Real-time Global Registration for Globally Consistent RGBD SLAM

Lei Han, Lan Xu, Dmytro Bobkov, Eckehard Steinbach, Qionghai Dai and Lu Fang

Abstract—**Real-time globally consistent camera localization is critical for visual SLAM applications. While Pose Graph Optimization (PGO) is popularly utilized as a backend in SLAM systems for its high efficiency, its deficiency in accuracy can hardly benefit the reconstruction application, as the pose estimation is simplified into frame-to-frame constraints in PGO. An alternative solution for the sake of high accuracy would be global registration, which minimizes the alignment error of all the corresponding observations, yet suffers from high complexity due to the tremendous observations need to be considered. In this paper, we start by analyzing the complexity bottleneck of global registration problem, i.e., each observation has to be linearized based on its local coordinate (camera poses), which however is non-linear and dynamically changing, resulting in extensive computation during optimization. We further prove that such nonlinearity can be decoupled into linear component (feature position) and nonlinear components (camera poses), where the former linear one can be effectively represented by its compact second-order statistics, while the latter nonlinear one merely requires 6 degrees of freedom for each camera pose. Benefit from the decoupled representation, the complexity of global registration can be reduced significantly without sacrifice of accuracy. Experiments show that the proposed algorithm achieves globally consistent pose estimation in real-time via CPU computing, and owns comparable accuracy as state-of-the-art that use GPU computing, enabling the practical usage of globally consistent RGBD SLAM on highly computational constrained devices.**

I. INTRODUCTION

Various approaches have been proposed for RGBD based indoor simultaneous localization and mapping (SLAM) since the emergence of consumer-level depth cameras [1], yet it remains a challenging problem to recover the globally consistent camera poses online [2], restricted by the linearization of nonlinear graph optimization through Taylor expansion that can hardly be accomplished in realtime under highly constrained computational resources. Reviewing recent progress in visual SLAM systems, Loop Closure Detection (LCD) [3], [4] and optimization techniques including Pose Graph Optimization (PGO) [5], [6], [7], Global Registration [8], [9], [10], Bundle Adjustment (BA) [11], [12] have played important roles in the progress of globally consistent camera pose estimation.

LCD aims to detect previously visited places online, thus avoiding traverse search of previous observations. TThe new loop closure constraints, which will be inconsistent due to the accumulated drift introduced by frame-to-frame tracking, provides additional information that allows further optimization techniques to correct this drift. PGO and global registration schemes are commonly used to minimize such inconsistency by averaging errors along the camera poses [5], [6], [7] or feature points [8], [9], [10], respectively. In general, as indicated in [10], [1] and [13], although PGO is much more efficient for real-time applications such as robot/UAV exploration [14], most state-of-the-art SLAM systems aiming for high quality 3D reconstruction and accurate pose estimations [10], [8], [9] still prefer global registration for globally consistent camera pose estimation.

Driven by the high quality 3D reconstruction, researchers have undertaken a considerable number of attempts to develop the global registration techniques [8], [9], achieving attractive precision yet at the expense of off-line computations. Recently, BundleFusion [10] was proposed as an on-line 3D reconstruction system that minimizes the alignment error of all correspondences at keyframe rate, showing great potential for the application of real-time global registration. However, a high-end GPU must be employed to enable the real-time localization in BundleFusion, prohibiting its applications on portable devices for the emerging VR/AR scenarios.

In this paper, we tackle the problem of extensive computation in global registration, and propose a preintegration technique that enables real-time globally consistent SLAM using CPU computing, while achieving competitive accuracy, especially for large-scale datasets. More specifically, by inspecting global registration, we find that:

- **Challenge**: global registration optimizes the alignment error of all the corresponding points, which is determined by both the camera pose and the local 3D position of all the corresponding points. Due to the high nonlinearity of the Euclidean transformation including both rotation and translation, the cost function of global registration can only be optimized iteratively. At each iteration step, all the points must be independently considered based on the updated camera pose and its local position, thereby making it impractical to handle the newly introduced constraints online.
- **Opportunity**: We find that the nonlinear cost function in global registration can be decoupled into two independent components: the linear feature positions and nonlinear camera poses. While the linear component contains a huge number of 3D points, it can be represented by the

Lei Han (lhanaf@connect.ust.hk) and Lan Xu are with the Dept. of ECE, Hong Kong University of Science and Technology. Dmytro Bobkov and Eckehard Steinbach are with the Technical University of Munich (TUM). Qionghai Dai and Lu Fang (fanglu@sz.tsinghua.edu.cn) are with the Tsinghua - Berkeley Shenzhen Institute.

This work is supported in part by the Alexander von Humboldt Foundation under a Research Fellowship for Dr. Lu Fang. This work is supported in part by Natural Science Foundation of China (NSFC) under contract No. 61722209 and 61331015.

The correspondence author is Dr. Lu FANG.

compact second-order statistics of these points, which only needs to be calculated once and can be reused afterwards. For the nonlinear component, each camera pose only requires 6 degrees of freedom on the Lie manifold.

Therefore, we propose a novel Fast Global Optimization (denoted as 'FastGO') scheme to extract the linear component from the non-linear cost function of global registration. While such linear component takes the majority of the computational complexity as in previous methods, we prove that it can be represented by its compact second-order statistics. Thus, the complexity of global registration can be reduced significantly, as shown in Sec. III-B, from $O(M * N_C)$ to $O(M)$, where M represents the number of successfully registered frame pairs and N_C indicates the average number of feature correspondences for each frame pair. We further show that increasing the number of pair-wise correspondences will not decrease the computational efficiency in the optimization procedure. Benefiting from this, we are able to include the dense correspondences generated from Iterative Closest Points (ICP) registrations in the objective function, which makes the estimation more accurate and robust, so as to reflect the overall registration accuracy of frame pair. In summary, the main characteristics of FastGO include:

- **FAST**: Supported by the analysis that the nonlinearity in global registration problem can be decoupled and represented using compact second-order statistics, the complexity of global registration can be reduced from the order of feature correspondence to the successfully registered frame pairs, assuring the possibility of globally consistent visual SLAM applications on highly computational constrained portable devices.
- **SCALABLE**: As the complexity of FastGO grows linearly with the number of keyframes in the database, FastGO scales well for thousands of keyframes, working within 100 ms. Regarding the memory, FastGO simply requires two 3x3 matrices for the second-order statistics of the feature correspondences to represent each pair of frame correspondences, thus avoiding storing the massive number of corresponding features explicitly as done in conventional global registration implementations.
- **ACCURATE**: Dense correspondence from ICP registrations is introduced in the cost function of global registration for better accuracy and robustness. Note that For non-textured scenes, ICP registration is more robust and accurate than visual feature based methods.
- **MODULAR**: As FastGO directly minimizes the alignment error of all correspondences, it can be easily combined with other measurements such as inertial/GPS in a Bayesian framework. For multi-robot applications, FastGO can be adopted for globally consistent map fusion from multiple camera observations. In other words, FastGO can serve as a modular component for globally consistent visual SLAM applications[1].

Based on the proposed FastGO algorithm, a Globally Consistent SLAM system is proposed (denoted as 'GC-SLAM'), running in realtime up to 100 Hz through CPU computing, where only two threads are invoked: front-end for frame tracking and back-end for global optimization. Also to avoid the noisy correspondences of ORB feature matching, an efficient outlier removal strategy based on the isometry in Euclidean transformations is proposed in Sec. IV.

The remainder of this paper is organized as follows. Related work is introduced in Sec. II where the background of global optimization and motivation of FastGO are presented. In Sec. III and Sec. IV, the technical details of FastGO and the framework of GC-SLAM are elaborated. Experimental results based on public datasets are presented in Sec. V. Conclusions and future works are presented in Sec. VI.

II. Related Work

Serving as the foundation of various applications in both robotics and computer vision communities, globally consistent localization has attracted sufficient attention from both academia and industry. Given loop closure cues introduced by loop closure detection techniques [4], [3], optimization works as the final step to ensure global consistency by correcting the inevitable drift caused by frame-by-frame or frame-by-model tracking, showing a considerable influence on accuracy. Recent years have witnessed extensive progress in both loop closure detection techniques and optimization techniques. As we pay more attention on the latter one, the literature review is conducted on pose graph optimization (PGO), bundle adjustment (BA) and global registration given their formulations, as elaborated in the following paragraphs respectively.

