Dense Planar-Inertial SLAM with Structural Constraints

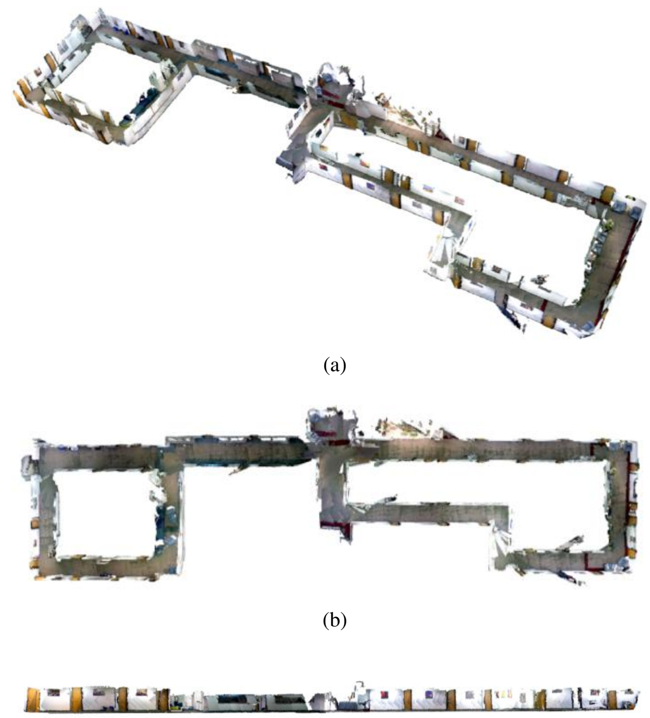
Ming Hsiao, Eric Westman, and Michael Kaess

***Abstract*— In this work, we develop a novel dense planar-inertial SLAM (DPI-SLAM) system to reconstruct dense 3D models of large indoor environments using a hand-held RGB-D sensor and an inertial measurement unit (IMU). The preintegrated IMU measurements are loosely-coupled with the dense visual odometry (VO) estimation and tightly-coupled with the planar measurements in a full SLAM framework. The poses, velocities, and IMU biases are optimized together with the planar landmarks in a global factor graph using incremental smoothing and mapping with the Bayes Tree (iSAM2). With odometry estimation using both RGB-D and IMU data, our system can keep track of the poses of the sensors even without sufficient planes or visual information (e.g. textureless walls) temporarily. Modeling planes and IMU states in the fully probabilistic global optimization reduces the drift that distorts the reconstruction results of other SLAM algorithms. Moreover, structural constraints between nearby planes (e.g. right angles) are added into the DPI-SLAM system, which further recovers the drift and distortion. We test our DPI-SLAM on large indoor datasets and demonstrate its state-of-the-art performance as the first planar-inertial SLAM system.**

I. Introduction

Recent studies have shown that using planes as landmarks in simultaneous localization and mapping (SLAM) systems provides advantages over other existing dense 3D reconstruction methods on accuracy, efficiency, or even both [12, 14, 20, 27]. However, all existing planar SLAM solutions can still drift over time due to the accumulation of small errors along the trajectory estimation, or lose tracking due to insufficient observations of geometric features or photometric textures. Fusing inertial sensors with vision-based SLAM systems is a good solution to reduce all these problems, which results from the complementary nature of these two types of sensors. An inertial measurement unit (IMU) can temporarily track the motion of the sensor even when there are not enough features or texture for visual tracking. On the other hand, consistent visual observations help correct the biases in the inertial measurements. Therefore, the main idea in this work is to fuse inertial measurements with planar observations to achieve better SLAM performance.

