iDFusion: Globally Consistent Dense 3D Reconstruction from RGB-D and Inertial Measurements

Dawei Zhong*
Tsinghua University

Lei Han*
Hong Kong Univ. of Sci. and Tech.

Lu Fang
Tsinghua University

ABSTRACT

We present a practical (*fast, globally consistent* and *robust*) dense 3D reconstruction system, iDFusion, by exploring the joint benefit of both the visual (RGB-D) solution and inertial measurement unit (IMU). A global optimization considering all the previous states is adopted to maintain high localization accuracy and global consistency, yet its complexity of being linear to the number of all previous camera/IMU observations seriously impedes real-time implementation. We show that the global optimization can be solved efficiently at the complexity linear to the number of keyframes, and further realize a real-time dense 3D reconstruction system given the estimated camera states. Meanwhile, for the sake of robustness, we propose a novel loop-validity detector based on the estimated bias of the IMU state. By checking the consistency of camera movements, a false loop closure constraint introduces manifest inconsistency between the camera movements and IMU measurements. Experiments reveal that iDFusion owns superior reconstruction performance running in 25 fps on CPU computing of portable devices, under challenging yet practical scenarios including texture-less, motion blur, and repetitive contents.

CCS CONCEPTS

• **Human-centered computing** → **Virtual reality**.

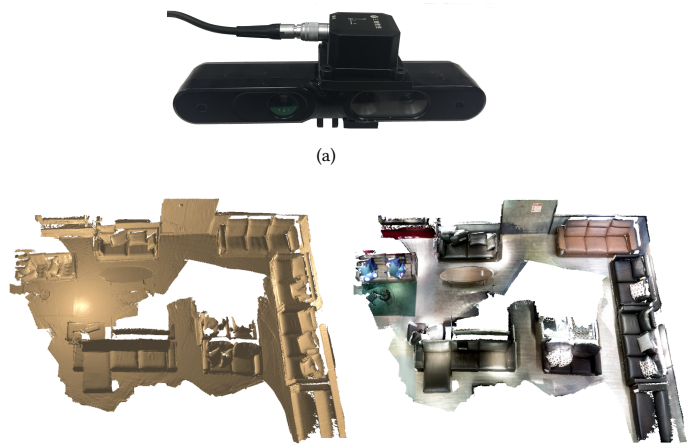
KEYWORDS

Visual-IMU global optimization, real-time SLAM, 3D reconstruction, loop closure

ACM Reference Format:
Dawei Zhong, Lei Han, and Lu Fang. 2019. iDFusion: Globally Consistent Dense 3D Reconstruction from RGB-D and Inertial Measurements. In *Proceedings of the 27th ACM International Conference on Multimedia (MM '19), October 21–25, 2019, Nice, France.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3343031.3351085

1 INTRODUCTION

Globally consistent dense 3D reconstruction has attracted a lot of attention as a fundamental component towards spatial AI computing [1]. With the popularity of RGBD sensors, various works [2–5] have been presented for solving this problem by optimizing the registration error of RGBD scans. However, based only on visual information, texture-less environments and motion blur may lead to failure of reconstruction or unpleasing artifacts. On the other hand, visual inertial navigation systems achieve impressive accuracy by combining visual and inertial observations together for camera state estimation. Most of the visual-inertial fusion systems are based on a Kalman filter or window-based non-linear optimization for scalability. They achieve high local accuracy yet cannot guarantee global consistency, due to the marginalization step employed in these methods.



(a)

(b)

Figure 1: (a) experimental device: RGB-D sensor (ASUS Xtion Pro) and IMU (SC-AHRS-100D2), (b) real-time reconstruction result of iDFusion on portable device[1].

In this paper, we aim for a '**practical**' dense 3D reconstruction system, which is expected to be *robust*, *globally consistent* and *real-time*. We propose iDFusion to explore the joint benefit, i.e., global consistency and high local accuracy, from visual observations and inertial measurements. We formulate the pose estimation problem as a joint optimization of all existing observations from both the camera and IMU. Such a formulation assures the globally consistent estimation of camera poses. We propose to directly solve it using Max-A-Posteriori estimation, which leads to a nonlinear Gauss-Newton optimization problem. Note that this nonlinear optimization suffers high computational complexity, as it requires traversal search of each visual and IMU observation. For the sake of efficiency, we further reduce the complexity of the visual part from the number of visual observations to the number of keyframes given the depth information of each point, as shown in FastGO [6]. Meanwhile, the high frequency inertial measurements can be grouped using the IMU pre-integration technique [7]. As a result, unlike previous

*Equal contribution. Correspondence author Lu Fang(fanglu@sz.tsinghua.edu.cn)

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
MM '19, October 21–25, 2019, Nice, France
© 2019 Association for Computing Machinery.
ACM ISBN 978-1-4503-6889-6/19/10…$15.00
https://doi.org/10.1145/3343031.3351085

[1]Microsoft surface pro5

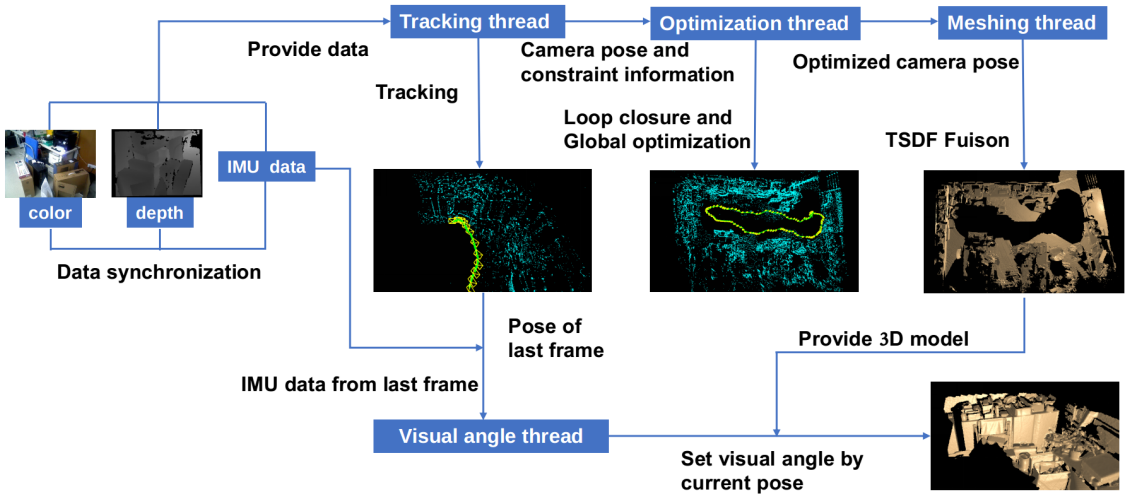


Figure 2: Illustration of system framework of iDFusion. The multi-threading architecture achieves modularization and real-time capacity.

sliding window-based optimization or filtering-based methods, we are able to optimize all previous observations in real-time, achieving state-of-the-art performance on public datasets, as shown in Sec. 4.1.

