CVIDS: A Collaborative Localization and Dense Mapping Framework for Multi-Agent Based Visual-Inertial SLAM

Tianjun Zhang, Lin Zhang, *Senior Member, IEEE*, Yang Chen, and Yicong Zhou, *Senior Member, IEEE*

***Abstract*— Nowadays, visual SLAM (Simultaneous Localization And Mapping) has become a hot research topic due to its low costs and wide application scopes. Traditional visual SLAM frameworks are usually designed for single-agent systems, completing both the localization and the mapping with sensors equipped on a single robot or a mobile device. However, the mobility and work capacity of the single agent are usually limited. In reality, robots or mobile devices sometimes may be deployed in the form of clusters, such as drone formations, wearable motion capture systems, and so on. As far as we know, existing SLAM systems designed for multi-agents are still sporadic, and most of them have non-negligible limitations in functions. Specifically, on one hand, most of the existing multi-agent SLAM systems can only extract some key features and build sparse maps. On the other hand, schemes that can reconstruct the environment densely cannot get rid of the dependence on depth sensors, such as RGBD cameras or LiDARs. Systems that can yield high-density maps just with monocular camera suites are temporarily lacking. As an attempt to fill in the research gap to some extent, we design a novel collaborative SLAM system, namely CVIDS (Collaborative Visual-Inertial Dense SLAM), which follows a centralized and loosely coupled framework and can be integrated with any existing Visual-Inertial Odometry (VIO) to accomplish the co-localization and the dense reconstruction. Integrating our proposed robust loop closure detection module and two-stage pose-graph optimization pipeline, the co-localization module of CVIDS can estimate the poses of different agents in a unified coordinate system efficiently from the packed images and local poses sent by the client-ends of different agents. Besides, our motion-based dense mapping module can effectively recover the 3D structures of selected keyframes and then fuse their depth information to the global map for reconstruction. The superior performance of CVIDS is corroborated by both quantitative and qualitative experimental results. To make our results reproducible, the source code has been released at https://cslinzhang.github.io/CVIDS.**

***Index Terms*— Multi-agent, monocular camera suite, dense mapping, visual-inertial odometry.**

I. Introduction

A PROFOUND knowledge of the environment is indispensable in numerous fields, ranging from augmented reality [1], [2], [3] to autonomous driving [4], [5], [6]. To obtain such an understanding, the SLAM (Simultaneous Localization And Mapping) technique, which can model the surrounding environment only by equipped sensors, is one of the most practical solutions. Among all of the research branches in this field, visual SLAM, VSLAM in short, has drawn many interests in recent years [7], [8], [9], [10], [11], [12], [13], on account of its compact and affordable sensor configurations. Existing VSLAM systems may be designed for UAVs (Unmanned Aerial Vehicle), wheeled robots, or hand-held devices, but mostly can only be applied in the single-agent manner. However, in reality, sometimes the robots or devices are put into use in the form of clusters or formations, such as drone formations and wearable motion capture systems. In such cases, since the single-agent-oriented systems cannot yield the relative poses of different agents in a unified coordinate system, they will no longer be appropriate. Instead, collaborative SLAM frameworks for multi-agent systems should be employed.

In terms of the density of the constructed map, SLAM systems roughly fall into two categories, sparse ones [7], [8], [9], [10] and dense ones [11], [12], [13]. For one frame, sparse systems extract and reconstruct only a set of key features in the image, whereas dense ones aim to utilize all pixels. Sparse maps usually play important roles in the tasks of tracking and localization. However, due to the insufficient density, they cannot effectively support some significant decision-making tasks, like obstacle avoidance. Therefore, in recent years, SLAM systems are often required to be able to recover the dense 3D structure of the surrounding environment, especially in industry. As a sub-branch, collaborative SLAM frameworks are naturally pinned on similar expectations.

However, most of the existing collaborative SLAM systems still cannot fully satisfy the researchers' requirements in terms of dense mapping. On the one hand, most existing

Manuscript received 9 March 2022; revised 12 September 2022; accepted 29 September 2022. Date of publication 14 October 2022; date of current version 20 October 2022. This work was supported in part by the National Key Research and Development Project under Grant 2020YFB2103900; in part by the National Natural Science Foundation of China under Grant 62272343, Grant 61973235, and Grant 61936014; in part by the Shanghai Science and Technology Innovation Plan under Grant 20510760400; and in part by the Shuguang Program of Shanghai Education Development Foundation and Shanghai Municipal Education Commission under Grant 21SG23. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Huanqiang Zeng. *(Corresponding author: Lin Zhang.)*

Tianjun Zhang, Lin Zhang, and Yang Chen are with the School of Software Engineering, Tongji University, Shanghai 201804, China (e-mail: 1911036@tongji.edu.cn; cslinzhang@tongji.edu.cn; 2011439@tongji.edu.cn).

Yicong Zhou is with the Department of Computer and Information Science, University of Macau, Macau 999078, China (e-mail: yicongzhou@um.edu.mo).

Digital Object Identifier 10.1109/TIP.2022.3213189

collaborative VSLAM systems can only yield sparse maps. Related researchers spend most of the processor's power in the co-localization of multi-agents, ignoring the denseness requirements on mapping. On the other hand, the existing dense systems almost all resort to specific sensors to recover the depth information. LiDARs, stereo cameras and RGB-D cameras are all commonly used depth sensors. Compared with monocular suites, like the monocular visual-inertial suites, LiDARs and stereo cameras are usually much more expensive and cumbersome, and RGB-D cameras usually underperform in outdoor environments. Thus far, to the best of our knowledge, a collaborative SLAM system that can reconstruct the scene densely just with the monocular suite is still lacking.

As an attempt to fill in the research gap to some extent, in this work we propose a novel dense collaborative SLAM framework, namely CVIDS. As far as we know, this is the first collaborative SLAM system that can accomplish dense reconstruction without the assistance of depth sensors. Our contributions can be mainly summarized as follows.

1) The first dense collaborative SLAM system, CVIDS, that does not rely on depth sensors is implemented, implying that by using CVIDS the depth information is recovered by algorithms rather than depth sensors, hereby the hardware cost can be greatly saved. Thanks to the acceleration of GPU, both the collaborative localization of multi-agents and the dense reconstruction can be achieved in real time.
2) A novel robust loop closure detection (LCD) strategy for the Visual-Inertial Odometry (VIO) based on the pairwise consistency evaluation is designed. Taking the observability of the VIO into consideration, we utilize the four-DoF (Degree of Freedom) error instead of the full-DoF one for consistency evaluation. Besides, since we have deduced the covariance of the error state, the error can be computed in a probabilistic way rather than heuristically. In CVIDS, such a strategy is adopted, thus the loop closure detection module in CVIDS can provide a set of consistent long-term data associations and eliminate outlier measurements, which significantly improves the localization stability.
3) An efficient two-stage pose-graph optimization pipeline of a cascade structure is designed and such a pipeline is integrated to the back-end of CVIDS. The first stage of the pipeline actually performs the EM-based pose smoothing, which aims to provide better initial values to the second stage for better convergence with an economical time cost, while the second stage fulfills the conventional non-linear optimization. Such a pipeline shows an excellent convergence speed while ensuring the accuracy, which guarantees the real-time performance of CVIDS.
4) A novel solution to the motion-based depth estimation is proposed. On the basis of the multi-view stereo, we further fuse the depth priori of sparse features and introduce the semi-global regularization for the smoothness of the recovered depth map. Thanks to the introduced extra prior knowledge, both the accuracy in weak-texture regions and the convergence speed of the depth estimation of our scheme have been significantly improved. Besides, we also apply a probabilistic depth filter to fuse the depth estimation from different matching frames so as to weaken the adverse effects brought by noise and outliers.

The remainder of this paper is organized as follows. Sect. II introduces related work and analyzes the existing research gaps. The overall framework of CVIDS is summarized in Sect. III. Details about the collaborative localization pipeline and the dense mapping module in CVIDS are presented in Sect. IV and V, respectively. Experimental results are reported in Sect. VI. Finally, Sect. VII concludes the paper.

II. Related Work

A. Monocular Suites Based Dense SLAM

Compared with RGBD-based mapping [14], [15], [16], owing to the affordable manufacturing cost and the lightweight structure of the sensor, the problem of monocular dense mapping in an online manner has attracted a lot of researching interests in the past decade or so. Since the depth information will be lost during the imaging process of a monocular camera, compared with RGBD-based or stereo-based ones, the dense reconstruction based on a monocular camera is much more challenging. Most of the solutions in this field in early years were designed for the offline environment, while with the continuous development in both algorithms and hardware, more and more researchers tried to complete the real-time dense reconstruction in an online manner.

In [17], Pradeep et al. firstly presented a system for real-time reconstruction with a web camera, namely MonoFusion. On account of its relatively rough implementation in the tracking module, the localization stability of MonoFusion is somewhat unsatisfactory, and it can only perform well in tasks with small-scale workspaces, such as the model scanning. Afterwards, as a milestone work of the monocular SLAM, LSD-SLAM [12] which exhibits a distinguished performance was proposed. LSD-SLAM can accomplish both the localization and the real-time semi-dense mapping simultaneously without the assistance of GPU. However, the densities of its yielded maps were insufficient so that it was considered as "semi-dense" rather than "dense". In the same year, another influential system, namely ReMode [18], was presented. Its authors resorted to the probabilistic depth filter presented in [19] and reused the localization module of SVO, which is the salient work proposed by Forster et al. in [20]. Since the depth estimation in ReMode is fully based on template matching, it usually underperforms in weak-texture regions.

Except for the aforementioned "pure" monocular systems, researchers also attempted to further introduce other lightweight sensors such as IMU (Inertial Measurement Unit) to their schemes. In 2017, Yang et al. presented a novel dense SLAM system under the monocular visual-inertial configuration, namely VI-MEAN [21]. Inheriting the implementations of VINS-Mono [22], VI-MEAN also integrates the classic stereo-vision algorithm, SGM (Semi-Global Matching) [23]. Thanks to the introduction of the semi-global regularization, VI-MEAN shows decent performance in weakly

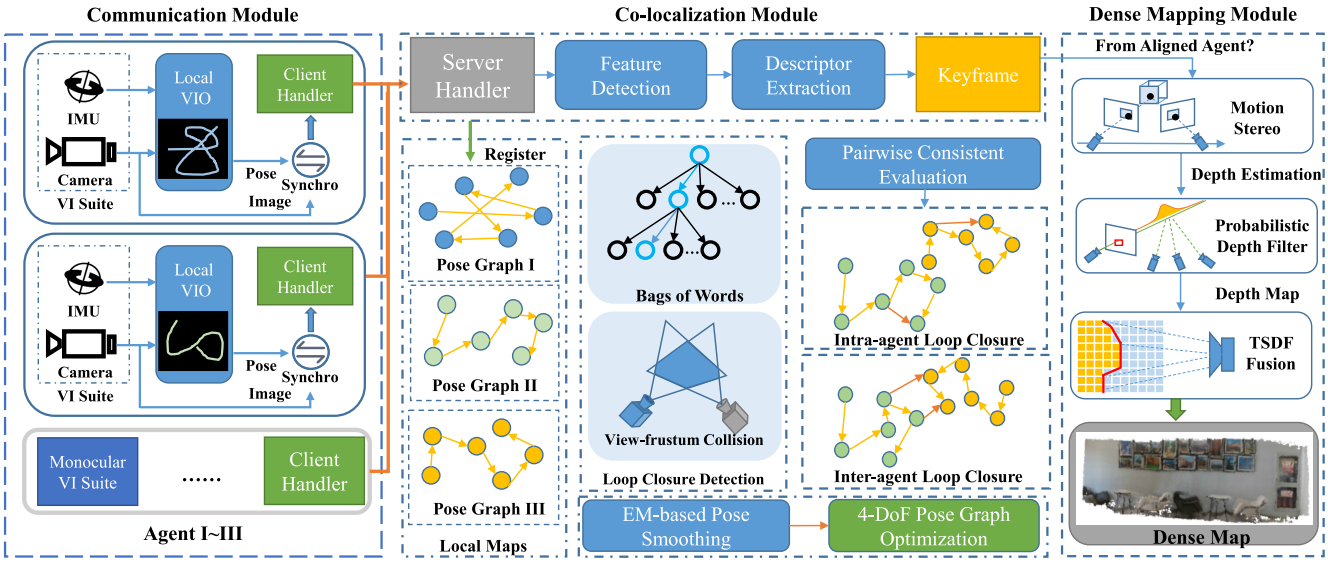


Fig. 1. Overall architecture of CVIDS. The client handler in the communication module running on each agent packs and then sends the raw data of keyframes to the handler of the central server-end. Then the co-localization module will align the local coordinate systems of different agents to a unified global one. After that, the dense mapping module will estimate the depth maps of selected keyframes and fused them to the global TSDF map. Finally, via meshing, the dense structure of the scene can be recovered.

textured regions. However, its authors ignored the culling of outliers in the depth estimation phase while directly fused the depth map recovered from two frames to the global TSDF map, so a considerable amount of outliers were prone to appear in the final constructed maps.