There are several existing methods that fuse IMU and camera for various tasks. Filtering methods are frequently used for visual-inertial navigation (VIN) [18, 28], which marginalize out all the previous *IMU states*, including poses, velocities, and biases, to achieve fast computation. However, because the highly nonlinear inertial states are marginalized out at each frame, which cannot be further corrected by later observations or loop closures, filtering methods cannot generate consistent solutions and therefore are not suitable for SLAM tasks. Smoothing or batch optimizing IMU states at each camera frame are the most accurate ways to solve the visual-inertial SLAM problems in theory, but the fast growing number of states and constraints can be very computationally expensive, which prohibits the use of such systems for real-time applications. Sliding window smoothing or local bundle adjustment (BA) solutions find a good balance between filtering and batch optimization methods to solve visual-inertial SLAM problems [17, 19, 22], which preserves the accuracy of batch optimization locally while achieving real-time performance. However, they also marginalize the previous states that exceed the local optimization window, and therefore the inconsistent problem still exists. To achieve both consistency and efficiency, we first apply the on-manifold IMU preintegration method [9] to combine all the IMU measurements between two sparsely selected frames as a single factor. Then we follow a framework similar to that in our previous work [12] to jointly optimize the



(a)

(b)

(c)

Fig. 1: The reconstruction of a large indoor environment using DPI-SLAM system with structural constraints. The drift in the output 3D dense model (a) is significantly reduced, which can also be observed from the top (b) and side (c) views.

The authors are with the Robotics Institute, Carnegie Mellon University, Pittsburgh, PA 15213, USA. {mhsiao, westman, kaess}@cmu.edu

This work was partially supported by National Science Foundation grant IIS-1426703.

preintegrated IMU factors with the planar observations as well as the visual odometry (VO) constraints from the fast dense RGB-D odometry method [12] in a global factor graph, which preserves all the IMU states of the selected frames and can be solved efficiently using iSAM2 [16, 6]. This framework also locally fuses the depth images into a dense depth maps [12], which preserves the detail structures of the environments and allows more accurate plane extraction. With a proper implementation of the online data acquisition and factor graph update (see section VI), this novel dense planar-inertial SLAM (DPI-SLAM) can reconstruct dense 3D models of indoor environments in real-time on a CPU only.

To further extend DPI-SLAM, we also implement a loop closing function that automatically detects loops and merges duplicate planes. Moreover, structural constraints between planes, such as orthogonality and parallelism, are added into the global optimization. They not only reduce the drift and distortion of the output map even before or without loop closing, but also speed up the loop closing process and generate more accurate maps. We demonstrate the improvements of our SLAM system on self-collected large indoor datasets.

The four main contributions of this work are:

1. Developing a novel dense planar-inertial SLAM (DPI-SLAM) system that applies IMU preintegration for consistent global optimization,

2. Implementing DPI-SLAM based on iSAM2 to achieve real-time performance on CPU,

3. Incorporating the structural constraints between landmark planes based on the planar-inertial SLAM framework to further correct the drift and distortion, and

4. Demonstrating the state-of-the-art performance of DPI-SLAM with structural constraints by comparing its reconstruction results with others as well as a ground truth model from a survey lidar.

II. Related Work

Various studies have focused on fusing inertial measurements with sparse feature-based VO methods in recent years. A simple way to fuse them is referred to as *loosely-coupled* [18, 28], which optimizes the 6-DoF output from the VO methods instead of the raw visual measurements together with the inertial measurements. In contrast, *tightly-coupled* methods [17, 19, 22] jointly optimize the raw sparse feature point constraints together with the inertial measurements in each iteration of the nonlinear optimization, which is more costly but can achieve more accurate results.

However, fusing IMU with direct dense VO methods, such as iterative closest point (ICP) [4] or dense RGB-D odometry [26], are not discussed much in the existing literature. In theory, smoothing or batch optimizing all the raw constraints from direct dense methods and inertial measurements in a tightly-coupled way can achieve the best possible estimates, but it is too expensive for global optimization and therefore cannot run in real-time. An alternative solution introduced in visual-inertial direct SLAM [5] tightly-couples the inertial measurements and the direct semi-dense VO constraints for each pose estimation only, and marginalizes the IMU and

Fig. 2: The IMU (Microstrain 3DM-GX4-25) is rigidly attached on the top of the RGB-D sensor (ASUS Xtion Pro Live).