Moreover, loop closure [8] serves to eliminate the accumulated localization drift. However, loop closure from visual observations introduces a false positive loop easily when two places share the same appearance, leading to wrong camera pose estimation and an inconsistent reconstructed 3D model. It is hard to distinguish false positives from true positives, as either accumulated drift from frame-to-frame registration or false positives may lead to inconsistent pose estimation.

Contrary to previous methods [3, 9], which employ robust cost functions to suppress the impact of false loop closures with the increase of computational complexity, we propose a novel loop-validity detector, which is based on IMU bias states. It employs the estimated bias of the IMU state to check if loop closure observations are consistent. When a new loop closure constraint is introduced, a truly positive loop closure will maintain previous bias estimations, while the false positive loop closure will influence them significantly to minimize the global energy function.

Given the globally consistent camera pose estimations, we use truncated signed distance field (TSDF) [10] at sub-centimeter resolution to fuse depth observations, where the VoxelHashing [11] technique is used for scalability, and FlashFusion [5] is used for efficiency. Experiments demonstrate that the presented iDFusion achieves real-time, robust, and globally consistent dense 3D reconstruction on portable devices, for various scenes including textureless or motion blur situations. Technical contributions can be summarized as follows:

- *A real-time solver for the full optimization of visual and inertial observations.* Pose estimation is formulated as a nonlinear optimization problem with all previous observations and states, assuring global consistency. For the sake of efficiency, we represent visual observations as point-to-point and plane-to-plane registrations, which could be efficiently optimized via FastGO [6], and IMU measurements group at keyframe rate to reduce the states to be optimized.
- *Online autonomous calibration of Cam-IMU transformation for plug-and-play usages of inertial sensors.* The initial transformation matrix is obtained by the visual initialization. Then the Cam-IMU transformation participates in the global optimization all the time, which avoids the error accumulation caused by the inaccuracy of the initial estimation.
- *A robust loop-validity detector based on IMU bias states.* When a new loop closure constraint is introduced, if it is truly positive, the previous bias estimations remain unchanged. While if it is false positive, to minimize the global energy function, the previous bias estimations will be affected significantly.

2 RELATED WORK

Dense 3D reconstruction serves as a fundamental part of 3D machine perception and benefits various applications including virtual reality and mobile robots. Camera pose estimation (localization) and environment mapping (reconstruction) are two critical components to achieve this target. In this paper, the observations from a RGB-D camera and inertial sensor are jointly optimized for globally consistent and robust localization, and we also achieve real-time dense 3D reconstruction.

Visual-Inertial Localization: [12, 13] achieves monocular camera localization. To achieve better localization, combining inertial measurements and visual measurements has been investigated for years [14, 15]. Most of the previous works can be divided into filter-based methods [16–18] or optimization-based approaches [19–21]. Filter-based methods mainly use the extended Kalman filter technique to fuse visual inertial measurements, which is fast but will lead to loss of information to some extent. On the other hand, optimization methods are more computationally expensive but utilize

all the information available and lead to higher accuracy. For real-time implementation, optimization-based approaches can only be utilized in a sliding window, thus cannot achieve global consistency. For high quality dense 3D reconstruction, global optimization of pose estimations is important however are neglected by previous methods due to the heavy computational burden. In this paper, we propose a real-time solver for global optimization of all previous visual and inertial observations for globally consistent pose estimation.

Dense 3D reconstruction system: The RGBD camera is widely used in dense 3D reconstruction systems because of its capacity of providing accurate depth directly. ElasticFusion [4] uses dense frame-to-model camera tracking and windowed surfel-based fusion. Without inertial information, the tracking is prone to failure. DPI-SLAM [22] uses a loosely-coupled IMU constraint with the visual estimation and uses point clouds for mapping. The loosely-coupled optimization causes loss of partial accuracy and point clouds cannot represent an elaborate 3D model. RGB-D-Inertial SLAM [23] achieves real-time reconstruction and tightly-coupled visual-inertial optimization but it essentially requires a GPU. It uses point clouds for mapping too. [24] uses an extended Kalman filter to fuse the visual and inertial information and achieves stable tracking under fast camera motions. However, the filter method does not optimize the former camera poses and will not provide a consistent optimum estimation. In this paper, we present iDFusion, which achieves CPU-based real-time dense reconstruction. It uses global tightly-coupled visual-inertial optimization and achieves a real-time solver by simplifying the FastGO global optimization and extending it to combine IMU constraints. It realizes spatial hashing-based volumetric fusion [11, 25] and TSDF fusion [26] to achieve real-time dense elaborate reconstruction.

3 IDFUSION

Our system framework is illustrated in Fig. 2. First, the camera and IMU data are used for tracking to obtain the camera pose. Then, all information is used in the global optimization to get the optimal estimation including the Cam-IMU transformation. The loop closure is validated to drop the wrong matching result. At the same time, depending on the camera poses and depth information, we can get the 3D model using TSDF fusion. The multi-threading architecture accelerates the speed practically. In the following, we elaborate on the core components of iDFusion, namely full optimization for globally consistent pose estimation in Sec. 3.1, camera-IMU extrinsic calibration and gravity optimization in Sec. 3.2, and robust loop-validity detector in Sec. 3.3.