B. Collaborative VSLAM With Monocular Suites

The earliest monocular collaborative SLAM system can be traced back to the work of Forster et al. in [24], which is specially designed for Micro Aerial Vehicles and follows a traditional structure-from-motion pipeline. As a seminal work, it is relatively straightforward in implementations and current state-of-the-art outperforms it in both accuracy and robustness. In [25], Zou and Tan presented another milestone collaborative SLAM system namely CoSLAM, which takes the interference brought by dynamic objects into account so as to improve the robustness. In CoSLAM, all agents are required to be strictly time-synchronized. Specifically, all agents' equipped cameras must capture images simultaneously, which undoubtedly increases the hardware cost. In [26], Deutsch et al. creatively abstracted the client-end of the collaborative SLAM system. In Deutsch et al.'s system, the odometry running on each agent is regarded as a black box, that is, only its output map points and poses are utilized, ignoring detailed implementations. Under such a mechanism, the system can theoretically be integrated with any appropriate visual odometry, exhibiting an outstanding adaptation ability to the hardware environment. However, as a 2D system, the current advanced 3D visual odometries can't be integrated with it. CCM-SLAM presented by Schmuck and Chli in [27] is a tightly-coupled monocular collaborative SLAM system with an outstanding localization accuracy. It possesses a relatively modern architecture and is currently the state-of-the-art in this field. In addition to typical monocular systems, in recent years, many researchers devoted themselves to upgrading the sensor to a visual-inertial monocular suite to improve the localization stability, and some remarkable works have already been released [22], [28], [29]. However, it's a pity that most of these systems just support the single-agent mode. As far as we know, the only collaborative SLAM system designed for monocular visual-inertial suites is CVI-SLAM proposed in [30]. CVI-SLAM follows a similar design as CCM-SLAM [27], while the equipped sensor on each agent is substituted from a monocular camera to a visual-inertial suite. Due to the upgrading of the sensor, the final localization accuracy of CVI-SLAM is also significantly improved.

Although a lot of research passions have been devoted to the collaborative SLAM under the sensor configurations of monocular suites, as far as we know, existing schemes can only recover the sparse structure of the scene rather than the dense one. Thus, application scopes of these schemes in reality may still be limited. Collaborative SLAM systems that can yield dense maps just with monocular camera suites are still lacking.

III. System Overview

The overall framework of CVIDS is illustrated in Fig. 1. It mainly consists of three modules, including the communication module, the co-localization module, and the dense mapping module. Among them, both the co-localization module and the dense mapping one only run on the central server, while the communication module runs on both the server-end and the client-end corresponding to each agent. The client-end of the communication module is well encapsulated and can theoretically be integrated with any existing VIO (Visual-Inertial Odometry), while in our current implementations, we temporarily selected VINS-Mono [22]. Such a sub-module takes poses, map points and images from the local VIO as the

input, packs the received raw data into messages and then send the messages to the central server. The server-end sub-module is responsible for unpacking the data received from the client-end of any agent. For the co-localization module and the dense mapping module, we will introduce them in detail in Sect. IV and Sect. V, respectively.

IV. Co-Localization Module

The co-localization module is mainly responsible for aligning the local reference coordinate systems (CSs) and then co-localizing all registered agents in a unified reference CS. After the server-end communication module receives and unpacks the raw data sent by the client, the co-localization module will primarily construct the keyframe object and register it to the corresponding local map in the server-end. Afterwards, the loop closure detection will be conducted, including the intra-agent detection and the inter-agent one. If the inter-agent loop closure is successfully triggered, corresponding unaligned local maps can then be aligned. Besides, the pairwise consistency of all loop closure measurements will be evaluated so as to eliminate outliers. Once a new loop closure is established, our two-stage optimization pipeline will be activated.

A. Keyframe Construction and Registration

The raw data of each frame sent by the client mainly consist of an image, the associated pose under the local reference CS and the corresponding map points (including the 2D pixel coordinates and the relevant 3D positions). During the construction of a new keyframe, preparing for the subsequent LCD, the BRIEF descriptor [31] of each map point is extracted. Besides, the sparsity of map points will limit the performance of the LCD. Thus, we also detect the Shi-Tomasi corners [32] on the image for supplementary, and then compute their corresponding BRIEF descriptors. For the sake of distinction, we call those points sent by clients as "map points" and the Shi-Tomasi corners as "2D features". Afterwards, a server-side keyframe object can be constructed and registered to the local map of the corresponding agent. It's worth mentioning that, when an agent sends a keyframe to the central server for the first time, a local map dedicated to the agent will be created and all subsequent frames from the same agent will be stored in the local map.

B. Loop Closure Detection and Map Alignment

For a newly constructed keyframe, we first search for loop candidates among all past frames via the Bags-of-Words (BoW) model. Subsequently, we match map points of the selected loop frame with 2D features of the current frame to form a set of 3D-2D pairs, and then solve the PnP (Perspective-n-Points) problem under the RANSAC framework to obtain the relative pose between those two frames. For the strategy of selecting the qualified loop frame from multiple candidates, please refer to Sect. IV-C.

If the selected qualified loop closure is across two different local maps and one of them has already been aligned while the other hasn't, the transformation between the reference CSs of these two local maps will be determined to align the unaligned one. It's worth mentioning that, since both the pitch angle and the roll angle are observable for VIO, the alignment parameters that need to determine are merely about the translation and the yaw angle.

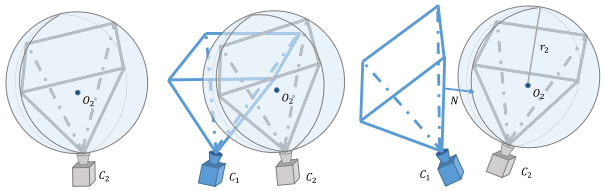


Fig. 2. Illustration of the view-frustum-based LCD. The left one shows the bounding sphere of one frame, the middle one illustrates the successful case of the detection while the right one is the failure case.

To fully make use of the advantages of multi-agent systems in terms of the coverage of observations, the view-frustum-based LCD is further conducted to supplement data associations. For two frames $\mathcal{F}_1$ and $\mathcal{F}_2$ from different aligned local maps, we first check if these two frames are theoretically common-view. Since it's complicated to determine whether two frustums intersect, a certain degree of simplification is made. Specifically, as shown in Fig. 2, if the bounding sphere of $\mathcal{F}_2$'s view-frustum is fully outside the view-frustum of $\mathcal{F}_1$, it's impossible for these two frames to be common-view. The center $\boldsymbol{O}_2$ and radius r_2 of $\mathcal{F}_2$'s bounding sphere are given as,

$$\begin{aligned} \boldsymbol{O}_2 &= \boldsymbol{T}_2^{-1}[0, 0, r_2, 1]^T \\ r_2 &= \frac{D(1 + 2\tan^2\theta)}{2}, \end{aligned} \tag{1}$$

where $\boldsymbol{T}_2$ is $\mathcal{F}_2$'s pose, D is the corresponding range of visibility, and θ is the angle of the FoV (Field of View) of $\mathcal{F}_2$'s corresponding camera. If two frames are common-view, the following condition should be satisfied,

$$\exists \boldsymbol{N} \in \mathcal{N}, |(\boldsymbol{T}_1\boldsymbol{O}_2) \cdot \boldsymbol{N}| - r_2 < 0, \tag{2}$$

where $\mathcal{N}$ is the set that consists of normal vectors of all surfaces of the view-frustum in its camera CS. Eq. 2 checks if the view-frustum of $\mathcal{F}_1$ is intersected with the bounding sphere of $\mathcal{F}_2$, which is also the prerequisite for $\mathcal{F}_1$ and $\mathcal{F}_2$ to be common-view. Only if Eq. 2 is satisfied, we will further check whether the relative pose between $\mathcal{F}_1$ and $\mathcal{F}_2$ can be recovered successfully, so as to determine whether the inter-agent constraint between $\mathcal{F}_1$ and $\mathcal{F}_2$ can be established.

C. Qualified Loop Frame Selection

In both the BoW-based loop closure detection and the view-frustum-based one, for the same current frame, generally there won't be a unique output but multiple candidates may be yielded. To ensure the stability of the loop closure, we only select the best one among all candidates each time. To align all local maps as soon as possible and eliminate accumulated errors more effectively, for all candidates, their priorities are assigned according to the following three principles:

1) If the local map which the current frame belongs to has already been aligned, frames in unaligned local maps are preferred, while in the unaligned case, the selected frame must be in aligned local maps.
2) On the premise of the first principle, frames in different local maps from the current one will be selected preferentially.
3) On the premise of the first two principles, the oldest candidate will be chosen as a priority.

To ensure the correctness of the loop closure, a stringent inspection mechanism to determine whether the loop closure is qualified is adopted, including the requirements on the quantity of matched feature pairs, and the limitation on the amount of rotation and translation of the relative pose. Only when a candidate passes all the checks, the frame can be selected to be the loop frame of the current one. For all candidate loop frames, according to the aforementioned priorities, we further try to recover the relative pose between candidate frames and the current one and check if the loop closure is qualified in order until success. And then the corresponding constraint representing the relative pose will be established between the current frame and the selected loop closure one.

D. Pairwise Consistency Evaluation of Loop Closure Measurements

Outlier measurements of loop closures usually have a devastating impact on the accuracy of localization. Therefore, before the optimization, the correctness of all loop closures should also be verified so as to eliminate wrong data associations. Motivated by [33], we adopted the Pairwise Consistency Maximization (PCM) to find a group of correct and consistent loop closures, and other measurements are considered to be outliers that should be abandoned.

We represent the loop closure measurement between two frames $\mathcal{F}_i$ and $\mathcal{F}_j$ as l_{ij}. Given a set of raw loop closure measurements, $\mathcal{L}$, we need to find its largest pairwise internally consistent subset, $\mathcal{L}_C$, and then eliminate other measurements that not belong to $\mathcal{L}_C$. For any two measurements $l_{ij}, l_{lk} \in \mathcal{L}_C$, they are consistent with respect to the consistency metric C and the threshold γ if,

$$C(l_{ij}, l_{lk}) < \gamma. \tag{3}$$

Based on the definition of "consistency", an undirected graph can be established, in which each node stands for a loop closure measurement and two nodes are connected if their corresponding measurements are consistent. Then, the task of finding the subset $\mathcal{L}_C$ can be transformed into an instance of the maximum clique problem from graph theory. As a classical problem, dozens of potential solutions have already been proposed [34], [35], [36], [37]. In our implementations, Pattabiraman et al.'s scheme [37] is adopted.

Since the maximum clique problem can be solved effectively and efficiently with existing schemes, in this paper, we mainly focus on the definition of the consistency metric C. Taking the observability of the VIO into consideration, different from the most commonly utilized full-DoF metrics, our metric C is only defined on four DoFs (the yaw angle and the translation). Given two measurements l_{ij} and l_{lk}, their consistency score can be given as,

$$C(l_{ij}, l_{lk}) = \|\boldsymbol{E}(l_{ij}, l_{lk})\|_{\boldsymbol{\Sigma}_{ijlk}^{-1}}, \tag{4}$$

where $\|*\|_\Sigma$ denotes the Mahalanobis distance, Σ_{ijlk} is the covariance matrix of the error state, and the error $\boldsymbol{E}(l_{ij}, l_{lk})$ is defined as,

$$\begin{aligned} \boldsymbol{E}(l_{ij}, l_{lk}) &= [e_{ijlk}^{yaw}, (\boldsymbol{e}_{ijlk}^{\boldsymbol{t}})^T]^T \\ e_{ijlk}^{yaw} &= \hat{\theta}_{ij}^{yaw} + \theta_{jl}^{yaw} + \hat{\theta}_{lk}^{yaw} + \theta_{ki}^{yaw} \\ \boldsymbol{e}_{ijlk}^{\boldsymbol{t}} &= [\hat{\boldsymbol{T}}_{ij}\boldsymbol{T}_{jl}\hat{\boldsymbol{T}}_{lk}\boldsymbol{T}_{ki}]_{\boldsymbol{t}}, \end{aligned} \tag{5}$$

where $\hat{\theta}_{ij}^{yaw}$ and $\hat{\theta}_{lk}^{yaw}$ are relative yaw angles of measurements l_{ij} and l_{lk}, respectively, $\hat{\boldsymbol{T}}_{ij}$ and $\hat{\boldsymbol{T}}_{lk}$ are relative pose matrices of measurements l_{ij} and l_{lk}, respectively, θ_{jl}^{yaw} and θ_{ki}^{yaw} are current relative yaw angles' estimates from $\mathcal{F}_j$ to $\mathcal{F}_l$ and from $\mathcal{F}_k$ to $\mathcal{F}_i$, respectively, and $\hat{\boldsymbol{T}}_{jl}$ and $\hat{\boldsymbol{T}}_{ki}$ are corresponding relative pose matrices. The symbol $(*)_{\boldsymbol{t}}$ stands for the translation vector of the inner pose matrix. Next, we will introduce how to compute the covariance matrix $\boldsymbol{\Sigma}_{ijlk}$ in detail.