VO measurements into pose-to-pose constraints in a pose graph. Even though this method can estimate the relative motion between frames more accurately, it does not allow the later observations to update the IMU biases for global optimization. A different approach in [21] loosely couples the constraints from a dense stereo tracking method with the inertial measurements within a sliding window, which also cannot correct the IMU biases globally. IMU preintegration [9] offers efficient solution to either loosely or tightly couple the IMU measurements with visual constraints, which is applied in this work to solve the planar-inertial SLAM problem. In our DPI-SLAM, the preintegrated IMU factors are loosely coupled with the dense VO constraints while tightly coupled with the planar observations. Even though the dense VO constraints and the inertial measurements are not tightly coupled, our approach allows global correction of the IMU biases in a much cheaper way, and therefore can achieve consistent and accurate solutions online in real-time.

Structural constraints, such as orthogonal or parallel planes, are expected to further correct the drift or distortion in the reconstructions of man-made environments. [23] and [24] demonstrate the advantages of applying these structural constraints on 2D and 3D mapping respectively. A similar but more limited concept called *Manhattan world assumption* is also applied in [8, 25] to achieve better mapping results. However, none of the recent studies of planar SLAM using hand-held RGB-D sensors [12, 14, 20, 27] considers applying any of these ideas to improve the outputs. In this work we will also exploit the structural constraints of the planes to further improve reconstruction results, especially when there is no loop closure constraint to correct the accumulated error in the system.

III. System Structure

Our DPI-SLAM system consists of three main parts (see Fig. 3). The first part (a) includes the odometry estimation and a frame labeling process. The second part (b) performs local depth fusion. The third part (c) combines global planar-inertial mapping with structural constraints and loop closing.

In the first part (a), the pose of each RGB-D frame is predicted using the preintegrated IMU measurements, and further estimated using our fast dense RGB-D odometry method. Both pose prediction and estimation are relative to the most recent *reference frame* R_j. Only the poses of the specially selected *keyframes*, reference frames, and *fusion frames* are estimated using the full precise fast dense RGB-D odometry method (solid black lines in Fig. 3-a). For all other frames, we only estimate their poses roughly using a simplified method (dotted black lines in Fig. 3-a) for efficiency. At each reference frame, the current preintegrated