3.1 Globally Consistent Pose Estimation

Globally consistent pose estimation plays a vital role in large-scale high quality dense 3D reconstruction. We propose to jointly minimize the alignment error of feature correspondence, plane correspondence and IMU measurements from all previous observations. As illustrated in Fig. 3, the IMU pre-integration provides status constraint between relevant keyframes, and visual measurements provide constraint between corresponding keyframes. Aligning IMU measurement with visual structure, we can optimize the camera pose, transformation matrix and gravity together. For better

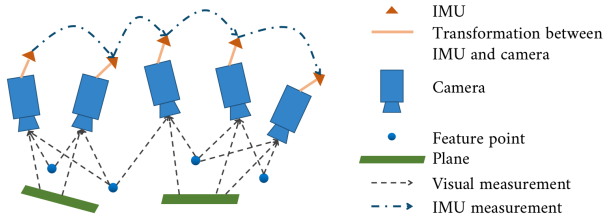


Figure 3: Illustration of our optimization framework. Constraint of visual and inertial measurement is shown by dotted lines.

presentation, we introduce the math notations firstly. The i_{th} frame is denoted as F_i, including RGB image I_i and depth image D_i. IMU measurements are denoted as $\tilde{\boldsymbol{\omega}}$ and $\tilde{\boldsymbol{\alpha}}$ for angular velocity and acceleration, respectively. Sequential IMU measurements between nearby frames are integrated to obtain IMU constraint $C^I_{i,i+1}$. The keyframe strategy is adopted, i.e., only the status of keyframes is applied for global optimization while other frames are adjusted based on its nearest previous keyframe. For the current keyframe, we use loop closure detection [8] to find candidate keyframes that can be registered with each other. Feature correspondence and plane correspondence ($C^F_{i,j}$ and $C^P_{i,j}$) are extracted from successfully registered frame pairs. All the collected frame correspondences are denoted as Ω.

For frame F_i, we denote its state vector as R_{ci}, P_{ci}, V_i, b^g_i, b^a_i, indicating the rotation, translation, velocity, bias of angular velocity, and bias of acceleration, respectively. R_{Ii} and P_{Ii} is the rotation and translation of IMU coordinate, respectively. R_0 and P_0 denote the Cam-IMU transformation matrix. g is the initial gravity, and R_g is the optimized direction of gravity.

A large scale scene will gradually amplify the tiny deviation, so the pre-calibrated Cam-IMU transformation and settled gravity obtained by the initialization process will cause the error accumulation. On the other hand, the cam-IMU transformation and direction of gravity obtained by the global optimization will be increasingly accurate, which can achieve global consistency and avoid error accumulation. Thus, we estimate the frame state R_{ci}, P_{ci}, V_{Ii}, b^g_i, b^a_i, the Cam-IMU transformation R_0 and P_0 and the direction of gravity R_g in one global function. R_0, P_0 and R_g are global variables. For simplicity, we use $\mathbf{S}$ to denote the status of all keyframes and global variables: $\mathbf{S} = \{\mathbf{R}, \mathbf{P}, \mathbf{V}, \mathbf{b^g}, \mathbf{b^a}, R_0, P_0, R_g\}$, where $\{\mathbf{R}, \mathbf{P}, \mathbf{V}, \mathbf{b^g}, \mathbf{b^a}\} = \{R_{ci}, P_{ci}, V_{Ii}, b^g_i, b^a_i, i = 0, \cdots, N-1\}$. Camera status and global variables are estimated based on

$$\mathbf{S}^* = \arg\min_{\mathbf{S}} \sum_{i=0}^{N-1} C^I_{i,i+1}(\mathbf{S}) + \sum_{(i,j)\in\Omega} [\lambda_1 C^P_{i,j}(\mathbf{S}) + \lambda_2 C^F_{i,j}(\mathbf{S})]. \quad (1)$$

Feature Constraint. For each frame pair(F_i, F_j), the corresponding points $C_{i,j} = \left\{C^k_{i,j} = \left(p^k_i, p^k_j\right)\right\}$ are collected from sparse feature matching, where P^k_i represents the k-th point observed in the local coordinates of the i-th frame. For the i-th frame, T_{ci} is the rigid transformation in Euclidean space, which can be represented by Lie algebra ξ_i on the SE3 manifold. R_{ci} and P_{ci} is the rotation matrix and the translation vector, which satisfies $T_i = [R_{ci}|P_{ci}]$. After finding the corresponding ORB features in a frame pair, we

can scheme the following residual for every corresponding point:

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

Combining all the Jacobian matrices of corresponding points, we have the feature correspondence constraint:

$$C_{i,j}^F = \sum_{k=0}^{\|C_{i,j}\|-1} \left\| T_i P_i^k - T_j P_j^k \right\|^2, \tag{3}$$

where P_i^k is the homogeneous coordinate of the local 3D point p_i^k. The pose of the first frame T_0 is initialized as the world coordinate. N represents the total number of collected frames. For visual initialization, pose estimation can be obtained by minimizing Eq. (3).

Plane Constraint. We adopt the agglomerative hierarchical clustering (AHC) plane extracting algorithm [27] for efficient plane detection in point clouds. The plane equation is

$$X^T P_c + D = 0, \tag{4}$$

where X is the unit normal vector, D is the distance from the origin to the plane, and P_c represents the point observed in the local coordinate. For each plane segment, we use the least squares plane fitting to estimate its plane parameters. If the pose of the local frame is R, P, the plane equation in the world frame can be obtained by the following transformation:

$$(RX)^T P_w + [D - (RX)^T T] = 0, \tag{5}$$

where P_w represents the point observed in the world coordinate.

The plane parameter in the i-th camera frame is X_i, D_i, and the pose of the local frame is R_i, P_i. So the plane parameter in the world frame is $m_i R_i X_i$ (normal vector) and $m_i[D_i - (R_i X_i)^T T_i]$ (distance). m_i is given by

$$m_i = \begin{cases} 1, & D_i - (R_i X_i)^T T_i > 0, \\ -1, & D_i - (R_i X_i)^T T_i < 0, \end{cases} \tag{6}$$

which is used to ensure that the distance is positive. $C_{P_{i,j}}^k$ is the k-th plane, which is simultaneously observed by frame i and frame j. The corresponding planes $C_{P_{i,j}}$ are collected by the plane parameter comparison. After finding the corresponding planes in a frame pair, the residual of the corresponding planes is

$$\begin{aligned} r_{N_{i,j}}^k &= m_i^k R_{ci} X_i^k - m_j^k R_{cj} X_j^k, \\ r_{D_{i,j}}^k &= m_i^k [D_i^k - (R_{ci} X_i^k)^T P_{ci}] - m_j^k [D_j^k - (R_{cj} X_j^k)^T P_{cj}]. \end{aligned} \tag{7}$$

Combining all the residuals of corresponding planes, we have the plane constraint:

$$C_{i,j}^P = \sum_{k=0}^{\|C_{P_{i,j}}\|-1} \left(\left\| r_{N_{i,j}}^k \right\|^2 + \left\| r_{D_{i,j}}^k \right\|^2 \right). \tag{8}$$

The plane observation Jacobian matrix is in the supplementary materials.