Defining $\boldsymbol{p}_{jl}$ and $\boldsymbol{p}_{ki}$ as the corresponding four-DoF pose vectors of $\boldsymbol{T}_{jl}$ and $\boldsymbol{T}_{ki}$, respectively. For the covariance matrix $\boldsymbol{\Sigma}_{ijlk}$, we approximate it linearly as,

$$\boldsymbol{\Sigma}_{ijlk} = \boldsymbol{J}_{jl}\boldsymbol{\Sigma}_{jl}\boldsymbol{J}_{jl}^T + \boldsymbol{J}_{ki}\boldsymbol{\Sigma}_{ki}\boldsymbol{J}_{ki}^T, \tag{6}$$

where $\boldsymbol{\Sigma}_{jl}$ and $\boldsymbol{\Sigma}_{ki}$ are covariance matrices of $\boldsymbol{p}_{jl}$ and $\boldsymbol{p}_{ki}$, respectively, $\boldsymbol{J}_{jl}$ and $\boldsymbol{J}_{ki}$ are Jacobian matrices of $\boldsymbol{E}(l_{ij}, l_{lk})$ to $\boldsymbol{p}_{jl}$ and $\boldsymbol{p}_{ki}$, respectively, which can be given as,

$$\begin{aligned} \boldsymbol{J}_{jl} &= \begin{bmatrix} 1 & \boldsymbol{0}_{1\times 3} \\ \frac{\partial \hat{\boldsymbol{R}}_{ij}\boldsymbol{R}_{jl}\hat{\boldsymbol{R}}_{lk}\boldsymbol{t}_{ki}}{\partial \theta_{jl}^{yaw}} + \frac{\partial \hat{\boldsymbol{R}}_{ij}\boldsymbol{R}_{jl}\hat{\boldsymbol{t}}_{lk}}{\partial \theta_{jl}^{yaw}} & \hat{\boldsymbol{R}}_{ij} \end{bmatrix} \\ \boldsymbol{J}_{ki} &= \begin{bmatrix} 1 & \boldsymbol{0}_{1\times 3} \\ \boldsymbol{0}_{3\times 1} & \hat{\boldsymbol{R}}_{ij}\boldsymbol{R}_{jl}\hat{\boldsymbol{R}}_{lk} \end{bmatrix}. \end{aligned} \tag{7}$$

As for the covariance matrices $\boldsymbol{\Sigma}_{jl}$ and $\boldsymbol{\Sigma}_{ki}$, they can be computed via the state propagation. In our scheme, the motion equation of the odometry is formulated as,

$$\begin{aligned} \boldsymbol{p}_{j+1} &= f(\boldsymbol{p}_j, \hat{\boldsymbol{p}}_{j+1j}) \\ \theta_{j+1}^{yaw} &= \theta_j^{yaw} + \hat{\theta}_{j+1j}^{yaw} \\ \boldsymbol{t}_{j+1} &= \hat{\boldsymbol{R}}_{j+1j}\boldsymbol{t}_j + \hat{\boldsymbol{t}}_{j+1j}. \end{aligned} \tag{8}$$

Thus, the propagation of the covariance matrix can be given as,

$$\boldsymbol{\Sigma}_{m+1j} = \boldsymbol{G}_m\boldsymbol{\Sigma}_{mj}\boldsymbol{G}_m^T + \boldsymbol{H}_m\boldsymbol{\Sigma}_n\boldsymbol{H}_m^T, \tag{9}$$

where $\boldsymbol{\Sigma}_n$ is the error state covariance of the client-end odometry, and $\boldsymbol{G}_m$ and $\boldsymbol{H}_m$ are Jacobians of $\boldsymbol{p}_{j+1}$ to $\boldsymbol{p}_j$ and $\hat{\boldsymbol{p}}_{j+1j}$, respectively, which are given as,

$$\begin{aligned} \boldsymbol{G}_m &= \begin{bmatrix} 1 & \boldsymbol{0}_{1\times 3} \\ \boldsymbol{0}_{3\times 1} & \hat{\boldsymbol{R}}_{j+1j} \end{bmatrix} \\ \boldsymbol{H}_m &= \begin{bmatrix} 1 & \boldsymbol{0}_{1\times 3} \\ \frac{\partial \hat{\boldsymbol{R}}_{j+1j}\boldsymbol{t}_j}{\partial \theta_{j+1j}^{yaw}} & \boldsymbol{I}_{3\times 3} \end{bmatrix}, \end{aligned} \tag{10}$$

where $\boldsymbol{I}_{3\times 3}$ is a 3×3 identity matrix. Since the initial covariance matrix $\boldsymbol{\Sigma}_{jj}$ can be initialized to a null matrix, both $\boldsymbol{\Sigma}_{jl}$ and $\boldsymbol{\Sigma}_{ki}$ in Eq. 6 can be easily solved.

In order to obtain the final form of $\boldsymbol{J}_{jl}$, $\boldsymbol{J}_{ki}$, $\boldsymbol{\Sigma}_{jl}$ and $\boldsymbol{\Sigma}_{ki}$, some reformulations are necessary. Specifically, the rotation matrix $\boldsymbol{R}$ can be decomposed as,

$$\boldsymbol{R} = \boldsymbol{R}^z \boldsymbol{R}^y \boldsymbol{R}^x = \boldsymbol{R}^z \boldsymbol{R}^{yx}, \tag{11}$$

where $\boldsymbol{R}^x$, $\boldsymbol{R}^y$ and $\boldsymbol{R}^z$ stand for the corresponding rotation of pitch, roll and yaw, respectively. From Eq. 11, the rotation matrix $\boldsymbol{R}_{jl}$ of the relative pose from frame $\mathcal{F}_j$ to $\mathcal{F}_l$ can be reformulated as,

$$\boldsymbol{R}_{jl} = (\boldsymbol{R}_j^{yx})^T \boldsymbol{R}_{jl}^z \boldsymbol{R}_l^{yx}, \tag{12}$$

where $\boldsymbol{R}_{jl}^z$ is defined as,

$$\boldsymbol{R}_{jl}^z = \begin{bmatrix} cos(\theta_{jl}^{yaw}) & -sin(\theta_{jl}^{yaw}) & 0 \\ sin(\theta_{jl}^{yaw}) & cos(\theta_{jl}^{yaw}) & 0 \\ 0 & 0 & 1 \end{bmatrix}, \tag{13}$$

where θ_{jl}^{yaw} is the relative yaw angle from $\mathcal{F}_j$ to $\mathcal{F}_l$. Via the decomposition, we have,

$$\begin{aligned} \frac{\partial \boldsymbol{R}_p \boldsymbol{R}_{jl} \boldsymbol{R}_n \boldsymbol{t}}{\partial \theta_{jl}^{yaw}} &= \frac{\partial \boldsymbol{A} \boldsymbol{R}_{jl}^z \boldsymbol{B} \boldsymbol{t}}{\partial \theta_{jl}^{yaw}} = \begin{bmatrix} J_1 \\ J_2 \\ J_3 \end{bmatrix} \\ J_k &= \sum_{i=1}^{3} -(A_{k1}B_{i1} + A_{k2}B_{i2}) t_i sin(\theta_{jl}^{yaw}) \\ &\quad + (A_{k2}B_{i1} + A_{k1}B_{i2}) t_i cos(\theta_{jl}^{yaw}), \end{aligned} \tag{14}$$

where $\boldsymbol{R}_p$ and $\boldsymbol{R}_n$ can be any rotation matrix. By combining Eq. 6 $\sim$ 10 and Eq. 14, we can finally deduce the final form of $\boldsymbol{\Sigma}_{ijlk}$.

E. Pose Optimization

Once a new keyframe successfully triggers the loop closure, it will be sent to the back-end thread to activate our two-stage pose optimization pipeline. The pipeline aims to solve the problem of the pose graph optimization and before the establishment of the basic structure of the pose graph, the corresponding data associations need to be determined. Except for the aforementioned loop closure associations, each frame is associated to N previous frames in the same local map with the relative pose constraints computed from the original local poses yielded by the corresponding VIO, so as to copy the short-term data associations from the client. In our implementations, N is set to 5. The final optimization problem amounts to,

$$\begin{aligned} \min_{\boldsymbol{\mathcal{T}}} & \sum_{(i,j)\in\mathcal{S}} ||\boldsymbol{e}(\boldsymbol{T}_i, \boldsymbol{T}_j, \hat{\boldsymbol{T}}_{ij})||_2^2 \\ & + \sum_{(i,j)\in\mathcal{A},\mathcal{R}} \rho(||\boldsymbol{e}(\boldsymbol{T}_i, \boldsymbol{T}_j, \hat{\boldsymbol{T}}_{ij})||_2^2), \end{aligned} \tag{15}$$

where $\mathcal{S}$, $\mathcal{A}$ and $\mathcal{R}$ are the sets of short-term constraints, intra-agent constraints and inter-agent constraints, respectively, $\boldsymbol{T}_i$ ($\boldsymbol{T}_j$) stands for the pose of the keyframe $\mathcal{F}_i$ ($\mathcal{F}_j$), $\hat{\boldsymbol{T}}_{ij}$ is the constraint of the relative pose between $\mathcal{F}_i$ and $\mathcal{F}_j$, $\rho(\cdot)$ is the Huber kernel function, and $\boldsymbol{\mathcal{T}}$ represents all poses to be optimized. Taking the observability of the VIO into consideration, motivated by [22], the four-DoF (the yaw angle and the translation) error is adopted and accordingly the error term $\boldsymbol{e}(\boldsymbol{T}_i, \boldsymbol{T}_j, \hat{\boldsymbol{T}}_{ij})$ is defined as,

$$\begin{aligned} \boldsymbol{e}(\boldsymbol{T}_i, \boldsymbol{T}_j, \hat{\boldsymbol{T}}_{ij}) &= [e_{ij}^{yaw}, (\boldsymbol{e}_{ij}^{\boldsymbol{t}})^T]^T \\ \boldsymbol{e}_{ij}^{yaw} &= \theta_j^{yaw} - \theta_i^{yaw} - \hat{\theta}_{ij}^{yaw} \\ \boldsymbol{e}_{ij}^{\boldsymbol{t}} &= \boldsymbol{R}_i(\boldsymbol{t}_j - \boldsymbol{t}_i) - \hat{\boldsymbol{t}}_{ij}, \end{aligned} \tag{16}$$

where θ_i^{yaw}, θ_j^{yaw} and $\hat{\theta}_{ij}^{yaw}$ are corresponding yaw angles of poses $\boldsymbol{T}_i$, $\boldsymbol{T}_j$ and $\hat{\boldsymbol{T}}_{ij}$, respectively, $\boldsymbol{R}_i$ is the rotation matrix of $\boldsymbol{T}_i$, and $\boldsymbol{t}_i$, $\boldsymbol{t}_j$ and $\hat{\boldsymbol{t}}_{ij}$ are translation vectors of corresponding poses.