In PGO, poses are adjusted to minimize the inconsistency of relative transformation of frames: $\min_{T_i \in SE(3)} \sum_{(i,j)\in\Omega} ||T_j T_i^{-1} - T_{ij}||_F^2$, where T_{ij} is the transformation matrix from frame f_i to f_j and Ω indicates the collection of frame pairs that have overlapped observations. Various approaches have been tried [6], [7] to solve this constrained optimization problem using convex relaxations and iterative Riemannian trust region methods, yielding an a-posteriori certifiably globally optimal solution. Solving the transformation matrix between frames directly assures the efficiency of PGO, yet it may lead to inferior estimations and be easily influenced by outliers, as it adopts a fixed or simplified camera pose uncertainty. Due to the high nonlinearity of SE3 space (including rotation and translation), such fixed uncertainty simplification can hardly hold at different camera pose configurations.

At the other end of the spectrum, BA [11], [15], [12], [16] aims to optimize both camera positions and the 3D poses of landmarks by minimizing the re-projection error of feature points directly. Typically the number of landmarks is much larger than the number of frames. Restricted by the huge amount of variables to be optimized, BA algorithms suffer from heavy computational burden, e.g., it may take several seconds for a map containing hundreds of keyframes. Such significant delay is unbearable for applications that require real-time performance.

[1] As this paper investigates the FastGO algorithm, the combination of visual observations and inertial measurements is out of scope and will be studied as future work.

Global registration lies between BA and PGO, where the alignment errors of all correspondences are minimized yet the local poses of features are remained as fixed. Global point cloud registration has been widely studied for decades and various methods have been proposed to get the optimal solution, either using semidefinite programming [17] or via low-rank and sparse decomposition [18]. Our work lies in the track of global registration, and we solve it on manifold though Gauss-Newton optimization for its high accuracy and relatively low complexity, which is also adopted by many offline approaches like [8], [9], [10]. Both [8] and [9] are designed for 3D reconstruction based on the depth observations merely, where the line-process technique is adopted to ensure robust pose estimation for frame-pairs and feature-pairs respectively. In particular, [8] aims to minimize the registration error of all the collected pair-wise correspondences, which is solved at the complexity of the number of frames instead of the number of correspondences. One may notice that we share the same objective as [8], nevertheless, [8] eventually solves an approximation of the original cost function. More specifically, for each correspondence pair (p_1, p_2) and its relative transformation T, direct observation of p_2 is approximated by Tp_1. While in our work, we demonstrate that the original cost function can be solved without approximation based on the proposed pre-integration technique. Our proposed GC-SLAM is further testified on the datasets provided by [8] in Section V-E. Note that while [8] runs at off-line and takes hours for correspondence collection, GC-SLAM achieves comparable performance yet running at real-time as shown in Table IV.

[19] minimizes the 3D alignment error of all correspondences without considering each correspondence independently. However, the rotation matrix is relaxed to an affine matrix firstly, then the best affine matrix is computed by minimizing the target function, followed by projecting the affine matrix back to the rotation matrix in terms of the Frobenius norm. Such procedure cannot guarantee the optimal rotation matrix, thus it is not trustable to serve as an initialization approach. In this paper, we step forward by optimizing the relative transformation directly in SE3 space using the non-linear Gauss-Newton algorithm, without introducing any intermediate variable.

The proposed system shares similar framework as BundleFusion [10], where loop closures are handled at the keyframe-rate, aqhdai.jpgchieving high-quality online indoor 3D reconstructions. It is worthy to note that BundleFusion requires for high-end GPU as the computing resource for real-time pose estimation, which is demanding for on-board implementation in portable devices. In this paper, aiming for the real-time performance using highly constrained computation, ORB features are employed instead of SIFT features, and a robust loop closure detector MILD [3] is adopted to approximate the exhaustive search strategy. More importantly, by analyzing the complexity bottleneck of global registration problem, we propose to decouple the nonlinearity into linear component (feature position) and nonlinear components (camera poses), where the former linear one can be effectively represented by its compact second-order statistics, while the latter nonlinear one merely requires 6 degrees of freedom for each camera pose. Benefit from the decoupled representation, the complexity of global registration can be reduced significantly without sacrifice of accuracy, realizing a globally consistent localization that runs up to $100Hz$ using CPU computing.

III. FastGO: Fast Global Optimization for Globally Consistent RGBD Localization

In this section, we elaborate the proposed fast global optimization scheme for globally consistent RGBD localization (denoted as FastGO), which minimizes the alignment error of feature points in Euclidean space, given the depth information obtained by stereo or RGBD cameras.

A. Problem Analysis

For ease of presentation, we denote the i-th frame as f_i, and the corresponding RGB image and depth image are denoted as I_i and D_i, respectively. The camera pose of f_i is denoted as T_i, i.e., the relative transformation from the local coordinates to world coordinates.

For each frame pair (f_i, f_j), the corresponding points $C_{i,j} = \{C_{i,j}^k = (p_i^k, p_j^k) | k = 0, 1, \cdots, ||C_{i,j}|| - 1\}$ are collected either from sparse feature association or dense ICP registration if they can be aligned by rigid transformation, where p_i^k represents the k-th point observed in the local coordinates of the i-th frame. Globally consistent pose estimations $T_i, i = 1, 2, \cdots, N-1$ can be found by minimizing the alignment error in Euclidean space:

$$E(T_i, i=1,\cdots,N{-}1) = \sum_{i=1}^{N-1}\sum_{j=0}^{i-1}\sum_{k=0}^{||C_{i,j}||-1} ||T_i P_i^k {-} T_j P_j^k||^2, \tag{1}$$

where $P_i^k = [p_i^k|1]$ represents the homogeneous coordinates of the local 3D point p_i^k. The pose of the first frame T_0 is initialized as the world coordinates, and N represents the total number of the collected frames. Eqn. (1) can be solved using nonlinear Gauss-Newton optimization on the Lie manifold [20], as indicated in [10].

Examining Eqn. (1), rigid transformation T_i in Euclidean space can be represented using Lie algebra ξ_i on the SE3 manifold. $T(\xi_i)$ maps ξ_i in Lie algebra to T_i in Euclidean space. SE3 parameterizations provide the most compact representations for 3D transformation: 6 variables for 6 DOF. Let $\boldsymbol{\xi}$ denote the vector of camera poses to be optimized: $\xi_i, i = 1, \cdots, N-1$. For each correspondence $C_{i,j}^k$, the alignment residual is defined as

$$r_{i,j}^k(\boldsymbol{\xi}) = T(\xi_i)P_i^k - T(\xi_j)P_j^k. \tag{2}$$

Then, the original objective in Eqn. (1) is represented as:

$$E(\boldsymbol{\xi}) = ||\boldsymbol{r}(\boldsymbol{\xi})||^2, \tag{3}$$

where $\boldsymbol{r}(\boldsymbol{\xi})$ is a vector containing all the alignment errors: $[\cdots, r_{i,j}^k(\boldsymbol{\xi}), \cdots], i \in [0, N{-}1], j \in [0, i{-}1], k \in [0, ||C_{i,j}||-1]$. Suppose we have N_{corr} correspondences in total, then $\boldsymbol{r}(\boldsymbol{\xi})$ should be a vector with size $3N_{corr} \times 1$.

By linearization, we have

$$\boldsymbol{r}(\boldsymbol{\xi}) = \boldsymbol{r}(\boldsymbol{\xi_0}) + J(\boldsymbol{\xi_0})\boldsymbol{\delta}, \tag{4}$$

where $J(\boldsymbol{\xi_0})$ is the Jacobian matrix of $\boldsymbol{r}(\boldsymbol{\xi_0})$, and $\boldsymbol{\xi} = \boldsymbol{\xi_0} + \boldsymbol{\delta}$. Following the standard non-linear Gauss-Newton optimization procedure, the Hessian matrix H is approximated using $2J(\boldsymbol{\xi_0})^T J(\boldsymbol{\xi_0})$ and the camera poses can be updated iteratively based on:

$$J(\boldsymbol{\xi_0})^T J(\boldsymbol{\xi_0})\boldsymbol{\delta} = -J(\boldsymbol{\xi_0})^T \boldsymbol{r}(\boldsymbol{\xi_0}). \tag{5}$$

During each iteration, the Jacobian matrix must be updated based on the latest pose estimation $\boldsymbol{\xi_0}$ for accuracy.

