conference on (Vol. 1, pp. 212-217). 2003.
[14] Amigoni, F., Gasparini, S., Gini, M. Merging partial maps without using odometry. In Multi-Robot Systems. From Swarms to Intelligent Automata Volume III (pp. 133-144). Springer, Dordrech, 2005t.
[15] Roy, N., Dudek, G. Collaborative robot exploration and rendezvous: Algorithms, performance bounds and observations. Autonomous Robots, 11(2), 117-136, 2001.
[16] Tungadi, F., Lui, W. L. D., Kleeman, L., Jarvis, R. Robust online map merging system using laser scan matching and omnidirectional vision. In Intelligent Robots and Systems (IROS), IEEE/RSJ International Conference on (pp. 7-14), 2010.
[17] Rusinkiewicz, S., Levoy, M. Efficient variants of the ICP algorithm. In 3-D Digital Imaging and Modeling. Proceedings. Third International Conference on (pp. 145-152), 2010.
[18] Biber, P., Straer, W. The normal distributions transform: A new approach to laser scan matching. In Intelligent Robots and Systems, (IROS 2003). Proceedings. 2003 IEEE/RSJ International Conference on (Vol. 3, pp. 2743-2748). 2003.
[19] Rusu, R. B., Blodow, N., Beetz, M. Fast point feature histograms (FPFH) for 3D registration. IEEE International Conference on Robotics and Automation, 2009. ICRA'09. (pp. 3212-3217).
[20] Quigley, Morgan, et al. "ROS: an open-source Robot Operating System." ICRA workshop on open source software. Vol. 3. No. 3.2. 2009.
[21] Holz, Dirk, et al. "Registration with the point cloud library: A modular framework for aligning in 3-D." IEEE Robotics Automation Magazine 22.4 : 110-124.(2015)