For better convergence, the problem in Eq. 15 is solved by a two-stage pipeline. Primarily, an EM-based pose smoothing is conducted to offer better initial values to the second stage optimization with an affordable time cost. Then, the objective function defined by Eq. 15 will be minimized by the LM (Levenberg-Marquardt) scheme [38] in the second stage. For the first stage, an important inequality is given as,

$$\begin{aligned} & ||\boldsymbol{e}(\boldsymbol{T}_i, \boldsymbol{T}_j, \hat{\boldsymbol{T}}_{ij})||_2^2 \\ & \quad \le \frac{1}{2}(||\boldsymbol{e}_i^k(\boldsymbol{T}_i, \hat{\boldsymbol{T}}_{ij})||_2^2 + ||\boldsymbol{e}_j^k(\boldsymbol{T}_j, \hat{\boldsymbol{T}}_{ij})||_2^2) \\ & \boldsymbol{e}_i^k(\boldsymbol{T}_i, \hat{\boldsymbol{T}}_{ij}) = [\theta_i^{yaw} + \hat{\theta}_{ij}^{yaw} - {}_E\hat{\theta}_{ij}^{yaw}, \boldsymbol{R}_i(\boldsymbol{t}_i - {}_E\hat{\boldsymbol{t}}_{ij}) - \hat{\boldsymbol{t}}_{ij}]^T \\ & \boldsymbol{e}_j^k(\boldsymbol{T}_j, \hat{\boldsymbol{T}}_{ij}) = [\theta_j^{yaw} - {}_E\hat{\theta}_{ij}^{yaw}, \boldsymbol{R}_i(\boldsymbol{t}_j - {}_E\hat{\boldsymbol{t}}_{ij})]^T, \end{aligned} \tag{17}$$

where ${}_E\hat{\theta}_{ij}^{yaw}$ and ${}_E\hat{\boldsymbol{t}}_{ij}$ can be any constant. For ease of representation, we use $\boldsymbol{e}_{ij}$ to represent $\boldsymbol{e}(\boldsymbol{T}_i, \boldsymbol{T}_j, \hat{\boldsymbol{T}}_{ij})$, and use $\boldsymbol{e}_i^k$ and $\boldsymbol{e}_j^k$ to represent $\boldsymbol{e}_i^k(\boldsymbol{T}_i, \hat{\boldsymbol{T}}_{ij})$ and $\boldsymbol{e}_j^k(\boldsymbol{T}_j, \hat{\boldsymbol{T}}_{ij})$, respectively. According to this inequality, by substituting $\boldsymbol{e}_{ij}$ to the sum of $\boldsymbol{e}_i^k$ and $\boldsymbol{e}_j^k$ and ignoring the Huber kernel, an approximated version of Eq. 15 can be obtained as,

$$\min_{\boldsymbol{\mathcal{T}}} \sum_{(i,j)\in\mathcal{D}} (||\boldsymbol{e}_i^k||_2^2 + ||\boldsymbol{e}_j^k||_2^2). \tag{18}$$

where $\mathcal{D}$ consists of all data associations and is the union of $\mathcal{S}$, $\mathcal{A}$ and $\mathcal{R}$. It can be easily proved that, ignoring the kernel function, the optimal solutions of Eq. 15 and Eq. 18 will be the same when,

$$\begin{aligned} {}_E\hat{\theta}_{ij}^{yaw} &= (\tilde{\theta}_i^{yaw} + \tilde{\theta}_j^{yaw} - \hat{\theta}_{ij}^{yaw})/2 \\ {}_E\hat{\boldsymbol{t}}_{ij} &= (\tilde{\boldsymbol{t}}_i + \tilde{\boldsymbol{t}}_j - \boldsymbol{R}_i^T \hat{\boldsymbol{t}}_{ij})/2, \end{aligned} \tag{19}$$

where $\tilde{\theta}_i^{yaw}$, $\tilde{\theta}_j^{yaw}$, $\tilde{\boldsymbol{t}}_i$ and $\tilde{\boldsymbol{t}}_j$ are all optimal solutions of Eq. 15. Since the optimal solutions are unavailable, the EM (Expectation-Maximum) framework [42] is adopted to smooth all poses iteratively. In the E-step, we utilize the current values of all frames' poses to compute ${}_E\hat{\theta}_{ij}^{yaw}$ and ${}_E\hat{\boldsymbol{t}}_{ij}$. Then in the M-step, since each error term is only related to the pose of one frame in Eq. 18, we can obtain the analytical optimal solution and update the poses efficiently. For example, in k^{th} iteration of the smoothing, the optimal yaw angle ${}_k\theta_i^{yaw}$ and position ${}_k\boldsymbol{t}_i$ are given as,

$$\begin{aligned} {}_k\theta_i^{yaw} &= Avg(\sum_{(i,j)\in\mathcal{D}} ({}_E\hat{\theta}_{ij}^{yaw} - \hat{\theta}_{ij}^{yaw}) + \sum_{(h,i)\in\mathcal{D}} {}_E\hat{\theta}_{hi}^{yaw}) \\ {}_k\boldsymbol{t}_i &= Avg(\sum_{(i,j)\in\mathcal{D}} ({}_E\hat{\boldsymbol{t}}_{ij} - \boldsymbol{R}_i^T \hat{\boldsymbol{t}}_{ij}) + \sum_{(h,i)\in\mathcal{D}} {}_E\hat{\boldsymbol{t}}_{hi}), \end{aligned} \tag{20}$$

where $Avg(*)$ stands for the average value of all terms. Once the new approximated optimal solutions ${}_k\theta_i^{yaw}$ and ${}_k\boldsymbol{t}_i$ are solved, ${}_E\hat{\theta}_{ij}^{yaw}$ and ${}_E\hat{\boldsymbol{t}}_{ij}$ can be further updated via Eq. 19 accordingly. By updating the poses of all keyframes and the approximated optimal solutions alternately, the smoothed poses will finally converge and they will be taken as the initial values of the second stage of the pipeline, the non-linear optimization.

V. Dense Mapping Module

The dense mapping module of CVIDS mainly consists of three core sub-components, the motion-stereo pipeline, the probabilistic depth filter and the global TSDF map. Based on our motion-stereo scheme, the depth information of the reference keyframe can be recovered effectively from a single pair of reference-matching frames. Then, the depth measurements of a single pixel given by different matching frames will be fused by our probabilistic filter, so as to eliminate the negative impact brought by noise and outliers. Finally, "mature" depth maps will be integrated to the global TSDF map to reconstruct the 3D structure of the scene incrementally. In this section, the specific ideas of these three sub-components will be elaborated on one by one.

A. Motion-Stereo Pipeline

Given a pair of the reference frame $\boldsymbol{I}_r$ and the matching frame $\boldsymbol{I}_m$, the target output of the motion-stereo pipeline is the depth map $\boldsymbol{D}_r$ of $\boldsymbol{I}_r$. Generally, the motion-stereo problem can be modelled as minimizing an energy function $E(\boldsymbol{D}_r)$. Except for the basic template matching loss, motivated by [21] and [23] the semi-global regularization term is also introduced to our energy function. Besides, the sparse 3D map points offered by clients can also provide high-quality supervision information to the depths of corresponding pixels, so that another "sparse prior" term is integrated. Finally, our energy function is defined as,

$$E(\boldsymbol{D}) = E_{temp}(\boldsymbol{D}) + E_{semi}(\boldsymbol{D}) + E_{sparse}(\boldsymbol{D}), \tag{21}$$

where $E_{temp}(\boldsymbol{D})$, $E_{semi}(\boldsymbol{D})$ and $E_{sparse}(\boldsymbol{D})$ are the template matching term, the semi-global regularization term and the map-point term, respectively. These three terms can be further represented as,

$$\begin{aligned} E_{temp}(\boldsymbol{D}) &= \sum_{\boldsymbol{p}} Cost[\boldsymbol{p}, D_{\boldsymbol{p}}] \\ E_{semi}(\boldsymbol{D}) &= \sum_{\boldsymbol{p}} (P_1 \sum_{\boldsymbol{q}\in\mathcal{N}(\boldsymbol{p})} T[|D_{\boldsymbol{p}} - D_{\boldsymbol{q}}| = 1] \\ &\quad + P_2 \sum_{\boldsymbol{q}\in\mathcal{N}(\boldsymbol{p})} T[|D_{\boldsymbol{p}} - D_{\boldsymbol{q}}| > 1]) \\ E_{sparse}(\boldsymbol{D}) &= P_3 \sum_{\boldsymbol{p}\in\mathcal{M}} \sum_{\boldsymbol{q}\in\mathcal{C}(\boldsymbol{p})} T[|D_{\boldsymbol{q}} - D_{\boldsymbol{p}}^{mp}| > 0], \end{aligned} \tag{22}$$

where $\mathcal{N}(\boldsymbol{p})$ is the set consisting of all neighbouring points of $\boldsymbol{p}$, $\mathcal{M}$ is the set of all sparse map points, and $\mathcal{C}(\boldsymbol{p})$ contains all points near $\boldsymbol{p}$ which are "connected" to $\boldsymbol{p}$, or more exactly share the same depth. For how to determine the connection region $\mathcal{C}(\boldsymbol{p})$, please refer to Sect. V-B. $D_{\boldsymbol{p}}$, $D_{\boldsymbol{q}}$ and $D_{\boldsymbol{p}}^{mp}$ are discretized inverse depths of $\boldsymbol{p}$, $\boldsymbol{q}$ and the sparse map point corresponding to $\boldsymbol{p}$, respectively. Taking $D_{\boldsymbol{p}}$ as an example, the relationship between $D_{\boldsymbol{p}}$ and $\boldsymbol{p}$'s depth $d_{\boldsymbol{p}}$ is given as,

$$d_{\boldsymbol{p}} = \frac{1}{D_{\boldsymbol{p}} \times D_S}, \tag{23}$$

where D_S is the constant of the searching step.

Since minimizing the energy function in Eq. 21 is actually an NP-hard problem, some approximations are necessary to prune the searching space. Specifically, according to [23], similar to the idea of the "scan line optimization", the depth of $\boldsymbol{p}$ is assumed to only be related to itself and one neighbouring pixel in direction $\boldsymbol{r}$. Then, the problem can be solved by dynamic programming efficiently,

$$\begin{aligned} L_{\boldsymbol{r}}(\boldsymbol{p}, D) = Cost[\boldsymbol{p}, D] &+ P_3 \cdot T[|D_{\boldsymbol{p}} - D_{p_n}^{mp}| > 0] \\ &+ min(L_{\boldsymbol{r}}(\boldsymbol{p}-\boldsymbol{r}, D-1), L_{\boldsymbol{r}}(\boldsymbol{p}-\boldsymbol{r}, D-1)+P_1, \\ &L_{\boldsymbol{r}}(\boldsymbol{p}-\boldsymbol{r}, D+1)+P_1, \min_i(L_{\boldsymbol{r}}(\boldsymbol{p}-\boldsymbol{r}, i))+P_2), \end{aligned} \tag{24}$$

where $\boldsymbol{p}$ is in $\boldsymbol{p}_n$'s connection region $\mathcal{C}(\boldsymbol{p}_n)$, $D_{\boldsymbol{p}_n}^{mp}$ is the depth of the sparse map point corresponding to $\boldsymbol{p}_n$, $\boldsymbol{r}$ is the direction vector, $L_{\boldsymbol{r}}(\boldsymbol{p}, D)$ is the aggregated loss. The final loss $S(\boldsymbol{D})$, which is the approximated value to $E(\boldsymbol{D})$, is given as,

$$S(\boldsymbol{D}) = \sum_{\boldsymbol{p}} \sum_{\boldsymbol{r}} L_{\boldsymbol{r}}(\boldsymbol{p}, D_{\boldsymbol{p}}). \tag{25}$$

The optimal solution of $S(\boldsymbol{D})$ can be solved efficiently through multiple times of the dynamic programming. So far, the dense 3D structure of the reference frame $\boldsymbol{I}_r$ has been recovered.

It's worth mentioning that, we utilized the geometry based motion-stereo pipeline in CVIDS mainly for engineering considerations, since such a pipeline performs relative well in both the speed and the generalization ability. It's also easy to replace such a pipeline with any other depth estimation scheme according to the users' own requirements, such as the learning-based monocular depth estimation network, and such replacement won't change the basic operation mechanism of CVIDS.

B. Connection Region Determination

The connection region $\mathcal{C}(\boldsymbol{p})$ of point $\boldsymbol{p}$ contains all points near $\boldsymbol{p}$ which share the same depth with $\boldsymbol{p}$, or more specifically, which are in the same plane with $\boldsymbol{p}$. In CVIDS, to determine the region, we adopted a gradient-based heuristic policy, which consists of two criteria:

1) The point in $\mathcal{C}(\boldsymbol{p})$ should be in a 10×10 local window whose center is $\boldsymbol{p}$.
2) On the path between any point in $\mathcal{C}(\boldsymbol{p})$ and $\boldsymbol{p}$, there should be no pixels with the gradient moduli larger than the threshold.