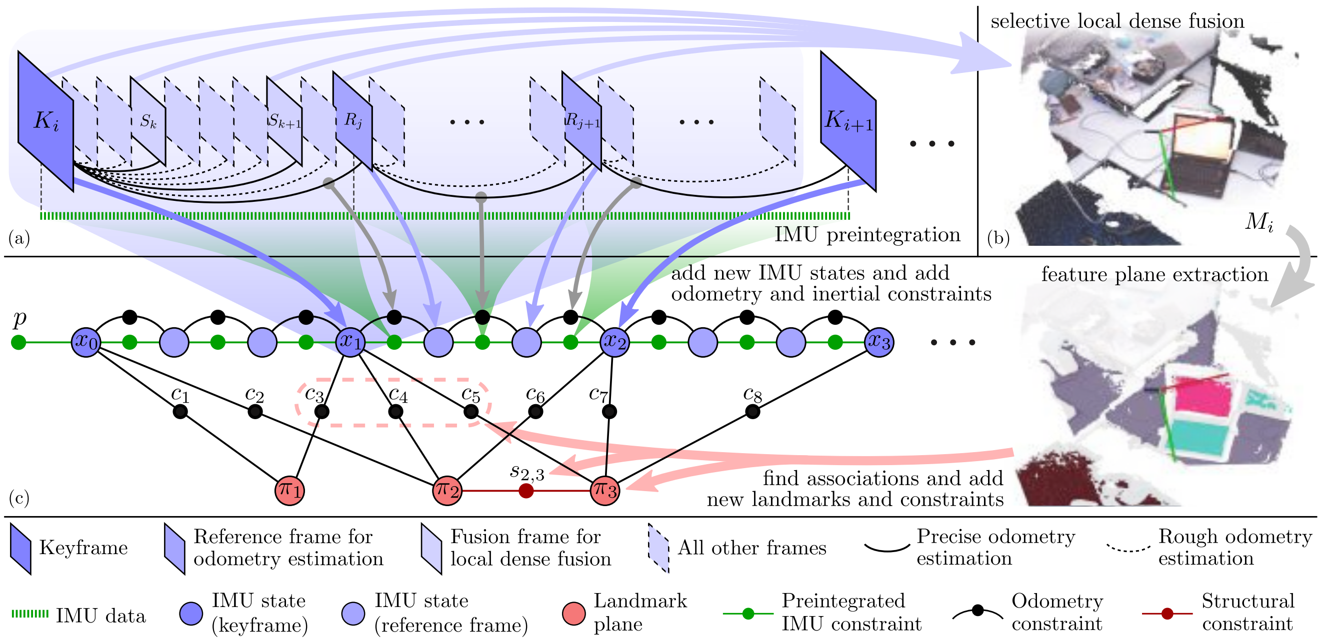


Fig. 3: The system structure of DPI-SLAM, which is similar to KDP-SLAM [12] but modified to allow fusing IMU measurements: (a) IMU preintegration, fast dense RGB-D odometry algorithm, and frame labeling process. (b) Selective local depth fusion algorithm. (c) Optimization of IMU states and planar landmarks in the global factor graph $\mathcal{G}$ with structural constraints and loop closing. Note that for the set of all keyframes $\mathcal{K}$, all reference frames $\mathcal{R}$, and all fusion frames $\mathcal{U}$, $\mathcal{K} \subset \mathcal{R} \subset \mathcal{U}$ holds. Also, the loop closing constraints are not shown here but in Fig. 5 for readability.

IMU factor and VO factor estimated by the fast dense RGB-D odometry method are added into the global factor graph $\mathcal{G}$ (the arrows from Fig. 3-a to Fig. 3-c) for joint optimization. If any new frame's pose is too far away from that of R_j, it will be set as the next reference frame R_{j+1}. Also, a new frame is selected as a new keyframe K_{i+1} if its pose is too far away from that of the current keyframe K_i.

In the second part (b), we fuse the depth of the selected fusion frames from K_i to the last frame before K_{i+1} into a local depth map L_i. Since the poses of the reference frames are further corrected by the inertial measurements, the estimated poses of the fusion frames as well as the resulting local depth maps L_i are also more accurate.

In the third part (c), we extract planes and their corresponding point clusters from L_i at each keyframe, associate them with existing landmark planes using a projective method, add the planar factors into the global factor graph $\mathcal{G}$, and jointly optimize them with the existing VO and inertial factors in $\mathcal{G}$. After that, $\mathcal{G}$ is updated again with the newly found structural constraints. Finally, loop closure constraints are detected and optimized in $\mathcal{G}$. The global factor graph $\mathcal{G}$ is shown in Fig. 3-c, where the IMU states at both keyframes and reference frames as well as the states of the landmark planes are represented as variable nodes and linked with each other by factors. The two types of factors between IMU states encode the preintegrated IMU constraints (green) and odometry constraints (black) from the fast dense RGB-D odometry method respectively. The factors between the IMU states of the keyframes and landmark planes encode plane observations $c_1, ..., c_q$. Additional factors $s_{p,p'}$ (dark red) are added between landmark planes as structural constraints. The system applies iSAM2 [16] to update $\mathcal{G}$ incrementally whenever an IMU state is added into the factor graph with its corresponding factors. When a loop is detected, the entire graph $\mathcal{G}$ is updated for several iterations until convergence.

For more details about the fast dense RGB-D odometry, local depth fusion, and projective data association of planes, please refer to our previous work KDP-SLAM [12].