IMU Constraint. For the adjacent keyframes F_i and F_j, we can get its pose R_{ci}, P_{ci} and R_{cj}, P_{cj} by the corresponding points. Also we can initialize V_{Ii} using the speed estimation. The Cam-IMU transformation matrix is $\boldsymbol{R_0}, \boldsymbol{P_0}$. Then, we have the transformation from the camera to the IMU as follows:

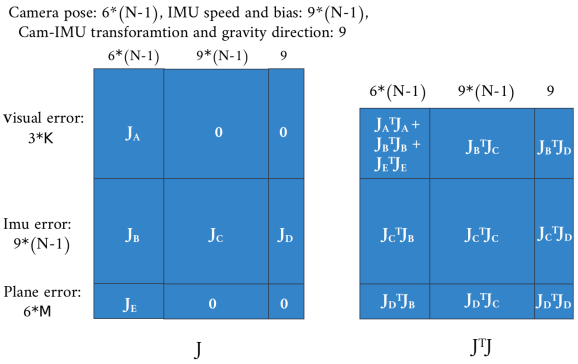


Figure 4: Jacobian in the Gauss-Newton function. The number of corresponding points is K. The number of keyframes is N and the number of corresponding planes is M. Jacobian of feature points constraint is J_A. Jacobian of IMU constraint is J_B, J_C, J_D. Jacobian of plane constraint is J_E.

$$\begin{aligned} R_{Ii} &= R_{ci} R_0, \\ P_{Ii} &= P_{ci} + R_{ci} P_0. \end{aligned} \tag{9}$$

After getting the pose of the IMU, we can write the expression of relative pose variation between the adjacent keyframes:

$$\begin{aligned} \Delta R_{ij} &= R_{Ii}^T R_{Ij}, \\ \Delta V_{ij} &\doteq R_{Ii}^\top \left(V_{Ij} - V_{Ii} - g\Delta t_{ij} \right), \\ \Delta P_{ij} &\doteq R_i^\top \left(P_{Ij} - P_{Ii} - V_{Ii}\Delta t_{ij} - \frac{1}{2} \mathbf{g} \Delta t_{ij}^2 \right). \end{aligned} \tag{10}$$

Meanwhile, IMU data is integrated for getting the relative motion variation $\Delta\tilde{R}_{ij}, \Delta\tilde{V}_{ij}$ and $\Delta\tilde{P}_{ij}$ between the adjacent keyframes. The relative motion variation is independent of the camera pose so there is no need to repeat the integration when the pose estimation changes. Aligning relative motion variation integrated by the IMU measurement with the visual structure, we can get the following residuals:

$$\begin{aligned} r_{\triangle R_{ij}} &= Log(\Delta R_{ij} \Delta\tilde{R}_{ij}), \\ r_{\triangle V_{ij}} &= \Delta V_{ij} - \Delta\tilde{V}_{ij}, \\ r_{\triangle P_{ij}} &= \Delta P_{ij} - \Delta\tilde{P}_{ij}. \end{aligned} \tag{11}$$

Combining these terms, the IMU constraint is given by the following equation:

$$C_{i,i+1}^I = \left\| r_{\triangle R_{ij}} \right\|^2 + \left\| r_{\triangle V_{ij}} \right\|^2 + \left\| r_{\triangle P_{ij}} \right\|^2. \tag{12}$$

The detailed deduction of the IMU residual and Jacobian is in the supplementary materials.

Efficient Solution. State variables estimation can be obtained by minimizing Eq. (1). We use the following standard non-linear Gauss-Newton optimization procedure to solve the optimization problem:

$$\boldsymbol{\delta} = -\left(J(\boldsymbol{\xi})^T J(\boldsymbol{\xi}) \right)^{-1} J(\boldsymbol{\xi})^T \boldsymbol{r}(\boldsymbol{\xi}). \tag{13}$$

The global optimization function is computationally complex with massive variables. To maintain the real-time capacity on portable

devices, we need to simplify the calculation process of Eq. (13) to accelerate the optimization speed. In the following, we take the calculation of $J(\boldsymbol{\xi})^T J(\boldsymbol{\xi})$ as an example to show how to simplify it. $J(\boldsymbol{\xi})^T \boldsymbol{r}(\boldsymbol{\xi})$ can be handled similarly.

First, decompose the optimization function. As illustrated in Fig. 4, the whole Jacobian can be divided into blocks, and $J(\boldsymbol{\xi})^T J(\boldsymbol{\xi})$ is the combination of their product. So the calculation of whole Jacobian can be divided into the calculation of the product of Jacobian blocks. Examining Eq. (3), (8) and (12), the computational complexity of C^F, C^P and C^I is $O(k * N_{corr})$, $O(m * N_{corr})$ and $O(N)$, respectively, where N is the number of keyframes, N_{corr} is the number of image pairs, k and m represent the average number of feature pairs and plane pairs per image pair, respectively. Given that the number of plane pairs and keyframes are much smaller than the number of feature points, the computational complexity is mainly dominated by C^F, which means the computational complexity of $J^T J$ in Fig. 4 is mainly dominated by the calculation of $J_A^T J_A$. The key problem is to accumulate the calculation of $J_A^T J_A$ while J_A is the Jacobian of the feature points. So we apply and improve FastGO [6] technology, which can get $J_A^T J_A$ directly by second-order statistics. This reduces the complexity of C^F from $O(k * N_{corr})$ to $O(N_{corr})$. The time occupation of $J^T J$ calculation will decrease notably. In the following we will show how to get $J_A^T J_A$ directly by using second-order statistics.

FastGO [6] minimizes the error of feature pairs (Eq. (2)) on the manifold of SE3 space [28]. The Jacobian matrix of $r_{i,j}^k(\boldsymbol{\xi})$ is

$$J_i^k(\xi_i) = [I_{3\times3} \quad -[T(\xi_i)P_i^k]^\wedge]. \tag{14}$$

Accordingly, the update mode of the variable is on SE3 as well:

$$exp(\xi^\wedge) = exp(\delta\xi^\wedge) \times exp(\xi^\wedge). \tag{15}$$

This is equivalent to

$$\begin{aligned} P &= P + exp(\delta\Phi^\wedge) \times \delta P, \\ R &= exp(\delta\Phi^\wedge) \times R, \end{aligned} \tag{16}$$

where R and P is the rotation and the transformation parts of T (ξ) and $\delta\Phi$ and δP is the rotation and the transformation parts of $\delta\xi$. The derivation on SE3 is unnecessary and more complex, so we propose to differentiate the R and P directly. The variable update mode becomes

$$\begin{aligned} P &= P + \delta P, \\ R &= exp(\delta\Phi^\wedge) \times R, \end{aligned} \tag{17}$$

and differentiating the R and P leads to the following Jacobian:

$$J_i^k(\xi_i) = [I_{3\times3} \quad -[R(\Phi_i)P_i^k]^\wedge]. \tag{18}$$

Compared to the original Eq. (14), there is no translation term in Eq. (18). Such a modification simplifies the calculation process and conforms to the IMU variable update mode.