To determine the connection regions, we firstly detect those pixels with large gradient moduli with the Sobel operator. For ease of representation, we call these pixels as "boundary pixels", and for a boundary pixel $\boldsymbol{p}_b$ on image $\boldsymbol{I}_b$, it must satisfy,

$$G(\boldsymbol{p}_b) > G_b^{mean} + \sigma_b, \tag{26}$$

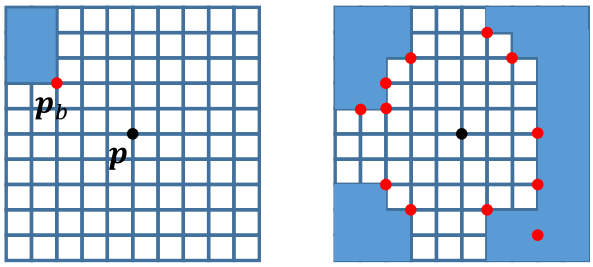


Fig. 3. Illustration of the connection region. The pixels in the 10×10 window but not belonging to the connection region are marked in blue, while the rest pixels constitute the connection region.

where $G(\boldsymbol{p}_b)$ is the gradient modulus of $\boldsymbol{p}_b$, G_b^{mean} is the mean gradient modulus of $\boldsymbol{I}_b$ and σ_b is the corresponding standard deviation. Then, for each boundary pixel $\boldsymbol{p}_b$, as Fig. 3 illustrated, we remove some corresponding "sheltered" pixels from the connection region in the local 10×10 window. After all boundary pixels' corresponding removed pixels are culling, the rest pixels in the local window constitute the connection region. For any removed pixel $\boldsymbol{p}_r$, it must satisfy,

$$\begin{aligned}([\boldsymbol{p}_r-\boldsymbol{p}]_x)^2 <= [\boldsymbol{p}_b-\boldsymbol{p}]_x\cdot[\boldsymbol{p}_r-\boldsymbol{p}]_x\\ ([\boldsymbol{p}_r-\boldsymbol{p}]_y)^2 <= [\boldsymbol{p}_b-\boldsymbol{p}]_y\cdot[\boldsymbol{p}_r-\boldsymbol{p}]_y,\end{aligned}\tag{27}$$

where $[*]_x$ and $[*]_y$ represent the coordinates in x and y axes of the inner point, respectively.

C. Probabilistic Depth Filter

As discussed in Sect. V-A, for a pixel $\boldsymbol{p}$ on the reference frame $\boldsymbol{I}_r$, its inverse depth estimation $d_{\boldsymbol{p}}^k$ from the matching frame $\boldsymbol{I}_k$ can be yielded by our motion-stereo pipeline. Based on the compact representation proposed in [19], the likelihood distribution of $\boldsymbol{p}$'s true inverse depth $\tilde{d}_{\boldsymbol{p}}$ can be modeled as a mixture of a normal distribution (effective measurement) and a uniform one (outlier measurement), which is represented as,

$$\begin{aligned}p(d_{\boldsymbol{p}}^k|\tilde{d}_{\boldsymbol{p}},\rho_{\boldsymbol{p}}) = \rho_{\boldsymbol{p}}\mathcal{N}(d_{\boldsymbol{p}}^k|\tilde{d}_{\boldsymbol{p}},(\tau_{\boldsymbol{p}}{}^k)^2)\\ +(1-\rho_{\boldsymbol{p}})\mathcal{U}(d_{\boldsymbol{p}}^k|d_{min},d_{max}),\end{aligned}\tag{28}$$

where $\tau_{\boldsymbol{p}}{}^k$ is the variance of the observation model, d_{min} and d_{max} stand for the minimum and maximum possible inverse depth of captured pixels, respectively, and the ratio $\rho_{\boldsymbol{p}}$ is the probability of getting an effective measurement at $\boldsymbol{p}$. Actually, for inlier observations, the measurement $d_{\boldsymbol{p}}^k$ will be predominantly affected by a white Gaussian noise, and accordingly, the expectation $\tilde{d}_{\boldsymbol{p}}$ is set to $d_{\boldsymbol{p}}^k$ and the variance $\tau_{\boldsymbol{p}}{}^k$ is set to $2D_S$.

Assuming independent observations, given a sequence of inverse depth estimations $d_{\boldsymbol{p}}^k, d_{\boldsymbol{p}}^{k+1},\ldots,d_{\boldsymbol{p}}^{k+r}$ of $\boldsymbol{p}$ from corresponding matching frames, the posteriori of $\boldsymbol{p}$'s inverse depth and the inlier probability $\rho_{\boldsymbol{p}}$ is given as,

$$p(\tilde{d}_{\boldsymbol{p}},\rho_{\boldsymbol{p}}|d_{\boldsymbol{p}}^k,\ldots,d_{\boldsymbol{p}}^{k+r}) \propto p(\tilde{d}_{\boldsymbol{p}},\rho_{\boldsymbol{p}})\prod_{n=0}^{r}p(d_{\boldsymbol{p}}^{k+n}|\tilde{d}_{\boldsymbol{p}},\rho_{\boldsymbol{p}}),\tag{29}$$

where $p(\tilde{d}_{\boldsymbol{p}},\rho_{\boldsymbol{p}})$ is the prior distribution. For ease of representation, the posteriori $p(\tilde{d}_{\boldsymbol{p}},\rho_{\boldsymbol{p}}|d_{\boldsymbol{p}}^k,d_{\boldsymbol{p}}^{k+1},\ldots,d_{\boldsymbol{p}}^{k+r})$ is represented as $p_{k+r}(\tilde{d}_{\boldsymbol{p}},\rho_{\boldsymbol{p}})$. Then the recurrence relationship of the posteriori can be expressed as,

$$p_n(\tilde{d}_{\boldsymbol{p}},\rho_{\boldsymbol{p}}) \propto p_{n-1}(\tilde{d}_{\boldsymbol{p}},\rho_{\boldsymbol{p}})p(d_{\boldsymbol{p}}^n|\tilde{d},\rho_{\boldsymbol{p}}).\tag{30}$$

To keep the forms of the distribution before and after the update uniform, the posteriori in Eq. 30 is further approximated by the product of a Beta distribution and a normal one, which is formulated as,

$$p_n(\tilde{d}_{\boldsymbol{p}},\rho_{\boldsymbol{p}}) \approx q(\tilde{d}_{\boldsymbol{p}},\rho_{\boldsymbol{p}}|a_{\boldsymbol{p}}^n,b_{\boldsymbol{p}}^n,\mu_{\boldsymbol{p}}^n,\sigma_{\boldsymbol{p}}^n) = Beta_{\boldsymbol{p}}^n\mathcal{N}_{\boldsymbol{p}}^n,\tag{31}$$

where $a_{\boldsymbol{p}}^n$ and $b_{\boldsymbol{p}}^n$ controls the Beta distribution, and $\mu_{\boldsymbol{p}}^n$ and $\sigma_{\boldsymbol{p}}^n$ are the expectation and the variance of $\mathcal{N}_{\boldsymbol{p}}^n$, respectively. For each time a new observation is received, the parameters in the posteriori, including $a_{\boldsymbol{p}}^n$, $b_{\boldsymbol{p}}^n$, $\mu_{\boldsymbol{p}}^n$ and $\sigma_{\boldsymbol{p}}^n$, will alter, but the form of the distribution remains, which allows to update the posteriori incrementally.

Once the parallax between the reference frame and the current frame exceeds the preset threshold, the old reference frame will be set to be "matured" and its depth information will be propagated to the current frame. Then, the current frame becomes the new reference frame. It's worth mentioning that, we found the propagation of the "Beta" component usually diverges the depth distribution, since the relative pose won't be absolutely accurate. Thus, the propagation is only conducted on the "normal" component. For a point $\boldsymbol{p}_r$ with the inverse depth d_r on the reference frame $\boldsymbol{I}_r$, as the inverse depth distribution of $\boldsymbol{p}_r$ is propagated to the corresponding point $\boldsymbol{p}_c$ on the current frame $\boldsymbol{I}_c$, the prior distribution of $\boldsymbol{p}_c$'s inverse depth is set to,

$$p(d_c,\rho|d_r) = Beta_{\boldsymbol{p}}^n\cdot\mathcal{N}(d_c|\frac{\partial d_c}{\partial d_r}\cdot\mu_r,(\frac{\partial d_c}{\partial d_r}\cdot\sigma_r)^2),\tag{32}$$

and the derivative of d_c to d_r is given as,

$$\frac{\partial d_c}{\partial d_r} = -[\boldsymbol{T}_{cr}]_{33}\cdot\frac{1}{d_r^2},\tag{33}$$

where $\boldsymbol{T}_{cr}$ is the relative pose between $\boldsymbol{I}_r$ and $\boldsymbol{I}_c$, and $[*]_{ij}$ represents the element in the i^{th} row and the j^{th} column of the inner matrix.

The matured depth maps are then integrated into the global map. Rather than utilizing all pixels, some outliers need to be eliminated first. Since a and b control the Beta distribution $Beta_{\boldsymbol{p}}^n = Beta(\rho|a,b)$, the ratio ρ, which reflects how confident the depth estimation is an inlier, can be given as,

$$\rho = \frac{a}{a+b}.\tag{34}$$

For a point $\boldsymbol{p}$ on the "matured" frame, only if $\boldsymbol{p}$'s corresponding ρ is larger than the threshold, which is set to 0.5 in our implementations, it can be considered to be an inlier and be fused to the global dense map. Besides, it's worth mentioning that rather than directly be fused into the global dense map, the depth maps of "matured" frames will be temporarily stored in a queue, since its global pose may still be unstable. Until the global pose of the frame maintained stable and alters little in the last two times of the global optimization, the depth map of such a frame will be integrated into the global TSDF map.

D. TSDF Fusion

Our implementations in this sub-component mostly follow the work in [39] while some necessary modifications were made. For each inlier pixel $\boldsymbol{p}_c$ on the "matured" frame $\boldsymbol{I}_c$, since its inverse depth and the pose of $\boldsymbol{I}_c$ are known, a ray emitting from the sensor origin to the scene will be cast and both the signed distance and the weight of each voxel in the hit region will be updated. The length of the hit region τ_c is determined by the variance $\sigma^d_{\boldsymbol{p}_c}$ of $\boldsymbol{p}_c$'s depth. Though the corresponding variance $\sigma_{\boldsymbol{p}_c}$ of $\boldsymbol{p}_c$'s inverse depth d_c has already been deduced, we found it's usually "over confident". Thus, instead of utilizing the result yielded by the depth filter, $\sigma_{\boldsymbol{p}_c}$ is set to the variance of a single measurement, $2D_S$, and then $\sigma^d_{\boldsymbol{p}_c}$ and τ_c can be given as,

$$\tau_c = 2\sigma^d_{\boldsymbol{p}_c} = \frac{2\sigma_{\boldsymbol{p}_c}}{(d_c)^2}. \tag{35}$$

For each voxel $\boldsymbol{v}_c$ in the hit region of $\boldsymbol{p}_c$, its signed distance $\Phi_r(\boldsymbol{v}_c)$ and the weight $W(\boldsymbol{v}_c)$ are updated as,

$$\begin{aligned} \Phi_r(\boldsymbol{v}_c) &\leftarrow \frac{W(\boldsymbol{v}_c)\Phi_r(\boldsymbol{v}_c) + \frac{1}{\tau_c}u}{W(\boldsymbol{v}_c) + \frac{1}{\tau_c}} \\ W(\boldsymbol{v}_c) &\leftarrow W(\boldsymbol{v}_c) + \frac{1}{\tau_c}, \end{aligned} \tag{36}$$

where u is the corresponding signed distance of the measurement. The weights and signed distances of all voxels will be updated incrementally, and finally, through the meshing, the global dense map can be constructed.

VI. Experimental Results

A. Evaluation Metrics, Benchmark Datasets and Hardware Architectures

The evaluation of CVIDS mainly focuses on two aspects, the localization and the mapping. In terms of the localization, we mainly tested CVIDS and its counterparts on the Euroc dataset [40], and the RMSE (Root Mean Squared Error) [43] is adopted as the metric to measure the accuracy, which is evaluated by the ATE (Absolute Trajectory Error) and can be given as,

$$e_{RMSE} = \left(\frac{1}{M}\sum_{i=1}^{M} ||trans(\mathbf{Q}_i^{-1}\mathbf{P}_i)||^2\right)^{\frac{1}{2}}, \mathbf{Q}_i, \mathbf{P}_i \in SE(3), \tag{37}$$

where $\{\mathbf{Q}_i\}_{i=1}^M$ and $\{\mathbf{P}_i\}_{i=1}^M$ are groundtruth and estimated poses of all frames, respectively. The $trans(.)$ represents the translation part of the pose. M is the total number of frames.

For the mapping aspect, on the one hand, we displayed the final mapping results of CVIDS over multiple sequences of images and inertial data collected by us in handheld manners using the Realsense D435i camera suites, so as to qualitatively corroborate the effectiveness of our scheme. The camera in each suite is a global shutter RGB camera produced by Intel and the IMU is the Bosch BMI055 six-axes IMU. The images captured will be firstly preprocessed by the Intel® RealSense™ Vision Processor D4. The resolution, frame rate and FoV of the camera are 1920×1080, 30 fps and 69.4×42.5, respectively.