Suppose we have N_{corr} pair-wise correspondences in total and $N-1$ frames to be estimated. The size of the Jacobian matrix would be $3N_{corr} \times 6(N-1)$. N_{corr} can be approximated by the average number of correspondences in each frame pair N_C and the number of successfully registered frame pairs M, i.e., $N_{corr} = M * N_C$. In practice, we do not need to compute $J(\boldsymbol{\xi_0})$ explicitly as only $J(\boldsymbol{\xi_0})^T J(\boldsymbol{\xi_0})$ and $J(\boldsymbol{\xi_0})^T r(\boldsymbol{\xi_0})$ are required in the iteration step of Eqn. (5). Each pairwise correspondence will contribute one additional term in the $J(\boldsymbol{\xi_0})^T J(\boldsymbol{\xi_0})$ and $J(\boldsymbol{\xi_0})^T r(\boldsymbol{\xi_0})$. Thus, $J(\boldsymbol{\xi_0})^T J(\boldsymbol{\xi_0})$ and $J(\boldsymbol{\xi_0})^T r(\boldsymbol{\xi_0})$ can be computed by traversing all the correspondences in the cost function term, with the complexity of $O(N_{corr})$. Due to the huge amount of correspondences involved, high-end GPU devices must be employed for parallel computing, e.g., each kernel for one correspondence. Although dense correspondences collected from ICP registration can improve the results, the cost function including dense correspondences can only be optimized offline, losing the opportunity for higher quality online 3D reconstruction and restricting their use on portable devices such as Google project TANGO$^{\text{TM}}$ [21] or Microsoft HoloLens$^{\text{TM}}$ [22], which can hardly employ the same level GPU equipment.

B. Fast Global Optimization

Based on the analysis in Section III-A, the main complexity of the global registration problem in Eqn. (1) lies in the formulation of the normal equation in Eqn. (5). Since $J(\boldsymbol{\xi_0})^T J(\boldsymbol{\xi_0})$ is a sparse matrix containing only $O(M)$ non-zero entries, the normal equation can be efficiently solved with the complexity of $O(M)$. However, to calculate the $J(\boldsymbol{\xi_0})^T J(\boldsymbol{\xi_0})$, all the corresponding features must be considered based on the latest camera poses, with the complexity of $O(N_{corr})$. In this section, we will show that the complexity of global registration can be effectively reduced to $O(M)$ based on a dedicated analysis of the Jacobian matrix.

Following the introduction in [20], which provides a detailed interpretation on the manifold of SE3 space and Lie algebra, the Jacobian of transformation $T(\xi_i)p_{ik}$ on the Lie manifold can be written as

$$J_i^k(\xi_i) = \begin{bmatrix} I_{3\times 3} & -[T(\xi_i)P_i^k]_\times \end{bmatrix}, \tag{6}$$

where $[p]_\times$ indicates the corresponding skew-symmetric matrix of vector p.

For the m-th pairwise correspondence $C_{i,j}^k$, the corresponding submatrix of the original Jacobian matrix $J(\boldsymbol{\xi_0})$ is

$$J_m(\boldsymbol{\xi_0}) = [\mathbf{0} \cdots J_i^k(\xi_i) \cdots \mathbf{0} \cdots - J_j^k(\xi_j) \cdots \mathbf{0}]. \tag{7}$$

In the following notations, we will omit $\boldsymbol{\xi_0}$ in $J(\boldsymbol{\xi_0})$ for simplicity. The corresponding residual $r_m(\boldsymbol{\xi})$ can be calculated based on Eqn. (2). J_m and r_m will contribute an additive term to $J^T J$ and $J^T r$, i.e.,

$$J_m^T J_m = \begin{bmatrix} \mathbf{0} & \cdots & \mathbf{0} & \cdots & \mathbf{0} & \cdots & \mathbf{0} \\ \vdots & & \vdots & & \vdots & & \vdots \\ \mathbf{0} & \cdots & {J_i^k}^T J_i^k & \cdots & -{J_i^k}^T J_j^k & \cdots & \mathbf{0} \\ \vdots & & \vdots & & \vdots & & \vdots \\ \mathbf{0} & \cdots & -{J_j^k}^T J_i^k & \cdots & {J_j^k}^T J_j^k & \cdots & \mathbf{0} \\ \vdots & & \vdots & & \vdots & & \vdots \\ \mathbf{0} & \cdots & \mathbf{0} & \cdots & \mathbf{0} & \cdots & \mathbf{0} \end{bmatrix}, \tag{8}$$

$$J_m^T r = \begin{bmatrix} \mathbf{0} \cdots {J_i^k}^T r_m \cdots - {J_j^k}^T r_m \cdots \mathbf{0} \end{bmatrix}^T. \tag{9}$$

Thus, $J^T J$ and $J^T r$ can be computed by accumulating $J_m^T J_m$ and $J_m^T r$, respectively,

$$J^T J = \sum_{m=0}^{N_{corr}-1} J_m^T J_m, \tag{10}$$

$$J^T r = \sum_{m=0}^{N_{corr}-1} J_m r, \tag{11}$$

for $m = 0, 1, \cdots, N_{corr} - 1$.

The Jacobian matrix is then determined by the camera pose $\boldsymbol{\xi}$ and local 3D position of each point p_i^k. For each iteration, we have to re-linearize $E(\boldsymbol{\xi})$ based on the updated camera pose. Hence, the computational cost of each iteration is proportional to the number of corresponding points N_{corr}, making it impractical to get the globally consistent camera poses at frame rate.

To simplify notation, we use $m \in C_{i,j}$ to indicate that the m_{th} correspondence belongs to $C_{i,j}$. Let $J_{C_{i,j}}$ denote the corresponding Jacobian matrix of all the correspondences in $C_{i,j}$. Thus, we have

$$J_{C_{i,j}}^T J_{C_{i,j}} = \sum_{m \in C_{i,j}} J_m^T J_m \tag{12}$$

$$J_{C_{i,j}}^T r = \sum_{m \in C_{i,j}} J_m^T r \tag{13}$$

$$J^T J = \sum_{\forall ||C_{i,j}||>0} J_{C_{i,j}}^T J_{C_{i,j}} \tag{14}$$

$$J^T r = \sum_{\forall ||C_{i,j}||>0} J_m^T r, \tag{15}$$

J_m, $J_{C_{i,j}}$ and $J_{C_{i,j}}^T J_{C_{i,j}}$ are sparse matrices as demonstrated in Fig. 1. As shown in Eqn 6, the Jacobian matrix of each correspondence $C_{i,j}^k$ is determined by both camera poses ξ_i, ξ_j and local positions of features P_i^k, P_j^k. For all the correspondences in the frame pair $C_{i,j}$, their corresponding Jacobian matrices share the same geometric terms but different structure terms. It is well known that $J_{Ci,j}^T J_{C_{i,j}}$ is nonlinear to the structure terms $P_i^k, P_j^k, k \in [0, ||C_{i,j}||-1]$. For the updated camera pose $\boldsymbol{\xi}$, all the correspondences have to be revisited to calculate $J^T J$. In the following parts of this section, we will demonstrate that the sparse matrix $J_{Ci,j}^T J_{C_{i,j}}$ is **element-wise linear** to the second-order statistics of the structure terms

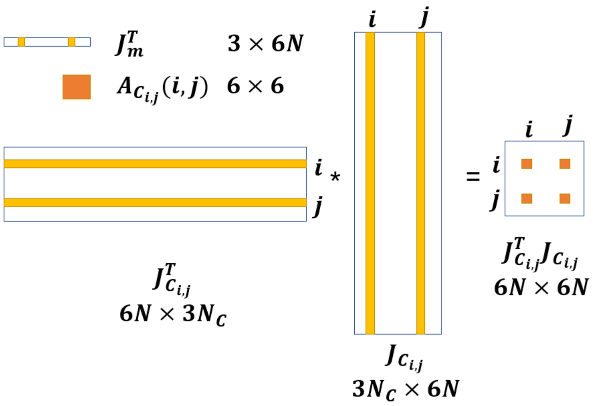


Fig. 1: Sparse matrix representation in the computation of $J_{C_{i,j}}^T J_{C_{i,j}}$. Blank areas represent zeros. $J_{C_{i,j}}$ is composed of $J_m, m = 0, 1, \cdots, N_C - 1$, where N_C represents the number of feature matches in $C_{i,j}$. N indicates the number of camera poses. The four non-zero 6×6 matrices in $J_{C_{i,j}}^T J_{C_{i,j}}$ are denoted as $A_{C_{i,j}}(i,i)$, $A_{C_{i,j}}(i,j)$, $A_{C_{i,j}}(j,i)$, $A_{C_{i,j}}(j,j)$, respectively.

in $C_{i,j}$. In this way, $J_{Ci,j}^T J_{C_{i,j}}$ can be efficiently computed with a constant complexity, independent of the number of correspondences in $C_{i,j}$. As introduced in Section III-A, $C_{i,j}$ represents the collections of correspondences between frame f_i and f_j.