IV. Planar-Inertial SLAM

A. IMU Preintegration in the Global Factor Graph

Each IMU state contains the pose, velocity, and bias terms for both gyroscope and accelerometer, which can be represented as a 15-vector

$$\mathbf{x}_t = \begin{bmatrix} \boldsymbol{\xi}_t^\top & \mathbf{v}_t^\top & \mathbf{b}_\omega^\top & \mathbf{b}_a^\top \end{bmatrix}^\top, \tag{1}$$

at time t, where $\boldsymbol{\xi}_t$ represents the 6-DoF pose of the IMU, which can also be represented as a rotation matrix $\mathtt{R}_t$ and a translation vector $\mathbf{p}_t$, and $\mathbf{v}_t$ is the 3-DoF velocity. $\mathbf{b}_\omega$ and $\mathbf{b}_a$ are the 3-DoF bias terms for the gyroscope and accelerometer respectively, which are assumed to be static over each preintegration interval (and can change between intervals). Assuming that the raw angular velocity $\boldsymbol{\omega}_t$ and acceleration $\mathbf{a}_t$ arrives every Δt seconds, we can define the *preintegrated rotation*, *velocity*, and *translation* between the two consecutive reference frames at time t and $t' = t + m\Delta t$ respectively as

$$\Delta\mathtt{R}_{t'}^{t} = \prod_{k=t}^{t'} \operatorname{Exp}\left(\left(\boldsymbol{\omega}_k - \mathbf{b}_\omega\right)\Delta t\right), \tag{2}$$

$$\Delta \mathbf{v}_{t'}^{t} = \sum_{k=t}^{t'} \Delta \mathtt{R}_{k}^{t} \left(\mathbf{a}_k - \mathbf{b}_a\right) \Delta t, \tag{3}$$

$$\Delta \mathbf{p}_{t'}^{t} = \sum_{k=t}^{t'} \left[\Delta \mathbf{v}_{k}^{t} \Delta t + \frac{1}{2} \Delta \mathtt{R}_{k}^{t} \left(\mathbf{a}_k - \mathbf{b}_a\right) \Delta t^2 \right], \tag{4}$$

and the error functions of their corresponding factors in $\mathcal{G}$ are

$$e_{\Delta \mathtt{R}_{t'}^{t}} = \mathrm{Log}\left\{ \left[\Delta \mathtt{R}_{t'}^{t} \mathrm{Exp}\left(J_{\Delta \mathtt{R}_{t'}^{t}} \begin{bmatrix} \delta \mathbf{b}_\omega \\ \delta \mathbf{b}_a \end{bmatrix} \right) \right]^{\top} \mathtt{R}_t^{\top} \mathtt{R}_{t'} \right\}, \tag{5}$$

$$e_{\Delta \mathbf{v}_{t'}^{t}} = \mathtt{R}_t^{\top} \left(\mathbf{v}_{t'} - \mathbf{v}_t - \mathbf{g} m \Delta t \right) - \Delta \mathbf{v}_{t'}^{t} - J_{\Delta \mathbf{v}_{t'}^{t}} \begin{bmatrix} \delta \mathbf{b}_\omega \\ \delta \mathbf{b}_a \end{bmatrix}, \tag{6}$$

$$\begin{aligned} e_{\Delta \mathbf{p}_{t'}^{t}} = {} & \mathtt{R}_t^{\top} \left(\mathbf{p}_{t'} - \mathbf{p}_t - \mathbf{v}_t m \Delta t_m - \frac{1}{2} \mathbf{g} m^2 \Delta t^2 \right) \\ & - \Delta \mathbf{p}_{t'}^{t} - J_{\Delta \mathbf{p}_{t'}^{t}} \begin{bmatrix} \delta \mathbf{b}_\omega \\ \delta \mathbf{b}_a \end{bmatrix}, \end{aligned} \tag{7}$$