TABLE I
QUALITATIVE COMPARISON WITH RELATED METHODS

Method	Sensors Configuration	Multi-agent	Map Density
Mohanarajah *et al.*'s	RGBD camera	√	Dense
Golodetz *et al.*'s	RGBD camera	√	Dense
Dong *et al.*'s	RGBD camera	√	Dense
Forster *et al.*'s	Monocular camera	√	Sparse
CoSLAM	Monocular camera	√	Sparse
Deutsch *et al.*'s	Monocular camera	√	Sparse
CCM-SLAM	Monocular camera	√	Sparse
CVI-SLAM	Monocular camera + IMU	√	Sparse
MonoFusion	Monocular camera	×	Dense
LSD-SLAM	Monocular camera	×	Semi-dense
Remode	Monocular camera	×	Dense
VI-MEAN	Monocular camera + IMU	×	Dense
CVIDS	Monocular camera + IMU	√	Dense

On the other hand, the accuracy of the meshed dense map is difficult to be measured quantitatively, while it is usually directly related to the recovered depth structure of each single frame. Thus, we utilized the average depth error of the recovered depth map as the metric in mapping, and conducted corresponding quantitative evaluations over the dataset proposed in [41].

For the hardware architecture of the central server, we built the server on a workstation with an Intel Xeon(R) CPU E5-2678 V3 processor and a TITAN RTX GPU.

B. Qualitative Experimental Results

1) Traits of Methods: From those three aspects shown in Table I, we compared all methods discussed in Sect. II and also our CVIDS to demonstrate their characteristics more clearly. 1) What kind of sensor configurations does the method utilize? 2) Can it be applicable to the multi-agent system? 3) How dense maps can be constructed by the scheme? It can be seen that among all compared schemes, our CVIDS is the only one which can both be applicable to the multi-agent system and be able to construct the dense map under a configuration of the monocular suite without the depth sensor, implying that CVIDS is blessed with stronger environmental adaptability compared with those RGBD oriented schemes and single-agent ones.

2) Typical Samples of Recovered Depth Maps: To qualitatively demonstrate the superiority of our proposed depth estimation pipeline in CVIDS, typical samples were selected from both the "over table" and the "fast motion" sequences in the dataset proposed in [41], and the yielded depth maps of CVIDS and its competitors, including ReMode [18] and VI-MEAN [21], are shown in Fig. 4. From Fig. 4, it can be seen that ReMode [18] performed relatively unsatisfactorily in weakly textured regions, such as the printer surfaces and the computer screen, and in the results of VI-MEAN [21], obvious outliers existed. By contrast, CVIDS obviously outperformed its two competitors, which qualitatively reflected its remarkable performance in mapping.

3) Reconstructed Dense Maps: We collected the data in three different indoor or outdoor scenes with the Realsense D435i camera suite in handheld manners, and then fed the images and the inertial data to CVIDS, so as to justify the effectiveness of CVIDS on online reconstructions. The final

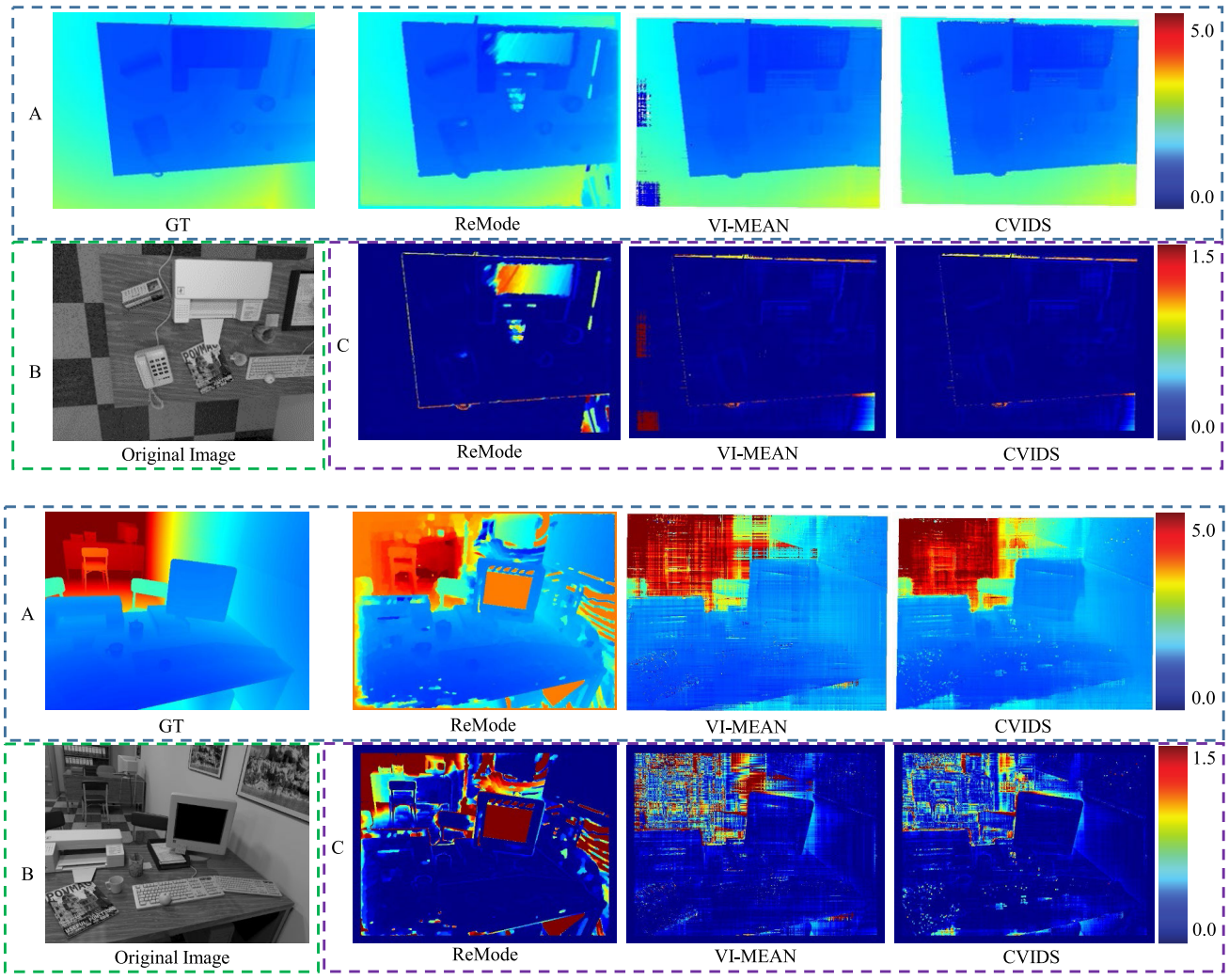


Fig. 4. Typical samples of recovered depth maps. Results provided in the upper group are from the "over table" sequence, while the bottom group corresponds to the "fast motion" sequence. In each group of the results, the recovered depth maps, the original image and the error maps are shown in regions A, B and C, respectively.

reconstruction results are displayed in Fig. 5, in which from top to bottom, the results of the indoor single-agent mapping, the indoor multi-agent mapping and the outdoor multi-agent mapping are offered, respectively. From the experimental results, it can be seen that our scheme can yield maps of the surrounding environment in high density and quality.

C. Quantitative Experimental Results

1) Collaborative Localization Accuracy: To evaluate the collaborative localization accuracy of CVIDS, we ran it under the multi-agent configuration on the Euroc machine hall (MH) dataset [40], and each sequence in the dataset (from MH_01 to MH_05) was fed to a single agent. Then, the yielded trajectories of all agents were recorded and the corresponding RMSEs were computed. Table II shows the quantitative comparison of RMSEs obtained by CVIDS and its main counterparts, including OKVIS [28], VINS-Mono [22], VIORB [29] and CVI-SLAM [30]. From Table II, it can be seen that CVIDS exhibits the lowest average RMSE of all sequences, confirming its superiority for localization. It's worth mentioning that, in the experiment, since the odometries we utilized in the client-end were the non-loop version of VINS-Mono [22], which performed unsatisfactory on the MH_05 sequence, the performance of CVIDS on the fifth sequence was also relatively inferior to that of CVI-SLAM and VIORB. However, on MH_05, CVIDS performs obviously stronger than VINS-Mono, which corroborates the significant performance gain that our framework brings to the single agent.

TABLE II
RMSEs OF COMPARED VISUAL-INERTIAL LOCALIZATION SCHEMES ON EUROC MH DATASET (*cm*)

	OKVIS	VINS-Mono	VIORB	CVI-SLAM	**CVIDS**
MH_01	33.00	12.00	7.50	8.50	**3.86**
MH_02	37.00	12.00	8.40	6.30	**3.64**
MH_03	25.00	13.00	8.70	**6.50**	7.01
MH_04	27.00	18.00	21.70	29.30	**12.92**
MH_05	39.00	21.00	8.20	**8.10**	14.53
Weighted Average	31.33	14.40	10.39	10.91	**7.48**

2) Effectiveness of Loop Closure Detection Strategy: To evaluate the superiority of our LCD strategy, we ran CVIDS

Fig. 5. Typical samples of reconstructed dense maps. From (a) to (c), the results correspond to the single-agent indoor mapping, the multi-agent indoor mapping and the multi-agent outdoor mapping, respectively. And in each group, the yielded dense map of CVIDS is shown on the left, while the snapshot of the real scene is on the right.

under the multi-agent mode on the Euroc MH dataset [40]. For comparison, two other baselines were also tested, which are 1) VINS-VIDS: The LCD module of VINS-Mono [22] is utilized to substitute that in CVIDS; 2) ORB-VIDS: The LCD module of VIORB [29] is adopted. The comparison are mainly in terms of two aspects: the speed and the accuracy.

On the speed aspect, the time cost of CVIDS to finish the LCD task for one frame is about 14.22ms, and the costs of VINS-VIDS and ORB-VIDS are 13.99ms and 16.83ms, respectively. From the results, it can be seen that the speed performance of the LCD modules in VINS-VIDS and CVIDS are almost the same, while both of them are faster than ORB-VIDS, corroborating the efficiency of our proposed LCD strategy. It's worth mentioning that since the pairwise consistency evaluation is running in the backend optimization thread rather than the main one, it is not considered in the comparison.

On the accuracy aspect, the RMSEs of all compared schemes are summarized in Table III. Thanks to our proposed view-frustum based common-view judgement and four-DoF pairwise consistency evaluation mechanism, CVIDS obviously outperforms other two competitors. Thus, we can say our LCD strategy shows excellent performance in both the speed and the accuracy.

3) Depth Recovery Accuracy: Based on the dataset presented in [41], we quantitatively evaluated the accuracy of the dense mapping module of CVIDS by the average depth errors of the recovered depth maps. As main competitors, the performance of ReMode [18] and VI-MEAN [21] were also tested. In the experiment, for each reference frame, its following multiple frames were considered as matching frames to recover the depth structure. Then, for each compared method, the average depth errors of all pixels were further computed. It's worth mentioning, for both VI-MEAN and CVIDS, five reference frames were utilized, while thirty frames are utilized in the evaluation of ReMode for its depth filter to converge. The experimental results were summarized in Table IV. From Table IV, it can be seen that CVIDS shows the overwhelming superiority with respect to depth estimation, implying its distinguished mapping performance.

4) Time Cost Analysis: The average time cost of each component in the main thread of CVIDS is offered in Table V. It can be seen that by means of the acceleration of the GPU, for a single frame, CVIDS can complete both the localization and the depth estimation in about 43ms, implying that CVIDS achieves a frame rate of more than 20 fps and such a speed performance satisfies the real-time requirements in most cases.

Since the pose graph optimization of CVIDS runs in the background thread rather than the main thread, the relevant time costs are not provided in Table V. Instead, we analyzed its speed performance separately. Our optimization pipeline consists of two stages, EM-based pose smoothing and the nonlinear optimization, and the speed performance of both stages is directly related to the number of frames involved in the optimization. To evaluate the speed performance of CVIDS more comprehensively, we recorded the time consumption of each stage when using different numbers of frames, and present the results in Fig. 6. From the result, it can be seen that even if there are up to 3,000 involved frames, the global optimization can still be completed within 2.5s, implying an outstanding efficiency of our optimization pipeline.