Recall that $T(\xi_i)$ is a Euclidean transformation matrix $T_i = [R_i|t_i]$ with the dimension of 3×4, then $T(\xi_i)P_i^k$ can be reorganized as

$$T_i P_i^k = R_i p_i^k + t_i = \begin{bmatrix} r_{i0}^T p_i^k + t_{i0} \\ r_{i1}^T p_i^k + t_{i1} \\ r_{i2}^T p_i^k + t_{i2} \end{bmatrix}, \tag{16}$$

where r_{il}^T represents the l-th row in rotation matrix R_i and t_{il} represents the l_{th} element in translation vector t_i. $J_{ik}^T J_{jk}$ is a 6×6 square matrix:

$$J_{ik}^T J_{jk} = \begin{bmatrix} I_{3\times3} & -\left[T_j P_j^k\right]_\times \\ -[T_i P_i^k]_\times^T & [T_i P_i^k]_\times^T [T_j P_j^k]_\times \end{bmatrix}. \tag{17}$$

Let $A_{C_{i,j}}(i,i)$, $A_{C_{i,j}}(i,j)$, $A_{C_{i,j}}(j,i)$ and $A_{C_{i,j}}(j,j)$ denote the four non-zero 6×6 submatrices in $J_{C_{i,j}}^T J_{C_{i,j}}$, $\sum_{k=0}^{||C_{i,j}||-1} J_i^{k\,T} J_i^k$, $\sum_{k=0}^{||C_{i,j}||-1} J_i^{k\,T} J_j^k$, $\sum_{k=0}^{||C_{i,j}||-1} J_j^{k\,T} J_i^k$ and $\sum_{k=0}^{||C_{i,j}||-1} J_j^{k\,T} J_j^k$, respectively, as demonstrated in Eqn. 8, then $A_{C_{i,j}}(i,i)$, $A_{C_{i,j}}(i,j)$, $A_{C_{i,j}}(j,i)$ and $A_{C_{i,j}}(j,j)$ can be computed with the complexity of $O(1)$ instead of $O(||C_{i,j}||)$, which will be elaborated later. For simplicity, we show the computation of $A_{C_{i,j}}(i,j)$ in this paper, while the other three terms can be computed accordingly.

Expanding $A_{C_{i,j}}(i,j)$, we have

$$\begin{aligned} A_{C_{i,j}}(i,j) &= \sum_{k=0}^{||C_{i,j}||-1} J_{ik}^T J_{jk} \\ &= \begin{bmatrix} ||C_{i,j}|| I_{3\times3} & -\left[T_j \sum_{k=0}^{||C_{i,j}||-1} P_j^k\right]_\times \\ -\left[T_i \sum_{k=0}^{||C_{i,j}||-1} P_i^k\right]_\times^T & \sum_{k=0}^{||C_{i,j}||-1} [T_i P_i^k]_\times^T [T_j P_j^k]_\times \end{bmatrix}, \end{aligned} \tag{18}$$

where

$$\begin{aligned} &\sum_{k=0}^{||C_{i,j}||-1} [T_i P_i^k]_\times^T [T_j P_j^k]_\times \\ =& \sum_{k=0}^{||C_{i,j}||-1} [R_i p_i^k + t_i]_\times [R_j p_j^k + t_j]_\times \\ =& \sum_{k=0}^{||C_{i,j}||-1} [R_i p_i^k]_\times [R_j p_j^k]_\times + [t_i]_\times [R_j \sum_{k=0}^{||C_{i,j}||-1} p_j^k]_\times \\ +& [R_i \sum_{k=0}^{||C_{i,j}||-1} p_i^k]_\times [t_j]_\times + [t_i]_\times [t_j]_\times. \end{aligned} \tag{19}$$

Denote $\sum_{k=0}^{||C_{i,j}||-1} p_i^k p_j^{k\,T}$ as W. Then the nonlinear term $\sum_{k=0}^{||C_{i,j}||-1} [R_i p_i^k]_\times [R_j p_j^k]_\times$ in Eqn. (19) can be simplified to Eqn. (20), where all the elements in this nonlinear term are linear to W. Finally, all the non-zero elements in $A_{C_{i,j}}(i,j)$, have been proven to be linear to the second-order statistics of the structure terms in $C_{i,j}$, namely $\sum_{k=0}^{||C_{i,j}||-1} p_i^k$, $\sum_{k=0}^{||C_{i,j}||-1} p_j^k$ and $\sum_{k=0}^{||C_{i,j}||-1} p_i^k p_j^{k\,T}$. Similarly, $J^T r$ can be computed based on the previous summations.

As a result, the sparse matrices $J^T J$ and $J^T r$ that are required in the iteration step of non-linear Gauss-Newton optimization in Eqn. (5) can be computed with the complexity of $O(M)$ instead of $O(N_{corr})$ as in previous work [10]. In general, N_{corr} can be approximately 300 times larger than M for sparse feature correspondences or $10,000$ times larger for dense correspondences.

IV. GC-SLAM: Globally Consistent SLAM System Based on FastGO

In this section, a Globally Consistent SLAM (GC-SLAM) system is presented based on the proposed FastGO optimization technique. In GC-SLAM, two threads are included as shown in Fig. 2: a front-end thread for camera tracking working at the frame-rate and a back-end thread for global pose optimization working at the keyframe-rate. Loop closures are detected based on MILD [3]. Compared with DBOW [4]

$$
\begin{aligned}
&\sum_{k=0}^{||C_{i,j}||-1} [R_i p_{ik}]_\times [R_j p_j^k]_\times \\
=&\sum_{k=0}^{||C_{i,j}||-1} \begin{bmatrix} 0 & -r_{i2}^T p_i^k & r_{i1}^T p_i^k \\ r_{i2}^T p_i^k & 0 & -r_{i0}^T p_i^k \\ -r_{i1}^T p_i^k & r_{i0}^T p_i^k & 0 \end{bmatrix} \begin{bmatrix} 0 & -r_{j2}^T p_j^k & r_{j1}^T p_j^k \\ r_{j2}^T p_j^k & 0 & -r_{j0}^T p_j^k \\ -r_{j1}^T p_j^k & r_{j0}^T p_j^k & 0 \end{bmatrix} \\
=&\sum_{k=0}^{||C_{i,j}||-1} \begin{bmatrix} -{p_i^k}^T (r_{i2} r_{j2}^T + r_{i1} r_{j1}^T) p_j^k & {p_i^k}^T r_{i1} r_{j0}^T p_j^k & {p_i^k}^T r_{i2} r_{j0}^T p_j^k \\ {p_i^k}^T r_{i0} r_{j1}^T p_j^k & -{p_i^k}^T (r_{i2} r_{j2}^T + r_{i0} r_{j0}^T) p_j^k & {p_i^k}^T r_{i2} r_{j1}^T p_j^k \\ {p_i^k}^T r_{i0} r_{j2}^T p_j^k & {p_i^k}^T r_{i1} r_{j2}^T p_j^k & -{p_i^k}^T (r_{i0} r_{j0}^T + r_{i1} r_{j1}^T) p_j^k \end{bmatrix} \\
=&\begin{bmatrix} -r_{i2}^T W r_{j2} - r_{i1}^T W r_{j1} & r_{i1}^T W r_{j0} & r_{i2}^T W r_{j0} \\ r_{i0}^T W r_{j1} & -r_{i2}^T W r_{j2} - r_{i0}^T W r_{j0} & r_{i2}^T W r_{j1} \\ r_{i0}^T W r_{j2} & r_{i1}^T W r_{j2} & -r_{i0}^T W r_{j0} - r_{i1}^T W r_{j1} \end{bmatrix}
\end{aligned}
\tag{20}
$$