For one feature point pair between frame i and frame j, we can get the following submatrix:

$$J_m = [0 \cdots J_i^k(\xi_i) \cdots 0 \cdots -J_j^k(\xi_j) \cdots 0]. \tag{19}$$

For a corresponding frame-pair $C_{i,j}$ between frame i and frame j, the Jacobian product $J_{C_{i,j}}^T J_{C_{i,j}}$ is the summation of all matching feature point submatrices :

$$J_{C_{i,j}}^T J_{C_{i,j}} = \sum_{m \in C_{i,j}} J_m^T J_m. \tag{20}$$

There are many corresponding frame pairs in the visual observation and $J_{C_{i,j}}^T J_{C_{i,j}}$ is a submatrix of the original Jacobian matrix $J_A^T J_A$, which is shown as follows:

$$J_A^T J_A = \sum J_{C_{i,j}}^T J_{C_{i,j}}. \tag{21}$$

As we can see in Eq. (19), J_m only has two non-zero 3×6 matrices, so $J_m^T J_m$ ($m \in C_{i,j}$) only has four non-zero 6×6 matrices at the same blocks. As the summation of $J_m^T J_m$ ($m \in C_{i,j}$), $J_{C_{i,j}}^T J_{C_{i,j}}$ has four non-zero $6 * 6$ matrices too and can be calculated by the second-order statistics of the structure terms. $J_A^T J_A$ is the linear summation of $J_{C_{i,j}}^T J_{C_{i,j}}$, which means we can get $J_A^T J_A$ directly by the linear combination of the second-order statistics of the structure terms. By doing so we reduce the complexity of C^F from $O(k \times N_{corr})$ to $O(N_{corr})$.

The dominated complexity of the whole $J^T J$ in Fig. 4 is significantly simplified. $J^T r$ can be handled by a similar method and then we can solve Eq. (13) efficiently using the sparse matrix solver.

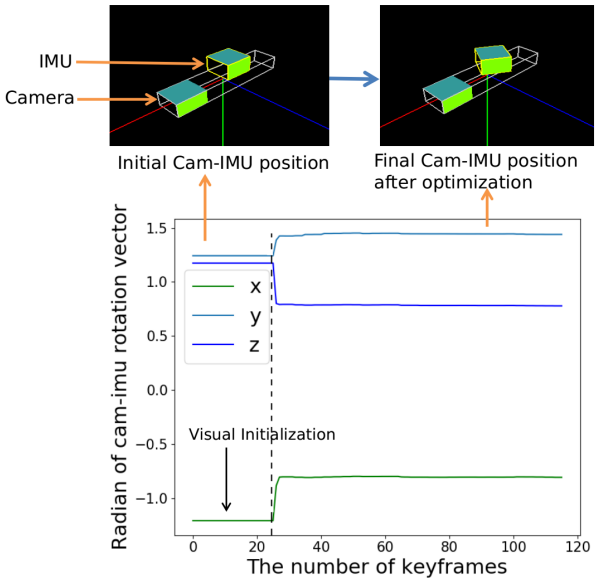


Figure 5: The Cam-IMU rotation vector converges to the accurate value rapidly after visual initialization.

3.2 Self-calibration and Gravity Optimization

An accurate Cam-IMU transformation matrix is essential for the visual-IMU optimization, as a tiny deviation will cause an accumulated error. While the transformation matrix can be calibrated off-line by calibration tools such as Kalibr [33], the off-line calibration is troublesome and conditionally sensitive, especially when the IMU is not strictly fixed with the camera. We propose an on-line self-calibration method to obtain Cam-IMU transformation. Our self-calibration optimization is tightly-coupled in the global optimization, so it will be more and more accurate with the increment

Table 1: Quantitative evaluations on public datasets in terms of absolute trajectory error (cm).

	kt0	kt1	kt2	kt3	fr1/desk	fr2/xyz	fr3/office	fr3/nst
ElasticFusion (GPU) [4]	0.9	0.9	1.4	10.6	2.0	1.1	**1.7**	1.6
BundleFusion on-line (GPU) [2]	0.8	**0.5**	1.1	**1.2**	**1.7**	1.4	2.9	**1.6**
CPA-SLAM (GPU) [29]	1.8	1.4	2.5	1.6	-	-	-	-
RGB-D-Inertial SLAM (GPU) [23]	0.9	1.2	**0.9**	1.9	-	-	-	-
RGBD SLAM (CPU) [30]	2.6	0.8	1.8	43.3	2.3	**0.8**	3.2	1.7
NIO (CPU) [31]	-	-	-	-	2.5	1.2	3.3	1.9
Non-iterative SLAM (CPU) [32]	-	-	-	-	2.5	1.1	-	1.7
FlashFusion (CPU) [5]	**0.7**	0.8	1.1	1.4	1.9	1.3	2.5	1.8
iDFusion (CPU)	**0.7**	0.7	**0.9**	**1.2**	2.0	1.0	**1.7**	1.7

of the trajectory, and the accumulated error caused by the settled transformation will be avoided. To obtain the optimization function of the transformation matrix, we use $R_{ci}R_0$, $P_{ci} + R_{ci}P_0$ to replace R_{Ii},P_{Ii} in the global optimization function. Then, the optimization variables become camera pose R_{ci},P_{ci} and camera-IMU transformation R_0, P_0. Global optimization will jointly optimize the coupled variables.

Gravity adjustment is necessary for visual-IMU optimization. The magnitude of the gravity vector can be refined by the known gravity constant, and the optimization should only adjust the direction of gravity. [23] optimizes the gravity vector directly, and [34] adjusts the gravity direction by optimizing the orthogonal basis. They both change the gravity norm and need to refine the gravity to the known gravity constant after optimization. Using R_gg (the initial value of R_g is an identity matrix) to replace g in the optimization function, we can optimize the gravity direction directly.

For the self-calibration, the optimization variables are R_0, P_0 and R_g. The optimization of R_g only changes the direction of gravity. The initial Cam-IMU transformation matrix and gravity direction may be very different with the accurate value, which may lead to non-convergence of the optimization at the very start. To solve this, we use the first 20 keyframes only for visual optimization. Then we fix the pose of the camera and optimize the velocity, Cam-IMU transformation and gravity direction, which will provide a basically accurate value as the initial value. Then we take the proposed global optimization and it will converge to an accurate result. Fig. 5 shows that the Cam-IMU rotation vector converges to the accurate value rapidly after visual initialization.