TABLE III

RMSEs OF OUR SYSTEMS UTILIZING DIFFERENT LCD SOLUTIONS ON EUROC MH DATASET (cm)

	ORB-VIDS	VINS-VIDS	**CVIDS**
MH_01	7.87	6.82	**3.86**
MH_02	10.36	6.88	**3.64**
MH_03	12.14	7.31	**7.01**
MH_04	16.40	16.29	**12.92**
MH_05	20.87	21.35	**14.53**
Weighted Average	12.53	10.36	**7.48**

TABLE IV

AVERAGE DEPTH MAP ERRORS OF COMPARED SCHEMES ON THE DATASET [41] (cm)

	ReMode	VI-MEAN	**CVIDS**
Over Table	3.64	5.03	**2.21**
Fast Motion	40.50	15.71	**9.10**
Slow Motion	108.11	27.98	**11.08**

TABLE V

TIME COST ANALYSIS IN THE MAIN THREAD OF CVIDS (ms/f)

	Localization	Motion-stereo	Depth Filtering	Sum
Processor	CPU	GPU	GPU	-
Cost	22.44	11.79	8.91	43.14

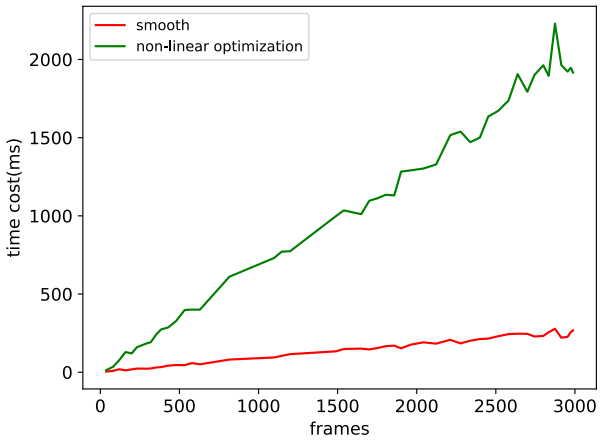


Fig. 6. Time costs of the two stages of our optimization pipeline with the evolvement of the number of involved frames.

TABLE VI
TRACKING TIME COSTS IN THE MAIN THREADS OF COMPARED SLAM SYSTEMS (ms/f)

	Forster *et al.*	CoSLAM	CCM-SLAM	CVI-SLAM	CVIDS
Processor	CPU	GPU	CPU	CPU	CPU
Cost	33.50	**19.40**	26.36	36.1	22.44

Besides, we also summarized the speed performance of other collaborative SLAM systems and compared them with CVIDS, including Forster et al.'s work [24], CoSLAM [25], CCM-SLAM [27] and CVI-SLAM [30]. Detailed experimental results are summarized in Table VI. It's worth mentioning that since most of the existing collaborative SLAM frameworks don't support the monocular dense mapping, we just compare the time cost of the localization. From Table VI, it can be seen that the speed performance of CVIDS is the second only to CoSLAM, which is accelerated by GPU. For the consideration of the localization accuracy and the support to monocular dense mapping, we say that CVIDS performs well in the speed while ensuring accuracy.

D. Ablation Study

1) Ablation Study for the LCD: As aforementioned, the LCD in CVIDS is mainly conducted in two manners, the BoW-based LCD and the view-frustum-based one. The BoW-based LCD guarantees the map alignment and the optimization can be accomplished on the rails, while the view-frustum-based one significantly enhances the accuracy of CVIDS by fully exploiting the advantages of the multi-agent system. Besides, the pairwise consistent evaluation is also adopted to cull outlier loop closures. To verify our claims, we try to justify the effectiveness of these three components, respectively. It's worth mentioning that without the BoW-based LCD, different local maps can't be aligned, and thus the view-frustum-based LCD won't be conducted. Hence, we mainly compared CVIDS with three baselines, which were 1) DV-VIDS: The view-frustum-based LCD was deactivated; 2) DL-VIDS: Two types of LCD were both deactivated; and 3) DP-VIDS: The pairwise consistent evaluation was deactivated. CVIDS and these three baselines were evaluated on the five sequences of the Euroc MH dataset [40], and the obtained quantitative experimental results are summarized in Table VII. From Table VII, it can be seen that CVIDS significantly overperforms the other three baselines in terms of the localization accuracy, corroborating the effectiveness of our LCD strategy.

TABLE VII
RMSES OF OUR SCHEME ON EUROC MH DATASET UNDER DIFFERENT LCD CONFIGURATIONS (cm)

	DV-VIDS	DL-VIDS	DP-VIDS	**CVIDS**
MH_01	6.57	15.41	4.02	**3.86**
MH_02	6.82	17.18	3.84	**3.64**
MH_03	7.21	19.62	7.17	**7.01**
MH_04	16.22	26.78	13.01	**12.92**
MH_05	21.18	28.89	**14.53**	**14.53**
Weighted Average	10.22	20.45	7.65	**7.48**

It's worth mentioning that, the reason why CVIDS performs only slightly better than DP-VIDS is that, in general cases, our loop closure detection strategy has already guaranteed the correctness of loop closure measurements on the Euroc MH dataset even without the pairwise consistent evaluation. Thus, we also further introduce outlier loop closure measurements manually in order to evaluate the performance gain brought by the pairwise consistent evaluation more comprehensively. For more details, please refer to the next paragraph.

2) Robustness Gain from Pairwise Consistency Evaluation: Before the pose optimization, the pairwise consistency evaluation will be conducted so as to cull those failure measurements of loop closures. Since in general there won't be quantities of outlier loop closures, so as to evaluate the performance gain brought by this module more comprehensively, we test our system on the Euroc machine hall dataset and manually add disturbance to the relative pose of the loop closure measurements. For comparison, we took DP-VIDS, in which the pairwise consistent evaluation was deactivated, as a baseline. Finally, We recorded the corresponding RMSEs of the compared schemes under different disturbance settings.

Actually, in reality, there are mainly two types of outlier loop closure measurements: a) The recovered relative poses are not accurate due to the influence of illumination, motion and other relevant factors; b) Frames are incorrectly matched in the process of the loop closure detection, resulting in loop closure measurements that should not exist. To simulate these two cases, two types of experimental settings were adopted. In the first type of setting, we added different levels of disturbances to all loop closure measurements to simulate case a). It's worth mentioning that, the unit disturbance is equivalent to two centimeters of translation in three orthogonal directions and two degrees of rotation corresponding to the yaw angle. Relevant experimental results are summarized in Table VIII. In the second type of setting, to simulate case b), we replaced a certain proportion of loop closure measurements with outlier observations generated between randomly selected frames, and evaluated the localization performance of CVIDS and DP-VIDS under different proportion settings. The experimental results are summarized in Table IX. From Table VIII and IX, it can be seen that our pairwise consistency evaluation module

TABLE VIII

RMSEs OF OUR SCHEME ON EUROC MH DATASET UNDER DIFFERENT LOOP CLOSURE DISTURBANCE CONFIGURATIONS (*cm*)

Sequence	Scheme \ Disturbance	0	1	2	3	4	5	6	7	8	9	10
MH_01	**CVIDS**	3.86	4.77	6.56	8.56	10.79	13.15	15.10	17.67	19.32	21.40	23.03
	DP-VIDS	3.99	4.67	6.77	8.56	11.54	13.74	15.82	17.78	19.99	22.04	24.08
MH_02	**CVIDS**	3.64	4.27	5.15	6.07	7.15	8.3	9.39	10.09	11.42	12.4	13.19
	DP-VIDS	3.76	4.38	5.61	6.66	7.93	9.31	10.35	11.37	12.51	13.41	14.23
MH_03	**CVIDS**	7.01	7.44	7.81	8.25	8.83	9.54	1.02	10.74	12.06	12.67	13.46
	DP-VIDS	7.16	7.1	7.48	8.03	8.69	9.45	1.03	11.24	11.9	12.61	13.47
MH_04	**CVIDS**	12.92	13.21	13.16	13.17	13.32	13.49	14.16	14.46	14.79	15.35	15.57
	DP-VIDS	13.0	12.74	12.95	13.08	13.32	13.78	14.21	14.88	15.69	16.1	16.86
MH_05	**CVIDS**	14.53	14.6	14.69	14.93	15.17	15.63	16.19	16.86	17.48	18.17	18.88
	DP-VIDS	14.55	14.55	14.48	14.7	14.97	15.37	15.9	16.6	17.32	18.1	19.4
Weighted Average	**CVIDS**	**7.48**	8.01	8.73	**9.56**	**10.54**	**11.64**	**10.22**	**13.76**	**14.92**	**15.97**	**16.88**
	DP-VIDS	7.59	**7.82**	**8.72**	9.57	10.82	11.97	10.56	14.2	15.38	16.42	17.62

TABLE IX

RMSEs OF OUR SCHEMES ON EUROC MH DATASET SUFFERING DIFFERENT NUMBERS OF OUTLIER LOOP CLOSURE MEASUREMENTS (*cm*)

Sequence	Scheme \ Outlier Proportion	0%	3%	5%	10%	20%
MH_01	**CVIDS**	3.86	4.71	4.57	4.85	34.63
	DP-VIDS	3.99	39.82	60.63	44.79	136.49
MH_02	**CVIDS**	3.64	4.63	4.45	4.52	9.50
	DP-VIDS	3.76	34.53	40.15	35.82	75.77
MH_03	**CVIDS**	7.01	11.54	11.21	10.68	16.12
	DP-VIDS	7.16	22.23	22.30	48.02	107.60
MH_04	**CVIDS**	12.92	20.82	19.47	14.00	49.69
	DP-VIDS	13.00	18.03	28.11	42.97	49.38
MH_05	**CVIDS**	14.53	18.35	15.06	14.33	22.50
	DP-VIDS	14.55	26.13	26.25	22.22	60.49
Weighted Average	**CVIDS**	**7.48**	**10.92**	**10.12**	**9.08**	**25.98**
	DP-VIDS	7.59	28.78	36.83	40.89	93.79

can make substantial improvement on the localization accuracy almost under all disturbance settings, implying its significance for the robustness of CVIDS.

3) Ablation Study for the Localization: As aforementioned, CVIDS supports the collaborative localization and mapping under the multi-agent frameworks. Compared with the single-agent ones, the perception of the multi-agent systems is much stronger. To quantitatively corroborate our claims, we performed detailed ablation analysis. Over the Euroc MH dataset [40], we ran CVIDS in both the multi-agent and the single-agent modes, and recorded the corresponding RMSEs over each sequence of the data, respectively. Besides, the performance gain in the localization accuracy brought by the four-DoF based optimization and our pose smoothing module were also evaluated. We demonstrate how different components in our framework affect the localization accuracy by comparing CVIDS with five baselines. Those baselines were 1) SA-V: CVIDS ran in the single-agent mode; 2) 6D-V: The six-DoF based optimization is utilized instead of the four-DoF based one; 3) DS-V: The EM-based smoothing in the optimization pipeline was deactivated; 4) DN-V: The second stage of the optimization pipeline was deactivated; and 5) DO-V: The pose optimization pipeline in the back-end were thoroughly abandoned. The quantitative experimental results were given in Table X. From the results, it can be seen that both the multi-agent configuration and each stage of our optimization pipeline are indispensable to guarantee the localization accuracy of CVIDS.

TABLE X

RMSEs OF OUR SCHEME ON EUROC MH DATASET UNDER DIFFERENT LOCALIZATION CONFIGURATIONS (*cm*)

	SA-V	6D-V	DS-V	DN-V	DO-V	**CVIDS**
MH_01	7.48	22.85	5.52	18.68	15.43	**3.86**
MH_02	9.40	16.98	5.24	14.79	17.23	**3.64**
MH_03	10.43	30.15	8.17	17.49	19.59	**7.01**
MH_04	17.66	28.97	14.42	25.20	26.79	**12.92**
MH_05	19.49	37.20	**13.22**	22.31	28.88	14.53
Weighted Average	11.82	26.60	8.62	19.18	20.45	**7.48**

TABLE XI

AVERAGE DEPTH MAP ERRORS OF OUR SCHEME ON THE DATASET [41] UNDER DIFFERENT STEREO-MOTION CONFIGURATIONS (*cm*)

	WS-VIDS	WM-VIDS	**CVIDS**
Over Table	6.73	2.42	**2.21**
Fast Motion	33.58	9.88	**9.10**
Slow Motion	67.65	13.27	**11.08**

4) Ablation Study for the Depth Estimation: We aim to evaluate the performance gain brought by different terms of the energy function $E(\boldsymbol{D})$ defined in Eq. 21 which we utilized in our stereo-motion pipeline. Thus, the performance of CVIDS in terms of depth estimation was compared with other two baselines on the dataset proposed in [41]. The two compared baselines were 1) WS-VIDS: Without the semi-global regularization term; and 2) WM-VIDS: Without the map-point term.

Experimental results are summarized in Table XI. From the table, it can be seen that the motion-stereo pipeline of CVIDS under our current energy settings shows obviously superior performance than the other two baselines, implying the indispensability of each of the aforementioned energy terms.