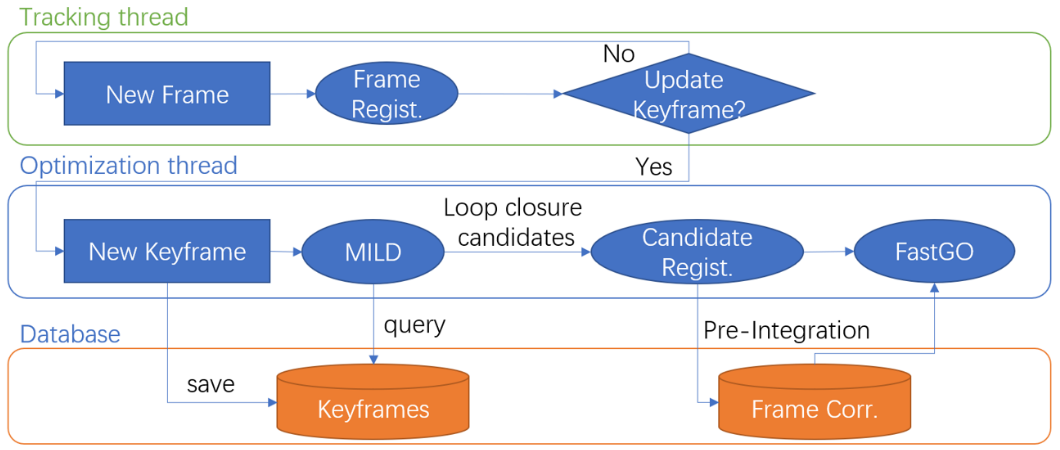


Fig. 2: Framework of GC-SLAM.

adopted by ORBSLAM, MILD does not require a training dictionary and hence is more reliable for real-world applications.

In GC-SLAM, global consistency is achieved by minimizing the alignment error of all correspondences that is collected by frame-pair registration. For frame f_i and f_j, if f_i and f_j can be registered successfully, their correspondences (corresponding local 3D points) will be stored in $C(i,j)$. The traverse search of current frame to all previous frames is approximated by matching loop closure candidates for scalability. Note that the cost function in Eqn. (21) can be solved efficiently without considering each point individually, as explained in Sec. III-B:

$$
\sum_{(f_i,f_j)\in C(i,j)} \sum_{k=0}^{||C(i,j)||} ||T_i P_i^k - T_j P_j^k||, \tag{21}
$$

where P_i^k represents the local 3D position of feature f_i^k in frame f_i. In the following, we will elaborate the main procedures in GC-SLAM.

A. Frame-pair Registration

Two configurations are tested for RGB-D registration, i.e., sparse feature based and dense direct method based, respectively. Sparse feature based registration is more efficient, enabling the GC-SLAM to run at approximately $50-100$ Hz for mobile applications, while the dense registration is more accurate and robust, considering both geometric and photometric consistency, with a registration frequency of 25 Hz.

1) Sparse registration:: The procedure of sparse feature based image registration is presented in Fig. 3, where ORB features are extracted from the RGB image, and the depth is acquired directly from the depth images. We use a previous work SparseMatch [23] for fast Approximate Nearest Neighbor (ANN) search based on the Hamming distances.

Although binary features are efficient to be extracted and matched, they are less distinctive than real-valued features and many outliers exist after an ANN search based on the Hamming distance. Various approaches have addressed the problem of outlier removal for better registration. [24] employs point pair features to describe object model and a voting scheme is adopted to align the local coordinate and object model for robust pose estimations. [25] employs affine invariants of 4-point sets for correspondence search and further accelerates the procedure using hashing technique. [9] proposes to remove outliers by employing the isometry of rigid transformations by randomly select three feature pairs and test if the selected feature pairs satisfy the isometry constraints. However, the test is success only when all the selected feature pairs are inliers, which will be quite inefficient when the outlier ratio

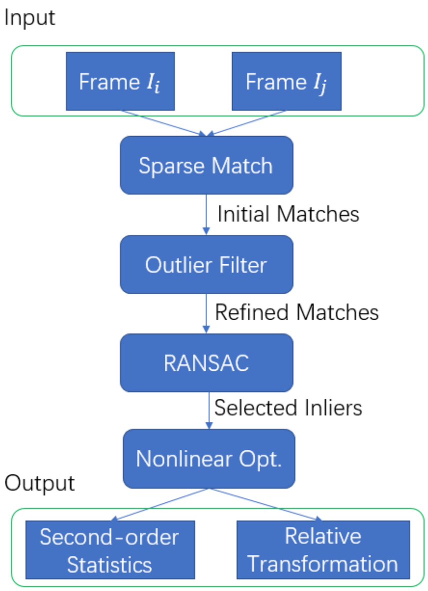


Fig. 3: Procedure of sparse RGB-D frame registration.

is large and cannot guarantee that all the inliers are selected. Here, a outlier removal strategy is presented with a detailed theoretical analysis, showing that outliers can be filtered out both effectively and efficiently, even when the outlier ratio is larger than 60%.

Specifically, we propose to filter outliers based on the isometry of Euclidean transformation: the relative distance between two points will not be changed after Euclidean transformation. However, for outliers, the relative distance is unpredictable, the possibility that their relative distance is unchanged is close to 0. Based on this observation, for each feature correspondence f_i^k, f_j^k in $C_{i,j}$, N feature correspondences are randomly selected: $f_i^{r(t)}, f_j^{r(t)}, t = 0, 1, \cdots, N-1$. The probability of at least one inlier correspondence being selected is denoted as p_r. Given the probability of the inlier ratio in the initial matching correspondences p_o, we would expect that $p_r = 1 - {p_o}^N$ based on the independent sampling assumption. As long as N is large enough, p_r will be close to 1.

$$\alpha = \min(|1 - \frac{||P_i^k - P_i^{r(t)}||}{||P_j^k - P_j^{r(t)}||}|), t \in [0, N-1]. \tag{22}$$

The statistic α computed in Eqn. 22 can be used to verify if f_i^k, f_j^k is an inlier match. If f_i^k, f_j^k is a true inlier match and at least one inlier match is selected in $f_i^{r(t)}, f_j^{r(t)}$, for $t = 0, 1, \cdots, N-1$, α should be close to 0 based on the isometry of Euclidean transformation. Otherwise, α is randomly distributed in $[0, 1]$. Even when the outlier ratio p_o is larger than 60%, the proposed outlier strategy can still filter out outliers accurately since no assumptions on the inlier distribution is made.

Given the refined matches, RANSAC is performed based on the rigid transformation in Euclidean space [26]. Note that the required iteration time is determined by the inlier ratio in the feature matches. The proposed outlier filtering strategy significantly increases the inlier ratio and reduces the required iterations in the RANSAC step. Inliers that fit the final model are selected and used for non-linear optimization in the Lie group to estimate the accurate relative transformation. For the keyframe pair registration, the second order statistics of correspondences are calculated for the global optimization step.

2) Dense registration: The RGB-D registration implemented in DVO [14] is used for dense registration, where both the photometric and geometric error over all of the pixels are minimized. Correspondences are selected if their re-projection error is less than a threshold. For the dense registration, we use a fixed resolution of 320×240 and nearly $50,000$ correspondences are selected based on the criterion of alignment error. Benefit from the proposed FastGO scheme, where the computational complexity of global registration is independent of the number of corresponding points in each frame pair, we are able to achieve online operation of global registration with the dense correspondences.

B. Tracking Thread

In the tracking thread, the current frame is registered to the latest keyframe either based on the sparse ORB features or dense RGB-D registration. If the relative translation between the current frame and keyframe is larger than a threshold T_{update}, then the current frame is regarded as a new keyframe and is stored in the database.

C. Optimization Thread

When a new keyframe is inserted, the loop closure detector (MILD [23]) queries the database to find previous keyframes indicating the same place. The top 5 candidates provided by MILD are registered to the new keyframe. Only sparse match is enabled in the optimization thread for efficiency. The correspondences of each frame-pair registration are stored in the database, and the second-order statistics that will be used for FastGO are computed and used afterwards. All the keyframe poses will be updated based on the newly introduced constraints using the FastGO technique. The poses of the local frames only depend on their corresponding keyframe. Note that the optimization thread only operates when a new keyframe is inserted.