where $J_{\Delta \mathtt{R}_{t'}^{t}}$, $J_{\Delta \mathbf{v}_{t'}^{t}}$, and $J_{\Delta \mathbf{p}_{t'}^{t}}$ are the Jacobians of $\Delta \mathtt{R}_{t'}^{t}$, $\Delta \mathbf{v}_{t'}^{t}$, and $\Delta \mathbf{p}_{t'}^{t}$ with respect to the bias vector $\left[\mathbf{b}_\omega^\top \ \mathbf{b}_a^\top\right]^\top$ respectively, which allows updating the bias terms linearly. The $\mathrm{Exp}(.)$ and $\mathrm{Log}(.)$ functions are the exponential and log maps that transform the rotation in 3D between $SO(3)$ and its minimal representation in $\mathbb{R}^3$. $\mathbf{g}$ is the constant gravity vector, which should be defined in advance.

Because of the two assumptions that (1) the bias is constant within each preintegration interval and (2) the bias states only require linear updates in each update step, the IMU preintegration interval should not be too long. Therefore, we choose the interval between each two consecutive *reference frames* instead of two keyframes for IMU preintegration, which helps improve the odometry estimate. We apply the implementation of IMU preintegration in GTSAM [2] library in our system, which calculates the preintegrated IMU measurements as well as the Jacobians incrementally. For more details of IMU preintegration, please refer to [9].

B. Fast Dense RGB-D Odometry with IMU Preintegration

The same fast dense RGB-D odometry method proposed in KDP-SLAM [12] is applied in this work with few modifications to calculate the VO constraints between frames. The original fast dense RGB-D odometry method initializes the pose of each new input RGB-D frame as the pose of its previous frame based on a small motion assumption. However, because of incorporating the IMU sensor in this work, we can predict the relative pose of each frame to the most recent reference frame using the preintegrated rotation $\Delta \mathtt{R}_{t+n\Delta t}^{t}$ and translation $\Delta \mathbf{p}_{t+n\Delta t}^{t}$ over the n number of IMU measurements between the two frames as

$$\tilde{\mathtt{R}}_{t+n\Delta t} = \mathtt{R}_t \Delta \mathtt{R}_{t+n\Delta t}^{t}, \tag{8}$$

$$\tilde{\mathbf{p}}_{t+n\Delta t} = \mathbf{p}_t + \mathtt{R}_t \Delta \mathbf{p}_{t+n\Delta t}^{t} + \mathbf{v}_t n \Delta t - \frac{1}{2} \mathbf{g} n \Delta t^2. \tag{9}$$

Using the preintegrated IMU measurements to predict the initial pose of each frame results in better odometry estimation, especially in the cases of fast rotation and lack of texture.

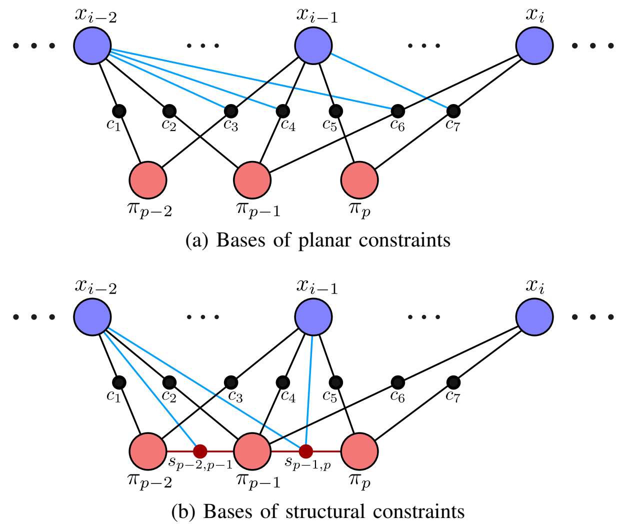


Fig. 4: Both planar and structural constraints have links (blue lines) to their bases. (a) All the planar factors that link to the same landmark plane are also linked to the same base, which is the IMU state of the keyframe that first observes that landmark plane. (b) Each structural constraint is linked to two bases, which are the IMU states of the keyframes that first observe the two landmark planes that the structural constraint is linking to.