VII. Conclusion

In this paper, we studied a practical problem, collaborative localization and mapping for the multi-agent systems only by monocular camera suites, and proposed a novel collaborative dense SLAM framework, namely CVIDS. It follows a loosely coupled and centralized architecture, and can be integrated with any existing visual-inertial odometry to co-localize multi-agents efficiently via our proposed robust loop closure detection module and the two-stage pose optimization pipeline. Furthermore, based on the accurate poses in a unified reference coordinate system of all keyframes, CVIDS can reconstruct the scene densely. One eminent feature of CVIDS is that it does not rely on any depth sensor, but utilizes the motion-stereo pipeline we proposed to recover the depth structure of the images collected by monocular cameras, which brings low hardware costs and a wide scope of applications. The experimental results corroborate the superior performance of CVIDS.

References

[1] J.-C. Piao and S.-D. Kim, "Real-time visual–inertial SLAM based on adaptive keyframe selection for mobile AR applications," *IEEE Trans. Multimedia*, vol. 21, no. 11, pp. 2827–2836, Nov. 2019.

[2] H. Liu, G. Zhang, and H. Bao, "Robust keyframe-based monocular SLAM for augmented reality," in *Proc. IEEE Int. Symp. Mixed Augmented Reality (ISMAR)*, Sep. 2016, pp. 340–341.

[3] D. Ramadasan, M. Chevaldonne, and T. Chateau, "Real-time SLAM for static multi-objects learning and tracking applied to augmented reality applications," in *Proc. IEEE Virtual Reality*, Mar. 2015, pp. 267–268.

[4] X. Du and K. K. Tan, "Comprehensive and practical vision system for self-driving vehicle lane-level localization," *IEEE Trans. Image Process.*, vol. 25, no. 5, pp. 2075–2088, May 2016.

[5] Y. Cai, L. Dai, H. Wang, and Z. Li, "Multi-target pan-class intrinsic relevance driven model for improving semantic segmentation in autonomous driving," *IEEE Trans. Image Process.*, vol. 30, pp. 9069–9084, 2021.

[6] L. Zhang, J. Huang, X. Li, and L. Xiong, "Vision-based parking-slot detection: A DCNN-based approach and a large-scale benchmark dataset," *IEEE Trans. Image Process.*, vol. 27, no. 11, pp. 5350–5364, Nov. 2018.

[7] G. Klein and D. Murray, "Parallel tracking and mapping for small AR workspaces," in *Proc. 6th IEEE ACM Int. Symp. Mixed Augmented Reality*, Nov. 2007, pp. 225–234.

[8] J. Engel, V. Koltun, and D. Cremers, "Direct sparse odometry," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 3, pp. 611–625, Mar. 2016.

[9] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardós, "ORB-SLAM: A versatile and accurate monocular SLAM system," *IEEE Trans. Robot.*, vol. 31, no. 5, pp. 1147–1163, Oct. 2015.

[10] R. Mur-Artal and J. D. Tardós, "ORB-SLAM2: An open-source SLAM system for monocular, stereo, and RGB-D cameras," *IEEE Trans. Robot.*, vol. 33, no. 5, pp. 1255–1262, Oct. 2017.

[11] R. A. Newcombe, S. J. Lovegrove, and A. J. Davison, "DTAM: Dense tracking and mapping in real-time," in *Proc. Int. Conf. Comput. Vis.*, Nov. 2011, pp. 2320–2327.

[12] J. Engel, T. Schöps, and D. Cremers, "LSD-SLAM: Large-scale direct monocular SLAM," in *Proc. Eur. Conf. Comput. Vis.*, 2014, pp. 834–849.

[13] T. Whelan, R. F. Salas-Moreno, B. Glocker, A. J. Davison, and S. Leutenegger, "ElasticFusion: Real-time dense SLAM and light source estimation," *Int. J. Robot. Res.*, vol. 35, no. 14, pp. 1697–1716, Sep. 2016.

[14] G. Mohanarajah, V. Usenko, M. Singh, R. D'Andrea, and M. Waibel, "Cloud-based collaborative 3D mapping in real-time with low-cost robots," *IEEE Trans. Autom. Sci. Eng.*, vol. 12, no. 2, pp. 423–431, Apr. 2015.

[15] S. Golodetz, T. Cavallari, N. A. Lord, V. A. Prisacariu, D. W. Murray, and P. H. S. Torr, "Collaborative large-scale dense 3D reconstruction with online inter-agent pose optimisation," *IEEE Trans. Vis. Comput. Graph.*, vol. 24, no. 11, pp. 2895–2905, Nov. 2018.

[16] S. Dong et al., "Multi-robot collaborative dense scene reconstruction," *ACM Trans. Graph.*, vol. 38, no. 4, pp. 84:1–84:16, 2019.

[17] V. Pradeep, C. Rhemann, S. Izadi, C. Zach, M. Bleyer, and S. Bathiche, "MonoFusion: Real-time 3D reconstruction of small scenes with a single web camera," in *Proc. IEEE/ACM Int. Sym. Mixed Augmented Reality*, 2013, pp. 83–88.

[18] M. Pizzoli, C. Forster, and D. Scaramuzza, "REMODE: Probabilistic, monocular dense reconstruction in real time," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2014, pp. 2609–2616.

[19] G. Vogiatzis and C. Hernández, "Video-based, real-time multi-view stereo," *Image Vis. Comput.*, vol. 29, no. 7, pp. 434–441, 2011.

[20] C. Forster, M. Pizzoli, and D. Scaramuzza, "SVO: Fast semi-direct monocular visual odometry," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2014, pp. 15–22.

[21] Z. Yang, F. Gao, and S. Shen, "Real-time monocular dense mapping on aerial robots using visual-inertial fusion," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2017, pp. 4552–4559.

[22] T. Qin, P. Li, and S. Shen, "VINS-Mono: A robust and versatile monocular visual-inertial state estimator," *IEEE Trans. Robot.*, vol. 34, no. 4, pp. 1004–1020, Aug. 2018.

[23] H. Hirschmuller, "Stereo processing by semiglobal matching and mutual information," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 30, no. 2, pp. 328–341, Feb. 2008.

[24] C. Forster, S. Lynen, L. Kneip, and D. Scaramuzza, "Collaborative monocular SLAM with multiple micro aerial vehicles," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Nov. 2013, pp. 3962–3970.

[25] D. Zou and P. Tan, "CoSLAM: Collaborative visual SLAM in dynamic environments," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 2, pp. 354–366, Feb. 2013.

[26] I. Deutsch, M. Liu, and R. Siegwart, "A framework for multi-robot pose graph SLAM," in *Proc. IEEE Int. Conf. Real-time Comput. Robot. (RCAR)*, Jun. 2016, pp. 567–572.

[27] P. Schmuck and M. Chli, "CCM-SLAM: Robust and efficient centralized collaborative monocular simultaneous localization and mapping for robotic teams," *J. Field Robot.*, vol. 36, no. 4, pp. 763–781, Jun. 2019.

[28] S. Leutenegger, S. Lynen, M. Bosse, R. Siegwart, and P. Furgale, "Keyframe-based visual–inertial odometry using nonlinear optimization," *Int. J. Robot. Res.*, vol. 34, no. 3, pp. 314–334, 2015.

[29] R. Mur-Artal and J. D. Tardós, "Visual-inertial monocular SLAM with map reuse," *IEEE Robot. Automat. Lett.*, vol. 2, no. 2, pp. 796–803, Apr. 2017.

[30] M. Karrer, P. Schmuck, and M. Chli, "CVI-SLAM—Collaborative visual-inertial SLAM," *IEEE Robot. Autom. Lett.*, vol. 3, no. 4, pp. 2762–2769, Oct. 2018.

[31] M. Calonder, V. Lepetit, C. Strecha, and P. Fua, "BRIEF: Binary robust independent elementary features," in *Proc. Eur. Conf. Comput. Vis.*, 2010, pp. 778–792.

[32] J. Shi, "Good features to track," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 1994, pp. 593–600.

[33] J. G. Mangelson, D. Dominic, R. M. Eustice, and R. Vasudevan, "Pairwise consistent measurement set maximization for robust multi-robot map merging," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2018, pp. 2916–2923.

[34] Q. Wu and J.-K. Hao, "A review on algorithms for maximum clique problems," *Eur. J. Oper. Res.*, vol. 242, no. 3, pp. 693–709, 2015.

[35] D. Zuckerman, "Linear degree extractors and the inapproximability of max clique and chromatic number," in *Proc. 38th Annu. ACM Symp. Theory Comput. (STOC)*, 2006, pp. 681–690.

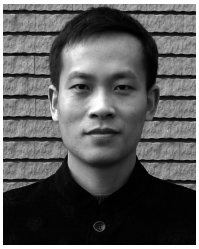
[36] U. Feige, S. Goldwasser, L. Lovasz, S. Safra, and M. Szegedy, "Approximating clique is almost NP-complete," in *Proc. 32nd Annu. Symp. Found. Comput. Sci.*, 1991, pp. 2–12.

[37] B. Pattabiraman, M. M. A. Patwary, A. H. Gebremedhin, W.-K. Liao, and A. Choudhary, "Fast algorithms for the maximum clique problem on massive graphs with applications to overlapping community detection," *Internet Math.*, vol. 11, nos. 4–5, pp. 421–448, Sep. 2015.

[38] J. J. Moré, "The Levenberg–Marquardt algorithm: Implementation and theory," in *Numerical Analysis*, G. A. Watson, Ed. Berlin, Germany: Springer, 1978, pp. 105–116.

[39] M. Klingensmith, I. Dryanovski, S. Srinivasa, and J. Xiao, "CHISEL: Real time large scale 3D reconstruction onboard a mobile device using spatially hashed signed distance fields," in *Proc. Robot., Sci. Syst.*, Rome, Italy, Jul. 2015.
[40] M. Burri et al., "The EuRoC micro aerial vehicle datasets," *Int. J. Robot. Res.*, vol. 35, no. 10, pp. 1157–1163, 2016.
[41] A. Handa, R. Newcombe, A. Angeli, and A. Davison, "Real-time camera tracking: When is high frame-rate best?" in *Proc. Eur. Conf. Comput. Vis.*, 2012, pp. 222–235.
[42] X. Wu, "Top 10 algorithms in data mining," *Knowl. Inf. Syst.*, vol. 14, no. 1, pp. 1–37, 2008.
[43] T. Chai and R. R. Draxler, "Root mean square error (RMSE) or mean absolute error (MAE)?—Arguments against avoiding RMSE in the literature," *Geosci. Model Develop.*, vol. 7, no. 3, pp. 1247–1250, Jun. 2014.

Tianjun Zhang received the B.S. degree from the School of Software Engineering, Tongji University, Shanghai, China, in 2019, where he is currently pursuing the Ph.D. degree. His research interests include collaborative SLAM, computer vision, and sensor calibration.



Lin Zhang (Senior Member, IEEE) received the B.Sc. and M.Sc. degrees from the Department of Computer Science and Engineering, Shanghai Jiao Tong University, Shanghai, China, in 2003 and 2006, respectively, and the Ph.D. degree from the Department of Computing, The Hong Kong Polytechnic University, Hong Kong, in 2011. From March 2011 to August 2011, he was a Research Associate with the Department of Computing, The Hong Kong Polytechnic University. In August 2011, he joined the School of Software Engineering, Tongji University, Shanghai, China, where he is currently a Full Professor. His current research interests include environment perception of intelligent vehicle, pattern recognition, computer vision, and perceptual image/video quality assessment. He was awarded as a Young Scholar of Changjiang Scholars Program, Ministry of Education, China. He serves as an Associate Editor for IEEE ROBOTICS AND AUTOMATION LETTERS and *Journal of Visual Communication and Image Representation*.

Yang Chen received the B.S. degree from the School of Software Engineering, Tongji University, Shanghai, China, in 2020, where she is currently pursuing the Ph.D. degree. Her research interests include SLAM systems, computer vision, and machine learning.

Yicong Zhou (Senior Member, IEEE) received the B.S. degree in electrical engineering from Hunan University, Changsha, China, and the M.S. and Ph.D. degrees in electrical engineering from Tufts University, Medford, MA, USA. He is currently a Full Professor and the Director at the Vision and Image Processing Laboratory, Department of Computer and Information Science, University of Macau, Macau, China. His research interests include chaotic systems, multimedia security, computer vision, and machine learning. He is a Senior Member of the International Society for Optical Engineering (SPIE). He was a recipient of the Third Prize of Macau Natural Science Award in 2014. He is a Co-Chair of the Technical Committee on Cognitive Computing in the IEEE Systems, Man, and Cybernetics Society. He serves as an Associate Editor for *Neurocomputing*, *Journal of Visual Communication and Image Representation*, and *Signal Processing: Image Communication*.