Certainly, sometimes many false loop closure candidates may be provided by the loop closure detector, which are usually similar in appearance but belong to different places. To prevent these false loop closures, we employ the observation that true loop closures reduce the covariance of the pose estimations of frames merely, and will not increase the global registration error after optimization. On the contrary, the false loop closures can bias the objective function and increase the global registration error significantly even after global optimization. In other words, we only accept loop closures if the newly introduced loop closure converges with previous observations, indicating that after optimization, the global registration error is not increased. Experiments on challenging datasets (AUG ICL-NUIM) [8] demonstrate that the proposed GC-SLAM works stably even in the case that many false loop closures exist (e.g., the same computer screens in the office dataset), as presented in Table IV.

V. Experiments

The proposed FastGO is tested on public datasets in terms of accuracy, efficiency, scalability and robustness, as shown in the following subsections respectively. Accuracy is measured by the Absolute Trajectory Error (ATE), while efficiency is evaluated considering both the runtime and the platform. It is worth noting that the performance of GC-SLAM is affected by the number of features extracted. A greater number of features leads to higher accuracy and higher complexity. Thus, we implement GC-SLAM for both sparse feature based and dense feature based tracking strategy, denoted as Sparse GC-SLAM and Dense GC-SLAM, respectively. Experiments are conducted on both synthetic ICL-NUIM [27], [8] dataset (with noise) and real world TUM RGBD dataset [28], on an Intel-core i7 @ 3.6 GHz processor.[2] For the sparse binary feature based methods: Sparse GC-SLAM and ORBSLAM2, 1000 ORB features are extracted for accuracy and efficiency comparisons.

State-of-the-art algorithms are evaluated, including BundleFusion [10], which optimizes the same cost function as in Eqn. (1) using a high-end GPU, and CPA-SLAM [29] that employs the "plane" prior and also relies on parallel computing based on GPU devices. DVO SLAM [14] and ORBSLAM [13] are CPU based methods, where the former one uses dense registration for frame tracking and PGO to reduce drift, while the latter one is based on the sparse ORB features and uses a back-end thread for bundle adjustment to realize global consistency.

A. Accuracy Evaluations

As presented in Tab. I, compared with state-of-the-art approaches, FastGO achieves high accuracy as expected for all the provided datasets, while running at 50 Hz with a standard CPU. ORBSLAM2 [13] achieves high accuracy on the TUM RGBD dataset, where only a small-scale scene is captured, e.g., a desk or a poster. Although BundleFusion [10] achieves the highest accuracy for a larger environment in the ICL-NUMI dataset, e.g., a living room, it requires a high-end GPU which is not feasible for portable devices.

Note that we use the similar front-end with DVO in the Dense GC-SLAM. However, DVO SLAM adopts PGO [30] for global consistency, which fixes the covariance of the frame pair registrations as an information matrix; while in Dense GC-SLAM, the relative constraints between frame pairs are modulated by the second-order statistics of the local points and the latest pose of each frame using the FastGO technique. As a result, the accuracy of dense GC-SLAM significantly exceeds DVO in all the datasets.

As an RGBD SLAM system, FastGO directly uses both appearance measurements and depth measurements in the cost function. However, in the TUM RGBD dataset, although the depth camera and RGB camera are registered in the spatial domain, they will diverge in the temporal domain, which results in a systematic bias for depth observations. While ORBSLAM2 uses only depth observations in the initialization step and optimizes the position of the feature points using multi-view observations, which will be less affected by the inaccurate depth observations. However, note that the performance of ORBSLAM2 is only accurate for small environments where loop closure happens repeatedly. For larger datasets as presented in the ICL-NUIM dataset, FastGO is more accurate.

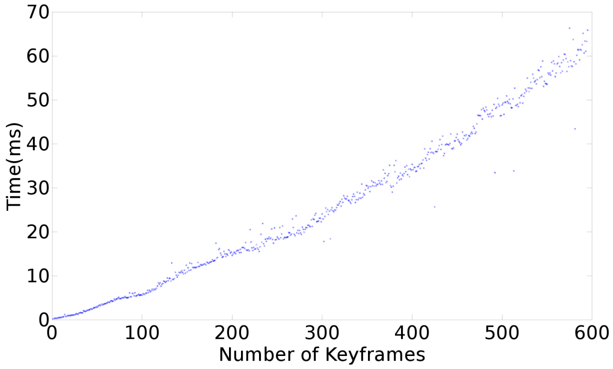


Fig. 4: Complexity of FastGO grows almost linearly with the number of keyframes in the map.

B. Efficiency Evaluations

Examining Tab. I, both GC-SLAM and ORBSLAM2 show competitive performance in terms of efficiency. However, it is worthwhile noticing that the BA optimization in ORBSLAM2 updates at a very low frequency (less than 1 Hz), while in GC-SLAM, once a new keyframe is inserted, BA can be finished within 20 ms using the proposed FastGO technique. Note that the inconsistencies may introduce an unacceptable experience for applications with user interactions and should be minimized as soon as possible. On the other end of the spectrum, the BA in ORBSLAM2 cannot handle large-scale scenes because its complexity is linear in the number of features in the global map; while FastGO is more efficient and scalable, with a complexity linear in the number of keyframes. Note that each frame may contain hundreds of feature points on average.

In addition, ORBSLAM2 maintains the accurate position and covariance of each feature independently in its global map; while in GC-SLAM, we focus on the relative constraints between frame pairs, which is determined by the re-projection error of corresponding points. Although previous PGO approaches aim at omitting features for large scale problems as well, they fail to maintain the accurate relative constraints between frames when the relative pose is updated due to the high nonlinearity in rotation space.

For better comparison, we further unfold the computations of GC-SLAM and ORBSLAM2 in Tab. II. The experiments are implemented on the fr3office dataset, which contains 2488 RGB and depth images with a resolution of 640×480. For each image, 1000 ORB features are extracted. It is shown

[2] The source code will be made public and maintained at github, please refer to www.luvision.net/FastGO.

TABLE I: Quantitative evaluations on public datasets in terms of absolute trajectory error (cm)

	kt0	kt1	kt2	kt3	fr1/desk	fr2/xyz	fr3/office	fr3/nst	Efficiency
BundleFusion on-line	0.8	0.5	1.1	1.2	1.7	1.4	2.8	1.4	30Hz@GPU
BundleFusion off-line	**0.6**	**0.4**	**0.6**	1.1	1.6	1.1	2.2	**1.2**	offline@GPU
CPA-SLAM	0.7	0.6	8.9	**0.9**	1.8	1.4	2.5	1.6	30Hz@GPU
DVO SLAM	10.4	2.9	19.1	15.2	2.1	1.8	3.5	1.8	30Hz@CPU
ORB SLAM 2	0.8	5.8	2.9	5.4	**1.6**	**0.4**	**1.0**	1.9	30Hz@CPU
GC-SLAM sparse	0.7	0.8	1.1	1.4	2.1	1.3	2.7	1.8	**50Hz@CPU**
GC-SLAM dense	**0.6**	0.6	0.8	1.0	1.9	1.1	2.6	1.6	**25Hz@CPU**

that ORBSLAM2 contains four threads: Tracking, Mapping, Loop and BA. While GC-SLAM merely requires two threads: Tracking and Optimization. Loop closure detection is accomplished in the Optimization thread in GC-SLAM. In particular, it takes 1600 ms to accomplish one full BA in ORBSLAM2, while merely 10 ms for FastGO.

In particular, the Sparse GC-SLAM can run at 100 Hz for the front-end thread while retaining competitive accuracy, indicating that the proposed FastGO can be easily applied on portable devices with limited computational resources, as illustrated in Tab. III.

C. Scalability Evaluations

To verify the scalability of FastGO, we run GC-SLAM on a large-scale dataset which is obtained by repeating the fr3office dataset 4 times (9952 RGBD images in total). Note that in practice, when visiting a previously visited place, we do not need to add extra keyframes to keep the compact representation of the environment. In this experiment, we want to verify the performance of FastGO on large-scale datasets so the new keyframes are continuously added. The runtime of the BA solver in FastGO is presented in Fig. 4. 600 keyframes are selected in total and the computation time required by FastGO grows roughly linearly with the number of keyframes to be optimized. Note that in GC-SLAM, the BA solver is the only step for which the complexity is determined by the number of keyframes in the global map, while the other steps are running at a constant complexity. For ORBSLAM2, a global bundle adjustment on the fr3office dataset (2488 RGBD images) requires 1600 ms, never mind the extended dataset.