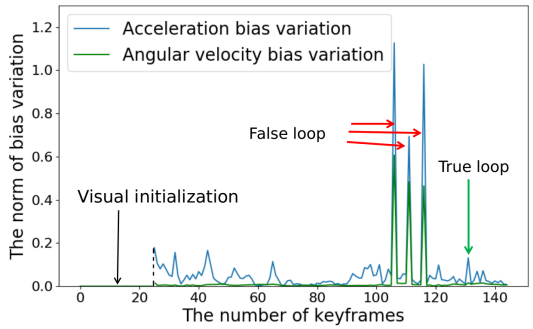


Figure 6: The norm of bias variation. False loop causes severe bias variation while true loop will not lead to sudden change.

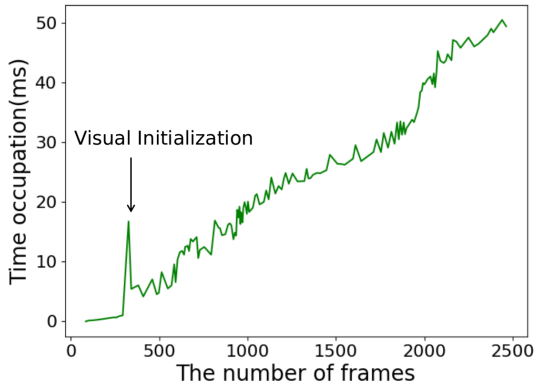


Figure 7: Efficiency evaluation on TUM dataset fr3/officet [35]. The time occupation of global optimization is in linear growth with the increment of frame amount.

3.3 Robust Loop-validity Detector

Appearance-based loop closure detection may introduce false loop closure constraints when two different places resemble each other, e.g, two posters sharing the same content are placed at different locations. Rejecting such false loop closures is critical for robust reconstruction, as a false loop closure constraint may introduce significant distortion to pose estimation and corrupt the reconstructed model. Aiming to distinguish the difference between true and false loop closure constraints, we have the following observations:

1) The bias status of the IMU can be accurately estimated with the proposed global optimization. The bias variation is small and stable. 2) Loop closure introduces inconsistency in visual observations and provides global constraints for pose estimation. The inconsistency is only caused by the accumulation drift of feature matches, thus introducing the correct loop closure constraint will not influence the bias state. However, a false loop closure will corrupt the energy function and distort the estimated IMU bias state obviously.

Based on these two observations, we propose to detect the false loop by depending on the change of the IMU bias state before and after optimization. When a new keyframe arrives, GN optimization

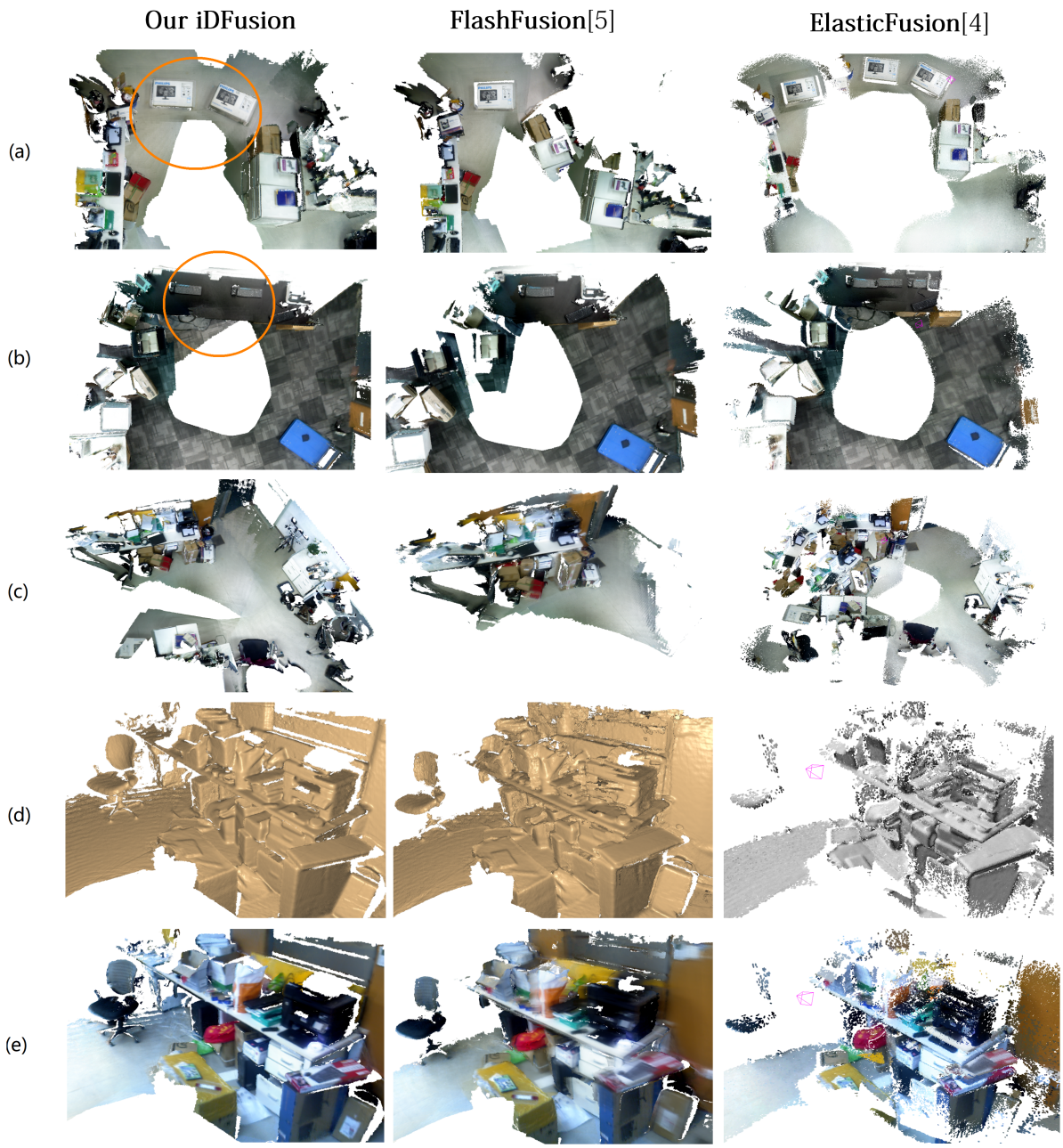


Figure 8: Quantitative evaluation. The concerned methods are the-state-of-art ElasticFusion [4] and FlashFusion [5]. (a), (b) present the scenario with two repetitive appearances (highlighted by circle). iDFusion discards false loop, achieving the global consistency. FlashFusion is misled by the false loop, producing a disordered result. ElasticFusion discards the true loop, causing accumulated error. (c) shows the scenario under fast motion. iDFusion tracks robustly, assuring complete and consistent reconstruction. FlashFusion suffers failed tracking, causing incomplete reconstruction. ElasticFusion applies wrong visual tracking, leading to overlapped reconstruction. (d), (e) show the zoomed local geometry and color details under fast motion. FlashFusion has a more uneven surface and a blurry color appearance. Both the surface and color of ElasticFusion appear more sparse losing many details. On the contrary, iDFusion achieves higher quality reconstruction by fusing visual and inertial observations for higher localization accuracy.