D. Robustness Evaluations

Robustness is critical to the practical usage of visual SLAM algorithms. In practice, different places may have similar appearance which will influence the pose estimations significantly, making the SLAM system fragile in real-world applications. To evaluate the robustness of the proposed GC-SLAM, we further run the algorithm on more challenging dataset: AUG ICL-NUIM [8] which has complex camera trajectories and a realistic noise model. In particular, there exits many places that have the same appearance in this dataset, e.g., the same computer and desk appear many times at different places in Off.1 and Off.2 dataset. GC-SLAM is compared with state-of-the-art algorithms: online method ElasticFusion [31] that requires a high-end GPU for computing, and offline methods including SUN3D [32], Choi2015 [8] and Lee2017 [33]. As shown in Tab. IV, GC-SLAM achieves comparable accuracy as state-of-the-arts, yet at the expense of much less computation resources.

VI. Conclusion

We have presented FastGO for real-time globally consistent visual localization, where the alignment error of all correspondences is efficiently minimized on the Lie manifold online thanks to the pre-integration technique used in Sec. III-B. To demonstrate the accuracy, efficiency and robustness of FastGO, GC-SLAM is presented as a RGBD SLAM system achieving state-of-the-art accuracy running at $50 - 100$ Hz on a CPU device, showing potential for portable devices with limited computational resources.

Limitations: Due to the pre-integration of FastGO as introduced in Section III-B, robust estimators such as Huber norm cannot be employed directly in the energy function. The Huber norm helps to improve pose estimation accuracy by reducing the influence of outliers, which will also be considered in the future work for further improvements of FastGO.

Future Work: we will consider working towards modular multi-sensor fusion. Specifically, additional sensors such as Inertial Measurement Units (IMU) can be combined with visual measurements and achieve more robust pose estimations. Following the framework of the proposed FastGO approach, the visual and IMU observations can be combined in a modular fashion following a Bayesian framework. [34] achieves more accurate point cloud registration by locally parameterizing the point cloud with a virtual camera. Such strategy inspires us to employ it for better correspondence collection instead of using the ICP registration in our future work.

References

[1] Peter Henry, Michael Krainin, Evan Herbst, Xiaofeng Ren, and Dieter Fox, “Rgb-d mapping: Using depth cameras for dense 3d modeling of indoor environments.,” in *ISER*, 2010, vol. 20, pp. 22–25.

[2] Feng Lu and Evangelos Milios, “Globally consistent range scan alignment for environment mapping,” *Autonomous robots*, vol. 4, no. 4, pp. 333–349, 1997.

[3] Lei Han and Lu Fang, “Mild: Multi-index hashing for appearance based loop closure detection,” in *Multimedia and Expo (ICME), 2017 IEEE International Conference on*. IEEE, 2017, pp. 139–144.

[4] Dorian Gálvez-López and Juan D Tardos, “Bags of binary words for fast place recognition in image sequences,” *IEEE Transactions on Robotics*, vol. 28, no. 5, pp. 1188–1197, 2012.

[5] Frank Dellaert and Michael Kaess, “Square root sam: Simultaneous localization and mapping via square root information smoothing,” *The International Journal of Robotics Research*, vol. 25, no. 12, pp. 1181–1203, 2006.

[6] David M Rosen, Luca Carlone, Afonso S Bandeira, and John J Leonard, “Se-sync: A certifiably correct algorithm for synchronization over the special euclidean group,” *arXiv preprint arXiv:1612.07386*, 2016.

TABLE II: Efficiency evaluations between ORBSLAM2(left) and GC-SLAM sparse(right)(mean ±2 std)

Thread (ORBSLAM2)	Steps	Time required (ms)
Tracking	ORB Extraction	11.48 ± 1.84
	Pose Prediction	2.65 ± 1.28
	Local Map Tracking	9.78 ± 6.42
	New Keyframe Decision	1.58 ± 0.92
	Total	25.58 ± 9.76

Thread (ORBSLAM2)	Steps	Time required (ms)
Mapping	Keyframe Insertion	11.36 ± 5.04
	Map Point Culling	0.25 ± 0.10
	Map Point Creation	53.99 ± 23.62
	Local BA	196.67 ± 213.42
	Keyframe Culling	6.69 ± 8.24
	Total	267.33 ± 245.10

Thread (ORBSLAM2)	Steps	Time required (ms)
Loop	Database Query	2.63 ± 2.26
	SE3 Estimation	0.66 ± 1.68
	Loop Fusion	298.45
	Essential Graph Opt.	281.99
	Total	598.70

Thread (ORBSLAM2)	Steps	Time required (ms)
BA	Full BA	1640.96
	Map Update	5.62
	Total	1793.02

Thread (GC-SLAM Sparse)	Steps	Time required (ms)
Tracking	ORB Extraction	10.01 ± 0.76
	Outlier Filtering	0.4 ± 0.01
	Frame Pair Registration	5.25 ± 0.65
	Structure Pre-Integration	0.05 ± 0.01
	Total	17.45 ± 1.76

Thread (GC-SLAM Dense)	Steps	Time required (ms)
Tracking	ORB Extraction	10.01 ± 0.76
	Sparse Registration	5.25 ± 0.65
	Dense Registration	20.31 ± 6.8
	Structure Pre-Integration	0.50 ± 0.15
	Total	38.24 ± 8.71

Thread (GC-SLAM Sparse & Dense)	Steps	Time required (ms)
Opt.	Loop Closure Detection	4.16 ± 1.83
	Map Registration	30.12 ± 8.30
	FastGO	8.86 ± 5.32
	Total	43.35 ± 14.59

TABLE III: Evaluations on the performance of FastGO influenced by the number of features

Number of Features	ORB Extraction (ms)	Feature Matching (ms)	Frame-Pair Registration (ms)	Total Tracking (ms)	Mean accuracy on kt0 (cm)
500	7.85	0.1	2.0	9.95	0.85
800	8.63	0.9	4.7	14.6	0.77
1000	9.6	1.2	5.6	17.4	0.74
2000	12.3	1.9	8.2	22.4	0.66
5000	18.5	6.4	9.5	34.4	0.63

[7] Jesus Briales and Javier Gonzalez-Jimenez, "Cartan-sync: Fast and global se (d)-synchronization," *IEEE Robotics and Automation Letters*, vol. 2, no. 4, pp. 2127–2134, 2017.

[8] Sungjoon Choi, Qian-Yi Zhou, and Vladlen Koltun, "Robust reconstruction of indoor scenes," in *Computer Vision and Pattern Recognition (CVPR), 2015 IEEE Conference on*. IEEE, 2015, pp. 5556–5565.

[9] Qian-Yi Zhou, Jaesik Park, and Vladlen Koltun, "Fast global registration," in *European Conference on Computer Vision*. Springer, 2016, pp. 766–782.

[10] Angela Dai, Matthias Nießner, Michael Zollöfer, Shahram Izadi, and Christian Theobalt, "BundleFusion: Real-time Globally Consistent 3D Reconstruction using On-the-fly Surface Re-integration," *arXiv preprint arXiv:1604.01093*, 2016.

[11] Bill Triggs, Philip F McLauchlan, Richard I Hartley, and Andrew W Fitzgibbon, "Bundle adjustmenta modern synthesis," in *International workshop on vision algorithms*. Springer, 1999, pp. 298–372.

[12] Manolis IA Lourakis and Antonis A Argyros, "Sba: A software package for generic sparse bundle adjustment," *ACM Transactions on Mathematical Software (TOMS)*, vol. 36, no. 1, pp. 2, 2009.

[13] Raul Mur-Artal and Juan D Tardos, "Orb-slam2: an open-source slam system for monocular, stereo and rgb-d cameras," *arXiv preprint arXiv:1610.06475*, 2016.

[14] Christian Kerl, Jurgen Sturm, and Daniel Cremers, "Dense visual slam for rgb-d cameras," in *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on*. IEEE, 2013, pp. 2100–2106.

[15] Christopher Zach, "Robust bundle adjustment revisited," in *European Conference on Computer Vision*. Springer, 2014, pp. 772–787.

[16] Yekeun Jeong, David Nister, Drew Steedly, Richard Szeliski, and In-So Kweon, "Pushing the envelope of modern methods for bundle adjustment," *IEEE transactions on pattern analysis and machine intelligence*, vol. 34, no. 8, pp. 1605–1617, 2012.

[17] Kunal N Chaudhury, Yuehaw Khoo, and Amit Singer, "Global registration of multiple point clouds using semidefinite programming," *SIAM Journal on Optimization*, vol. 25, no. 1, pp. 468–501, 2015.

[18] Federica Arrigoni, Beatrice Rossi, and Andrea Fusiello, "Global registration of 3d point sets via lrs decomposition," in *European Conference on Computer Vision*. Springer, 2016, pp. 489–504.

[19] Shankar Krishnan, Pei Yean Lee, John B Moore, Suresh Venkatasubramanian, et al., "Global registration of multiple 3d point sets via optimization-on-a-manifold.," in *Symposium on Geometry Processing*, 2005, pp. 187–196.

[20] José-Luis Blanco, "A tutorial on se (3) transformation parameterizations and on-manifold optimization," *University of Malaga, Tech. Rep*, vol. 3, 2010.

TABLE IV: Evaluations on the performance of GC-SLAM on AUG ICL-NUIM dataset (cm)

	ElasticFusion	SUN3D	Choi2015	Lee2017	GC-SLAM Sparse	GC-SLAM dense
$Liv.1$	59.02	32.22	9.87	9.49	10.6	**8.00**
$Liv.2$	37.09	29.13	13.63	12.18	8.89	**6.99**
$Off.1$	18.29	50.84	**6.22**	9.95	9.85	8.7
$Off.2$	27.18	29.75	8.89	**6.93**	11.0	9.75
$average$	35.39	35.49	9.65	9.63	10.1	**8.36**

[21] Deval Keralia, KK Vyas, and Khushali Deulkar, "Google project tango–a convenient 3d modeling device," *International Journal of Current Engineering and Technology*, vol. 4, no. 5, pp. 3139–3142, 2014.

[22] Henry Chen, Austin S Lee, Mark Swift, and John C Tang, "3d collaboration method over hololens and skype end points," in *Proceedings of the 3rd International Workshop on Immersive Media Experiences*. ACM, 2015, pp. 27–30.

[23] Lei Han, Gu-yue Zhou, Lan Xu, and Lu Fang, "Beyond sift using binary features in loop closure detection," in *Intelligent Robots and Systems (IROS), 2017 IEEE/RSJ International Conference on*. IEEE, 2017.

[24] Bertram Drost, Markus Ulrich, Nassir Navab, and Slobodan Ilic, "Model globally, match locally: Efficient and robust 3d object recognition," in *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*. Ieee, 2010, pp. 998–1005.

[25] Nicolas Mellado, Dror Aiger, and Niloy J Mitra, "Super 4pcs fast global pointcloud registration via smart indexing," in *Computer Graphics Forum*. Wiley Online Library, 2014, vol. 33, pp. 205–215.

[26] David W Eggert, Adele Lorusso, and Robert B Fisher, "Estimating 3-d rigid body transformations: a comparison of four major algorithms," *Machine vision and applications*, vol. 9, no. 5-6, pp. 272–290, 1997.

[27] Ankur Handa, Thomas Whelan, John McDonald, and Andrew J Davison, "A benchmark for rgb-d visual odometry, 3d reconstruction and slam," in *Robotics and automation (ICRA), 2014 IEEE international conference on*. IEEE, 2014, pp. 1524–1531.

[28] Jürgen Sturm, Nikolas Engelhard, Felix Endres, Wolfram Burgard, and Daniel Cremers, "A benchmark for the evaluation of rgb-d slam systems," in *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*. IEEE, 2012, pp. 573–580.

[29] Lingni Ma, Christian Kerl, Jörg Stückler, and Daniel Cremers, "Cpa-slam: Consistent plane-model alignment for direct rgb-d slam," in *Robotics and Automation (ICRA), 2016 IEEE International Conference on*. IEEE, 2016, pp. 1285–1291.

[30] Rainer Kümmerle, Giorgio Grisetti, Hauke Strasdat, Kurt Konolige, and Wolfram Burgard, "g 2 o: A general framework for graph optimization," in *Robotics and Automation (ICRA), 2011 IEEE International Conference on*. IEEE, 2011, pp. 3607–3613.

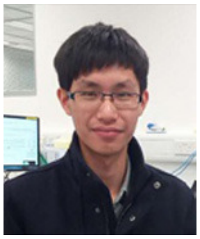
[31] Thomas Whelan, Stefan Leutenegger, R Salas-Moreno, Ben Glocker, and Andrew Davison, "Elasticfusion: Dense slam without a pose graph," Robotics: Science and Systems, 2015.

[32] Jianxiong Xiao, Andrew Owens, and Antonio Torralba, "Sun3d: A database of big spaces reconstructed using sfm and object labels," in *Computer Vision (ICCV), 2013 IEEE International Conference on*. IEEE, 2013, pp. 1625–1632.

[33] Jeong-Kyun Lee, Jaewon Yea, Min-Gyu Park, and Kuk-Jin Yoon, "Joint layout estimation and global multi-view registration for indoor reconstruction," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 162–171.

[34] Jaesik Park, Qian-Yi Zhou, and Vladlen Koltun, "Colored point cloud registration revisited," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 143–152.

Lei Han Lei Han studied Electrical Engineering at the Hong Kong University of Science and Technology (China) and Tsinghua University (China). He received the B.S. degree in July 2013 and joined the Department of Electrical Computing Engineering at the Hong Kong University of Science and Technology in September 2016, where he is pursuing the PhD degree. His current research focuses on multi-view geometry and 3D computer vision.



Lan XU received the B.S. degree in Department of Information and Communication, College of Information Science & Electronic Engineering, Zhejiang University, Hangzhou, China, in 2015. He is currently pursuing the Ph.D. degree in the Department of Electronic and Computer Engineering, Hong Kong University of Science and Technology, Hong Kong SAR.

Dmytro Bobkov Dmytro Bobkov studied Electrical Engineering at the Technical University of Munich (Germany) and National Technical University of Ukraine, Kyiv (Ukraine). He received the degree M.Sc. in October 2012. After this he joined the Chair of Media Technology at the Technical University of Munich in April 2013, where he is working as a member of the research staff and pursuing the PhD degree. His current research focuses on 3D computer vision and machine learning.

Eckehard Steinbach (IEEE M96, SM08, F15) studied Electrical Engineering at the University of Karlsruhe (Germany), the University of Essex (Great-Britain), and ESIEE in Paris. From 1994-2000 he was a member of the research staff of the Image Communication Group at the University of Erlangen-Nuremberg (Germany), where he received the Engineering Doctorate in 1999. From February 2000 to December 2001 he was a Postdoctoral Fellow with the Information Systems Laboratory of Stanford University. In February 2002 he joined the Department of Electrical and Computer Engineering of the Technical University of Munich (Germany), where he is currently a Full Professor for Media Technology. His current research interests are in the area of haptic and visual communication, teleoperation over the Tactile Internet, indoor mapping and localization.

Qionghai Dai received the M.S. and Ph.D. degrees in computer science and automation from Northeastern University, Shenyang, China, in 1994 and 1996, respectively. He is currently a professor in the Department of Automation and is the Director of the Broadband Networks and Digital Media Laboratory at Tsinghua University, Beijing. He has authored or co-authored over 200 conference and journal papers and two books. His research interests include computational photography and microscopy, computer vison and graphics, intelligent signal processing. He is associate Editor of JVCI, IEEE TNNLS and IEEE TIP.

Lu FANG is currently an Associate Professor in Tsinghua-Berkeley Shenzhen Institute (TBSI). She received Ph.D from Hong Kong University of Science and Technology in 2011, and B.E. from University of Science and Technology of China in 2007, respectively. Dr. Fang's research interests include computational photography and visual computing. Dr. Fang used to receive NSFC Excellent Young Scholar Award, Best Student Paper Award in ICME 2017, Finalist of World?s First 10K Best Paper Award in ICME 2017, Finalist of Best Paper Award in ICME 2011 etc. Dr. Fang is currently IEEE Senior Member.