Figure 9: Illustration of localization by IMU. With IMU optimization, (a) has accurate IMU localization so the reconstruction is smooth. Without IMU optimization, (b) has obvious gaps because of the inaccurate localization result.

is applied and we can obtain total acceleration bias variation $\delta\mathbf{b}^{\mathbf{a}}$ and total angular velocity bias variation $\delta\mathbf{b}^{\mathbf{g}}$. T_{b_a} and T_{b_g} is a predefined threshold of acceleration bias variation and angular velocity bias variation, respectively. If the norm of the bias variation satisfies $[\delta\mathbf{b}^{\mathbf{a}}] > T_{b_a}$ and $[\delta\mathbf{b}^{\mathbf{g}}] > T_{b_g}$, we can judge the false loop.

4 EXPERIMENT

We evaluate the performance of iDFusion on the public ICL-NUIM [36] and TUM RGBD [35] datasets. The IMU data is synthetic, which is proposed in RGB-D-Inertial SLAM [23]. Moreover, we adopt ASUS Xtion and a commercial IMU (SC-AHRS-100D2) to capture the challenging real-world data under various scenarios, i.e., the fast motion, textureless environment and repetitive environment. The state-of-art dense 3D reconstruction systems ElasticFusion [4] and FlashFusion [5] are used for comparison. Note that we use an Intel Core i7-9700K @3.6GHz CPU for computing, while the GPU works for merely visualization.

4.1 Quantitative Evaluation

Localization Accuracy. It is evaluated using the RMSE of absolute trajectory error [37]. Table I shows the result in the TUM RGBD and ICL-NUIM datasets, where a smaller value of ATE implies higher accuracy. Other state-of-art reconstruction systems include ElasticFusion [4], BundleFusion [2], CPA-SLAM [29], RGB-D SLAM [30], RGB-D-Inertial SLAM [23], NIO [31], Non-iterative SLAM [32] and FlashFusion [5]. We can see iDFusion has a better performance than the other CPU-based approaches and has comparable performance with the GPU-based construction system. Also, the RMSE of the state-of-the-art monocular visual-inertial approach VINS [19] on TUM fr3/office are around 9.8 cm, which is a lot worse than our RGBD-based approaches.

Complexity Evaluation. After visual initialization, the global optimization thread is running all the time to achieve global consistency. On TUM data fr3/office, the optimization time of a whole loop is presented in Fig. 7. The added IMU and plane terms do not add much time. For 2488 frames, the average time occupation of the proposed global optimization is 32.08 ms. Also the global optimization is in the thread and does not delay the tracking.

4.2 Qualitative Evaluation

Reconstruction under Repetitive Environment. Fig. 6 shows the norm of bias variation. The red arrow shows the bias variation is severe after detecting a false loop. Conversely, the amplitude of bias variation is normal when a true loop happens. So we can decide to accept or discard a loop observation by the norm of bias variation.

Fig. 8(a,b) shows the result of false loop closure detection. There are two similar scenes, and this is prone to be considered as a loop closure. ElasticFusion discard both the false loop and the true loop. So the accumulated error cannot be eliminated. FlashFusion accepts the false loop and the reconstruction is overlapping. They only use visual information to check loop closure, and the result is not robust. Our method uses IMU bias variation to check loop closure, which can detect the false loop and true loop accurately then the global optimization will eliminate the accumulated error effectively.

Reconstruction under Challenging Movement. When moving fast or there is a lack of feature points, the visual tracking is prone to fail while frame pose can be obtained by integrating the IMU data. The result of IMU integration is sensitive to gravity, Cam-IMU transformation matrix, velocity and bias estimation. So the accuracy of IMU integration depends on the accuracy of state optimization. Fig. 9(a) shows the IMU localization result with state optimization. The smoothness shows the accuracy of IMU integration, which proves the accuracy of state optimization. Fig. 9(b) is the IMU localization result without state optimization. The obvious gap shows the importance of global optimization for the IMU status.

Fig. 8(c) shows the reconstruction result under fast motion. FlashFusion will lose tracking and will not continue the reconstruction. ElasticFusion will get an inaccurate pose and the mapping will be overlapping. Our method uses IMU data integration to obtain the camera pose so iDFusion will not lose tracking. The consistency of the reconstruction shows the accuracy of the IMU integration. Fig. 8(d,e) shows the zoomed local geometry and color details under fast motion. iDFusion achieves higher quality reconstruction than FlashFusion and ElasticFusion.

5 CONCLUSIONS

In this paper, we aimed for the solution of dense 3D reconstruction of the environment, which serves as the fundamental component for various applications including virtual reality and mobile robotics. iDFusion was proposed for this target by introducing a real-time solver for the full optimization of visual and inertial observations. Experiments demonstrate that iDFusion achieves state-of-the-art accuracy while is able to reconstruct the environment at sub-centimeter resolution on portable devices. We also calibrate the Cam-IMU transformation matrix online for the plug-and-play usages of inertial sensors, and include a robust loop validity detector to reject false loop closures for robust dense 3D reconstruction.

Currently, the reconstructed 3D model is represented using the raw meshes extracted from the truncated signed distance fields, where the memory usage grows linearly with the area of the reconstructed environment, limiting the reconstructed space to room-scale on portable devices. As for future work, we are going to investigate the online mesh simplification and texture representations for online building scale dense reconstruction.

ACKNOWLEDGEMENT

This work is supported in part by Natural Science Foundation of China (NSFC) under contract No. 61722209 and 6181001011.

REFERENCES

[1] Andrew J. Davison, "Futuremapping: The computational structure of spatial AI systems," *CoRR*, vol. abs/1803.11288, 2018.

[2] 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," *ACM Transactions on Graphics 2017 (TOG)*, 2017.

[3] 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.

[4] Thomas Whelan, Renato F Salas-Moreno, Ben Glocker, Andrew J Davison, and Stefan Leutenegger, "Elasticfusion: Real-time dense slam and light source estimation," *The International Journal of Robotics Research*, vol. 35, no. 14, pp. 1697–1716, 2016.

[5] Lei Han and Lu Fang, "Flashfusion: Real-time globally consistent dense 3d reconstruction using cpu computing," RSS, 2018.

[6] Lei Han, Lan Xu, Dmytro Bobkov, Eckehard Steinbach, and Lu Fang, "Real-time global registration for globally consistent rgb-d slam," *IEEE Transactions on Robotics*, 2019.

[7] Christian Forster, Luca Carlone, Frank Dellaert, and Davide Scaramuzza, "Imu preintegration on manifold for efficient visual-inertial maximum-a-posteriori estimation," Georgia Institute of Technology, 2015.

[8] 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.

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

[10] Brian Curless and Marc Levoy, "A volumetric method for building complex models from range images," 1996.

[11] M. Nießner, M. Zollhöfer, S. Izadi, and M. Stamminger, "Real-time 3d reconstruction at scale using voxel hashing," *ACM Transactions on Graphics (TOG)*, 2013.

[12] Raul Mur-Artal and Juan D Tardós, "Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras," *IEEE Transactions on Robotics*, vol. 33, no. 5, pp. 1255–1262, 2017.

[13] Jonathan Ventura, Clemens Arth, Gerhard Reitmayr, and Dieter Schmalstieg, "Global localization from monocular slam on a mobile phone," *IEEE transactions on visualization and computer graphics*, vol. 20, no. 4, pp. 531–539, 2014.

[14] Sebastian Thrun, Wolfram Burgard, and Dieter Fox, *Probabilistic robotics*, MIT press, 2005.

[15] Anastasios I Mourikis and Stergios I Roumeliotis, "A multi-state constraint kalman filter for vision-aided inertial navigation," in *Proceedings 2007 IEEE International Conference on Robotics and Automation*. IEEE, 2007, pp. 3565–3572.

[16] Mingyang Li, Byung Hyung Kim, and Anastasios I Mourikis, "Real-time motion tracking on a cellphone using inertial sensing and a rolling-shutter camera," in *2013 IEEE International Conference on Robotics and Automation*. IEEE, 2013, pp. 4712–4719.

[17] Joel A. Hesch, Dimitrios G. Kottas, Sean L. Bowman, and Stergios I. Roumeliotis, "Observability-constrained vision-aided inertial navigation," 2012.

[18] Martin Brossard, Silvere Bonnabel, and Axel Barrau, "Invariant kalman filtering for visual inertial slam," *2018 21st International Conference on Information Fusion (FUSION)*, pp. 2021–2028, 2018.

[19] Tong Qin, Peiliang Li, and Shaojie Shen, "Vins-mono: A robust and versatile monocular visual-inertial state estimator," *IEEE Transactions on Robotics*, vol. 34, no. 4, pp. 1004–1020, 2018.

[20] Stefan Leutenegger, Simon Lynen, Michael Bosse, Roland Siegwart, and Paul Furgale, "Keyframe-based visual–inertial odometry using nonlinear optimization," *The International Journal of Robotics Research*, vol. 34, no. 3, pp. 314–334, 2015.

[21] R. Mur-Artal and J. D. TardÃąs, "Visual-inertial monocular slam with map reuse," *IEEE Robotics and Automation Letters*, vol. 2, no. 2, pp. 796–803, April 2017.

[22] Ming Hsiao, Eric Westman, and Michael Kaess, "Dense planar-inertial slam with structural constraints," *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 6521–6528, 2018.

[23] T. Laidlow, M. Bloesch, W. Li, and S. Leutenegger, "Dense rgb-d-inertial slam with map deformations," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Sep. 2017, pp. 6741–6748.

[24] Zunjie Zhu and Feng Xu, "Real-time indoor scene reconstruction with rgbd and inertia input," 12 2018.

[25] O. KÃđhler, V. Adrian Prisacariu, C. Yuheng Ren, X. Sun, P. Torr, and D. Murray, "Very high frame rate volumetric integration of depth images on mobile devices," *IEEE Transactions on Visualization and Computer Graphics*, vol. 21, no. 11, pp. 1241–1250, Nov 2015.

[26] Matthew Klingensmith, Ivan Dryanovski, Siddhartha Srinivasa, and Jizhong Xiao, "Chisel: Real time large scale 3d reconstruction onboard a mobile device using spatially hashed signed distance fields," 07 2015.

[27] C. Feng, Y. Taguchi, and V. R. Kamat, "Fast plane extraction in organized point clouds using agglomerative hierarchical clustering," in *2014 IEEE International Conference on Robotics and Automation (ICRA)*, May 2014, pp. 6218–6225.

[28] Jose Luis Blanco, "A tutorial on se(3) transformation parameterizations and on-manifold optimization," 09 2010.

[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] F. Endres, J. Hess, N. Engelhard, J. Sturm, D. Cremers, and W. Burgard, "An evaluation of the rgb-d slam system," in *2012 IEEE International Conference on Robotics and Automation*, May 2012, pp. 1691–1696.

[31] Chen Wang, Minh-Chung Hoang, L. Xie, and Junsong Yuan, "Non-iterative rgb-d-inertial odometry," 2017.

[32] Chen Wang, Junsong Yuan, and Lihua Xie, "Non-iterative slam," *2017 18th International Conference on Advanced Robotics (ICAR)*, pp. 83–90, 2017.

[33] P. Furgale, J. Rehder, and R. Siegwart, "Unified temporal and spatial calibration for multi-sensor systems," in *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Nov 2013, pp. 1280–1286.

[34] T. Qin, P. Li, and S. Shen, "Vins-mono: A robust and versatile monocular visual-inertial state estimator," *IEEE Transactions on Robotics*, vol. 34, no. 4, pp. 1004–1020, Aug 2018.

[35] J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers, "A benchmark for the evaluation of rgb-d slam systems," in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Oct 2012, pp. 573–580.

[36] A. Handa, T. Whelan, J. McDonald, and A. J. Davison, "A benchmark for rgb-d visual odometry, 3d reconstruction and slam," in *2014 IEEE International Conference on Robotics and Automation (ICRA)*, May 2014, pp. 1524–1531.

[37] 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.