Jointly optimizing RGB-D odometry with IMU measurements at every reference frame further results in more robust and accurate estimation of IMU states in the global optimization, which also allows the correction of the biases of the IMU measurements. The inertial constraints between each two consecutive reference frames is defined as in Eq. 5-7, and the RGB-D odometry constraint is defined as a 6-DoF pose-to-pose factor. Notice that since the poses of the IMU states represent the poses of the IMU instead of RGB-D camera, each original RGB-D odometry estimation ${}^{\mathrm{c}}\mathtt{T}_{\mathrm{rgbd}}$ has to be transformed into the IMU coordinates before taken as a odometry factor between IMU states. The transformation

$$\mathtt{T}_{\mathrm{rgbd}} = \mathtt{T}_{\mathrm{c}} \cdot {}^{\mathrm{c}}\mathtt{T}_{\mathrm{rgbd}} \cdot \mathtt{T}_{\mathrm{c}}^{-1} \tag{10}$$

is based on the relative camera pose $\mathtt{T}_{\mathrm{c}}$ in the IMU coordinates, which can be calibrated offline in advance.

C. Global Planar Mapping with IMU Preintegration

Planes are segmented and extracted at each keyframe from the locally fused depth map using [11] and [7], associated with each other using a projective method proposed in [12], and tightly-coupled with the IMU measurements in the global optimization. Each plane π is modeled as a unit length homogeneous vector $\boldsymbol{\pi} = \left[\mathbf{n}^\top \ d\right]^\top \in \mathbb{P}^3$, $\|\boldsymbol{\pi}\| = 1$ in the projective space on the unit sphere S^3, where $\mathbf{n}$ is its normal vector and d the distance to the origin. The overparametrized plane model has the same minimal representation $\boldsymbol{\omega} \in \mathbb{R}^3$ as quaternions, which is used to update each plane in the optimization through the exponential map [14]:

$$\exp\left(\boldsymbol{\omega}\right) = \begin{pmatrix} \frac{1}{2}\mathrm{sinc}\left(\frac{1}{2}\|\boldsymbol{\omega}\|\right)\boldsymbol{\omega} \\ \cos\left(\frac{1}{2}\|\boldsymbol{\omega}\|\right) \end{pmatrix} \in S^3. \tag{11}$$

The plane observations will be added as factors between the corresponding landmark planes and IMU states of keyframes, which tightly-couples planar and inertial measurements in the global optimization. As a result, the biases of the IMU measurements can be corrected again by the planar constraints at every keyframe, which allows even better initialization for the next preintegration interval and refines the estimations of previous IMU states. Similar to the RGB-D odometry constraints in Eq. 10, since each plane measurement $\boldsymbol{\pi}_{\mathrm{c}}$ is originally observed in the camera coordinates, it should be transform into the IMU coordinates as $\boldsymbol{\pi} = \mathtt{T}_{\mathrm{c}}^{-\top}\boldsymbol{\pi}_{\mathrm{c}}$ before being added into the optimization.

A relative formulation is applied for each landmark plane π_p by setting its base as the IMU state x_i that corresponds to the keyframe K_i that first observes π_p. Every plane observation factor c_q that links an IMU state node to π_p will be additionally linked to x_i (the corresponding IMU state of K_i) through a ternary factor (see Fig. 4-a). This allows faster convergence especially in loop closures since the planes anchored to a pose will be automatically moved along with the pose when there is a global update.

V. Structural Constraints

Orthogonality and parallelism are the two most common structural constraints found between two planar surfaces in indoor environments, which can be added into our planar-inertial SLAM system to further correct the drift in rotation.

In our system, two landmark planes π_a and π_b that are observed within a short interval are the candidate pairs for structural constraints. For each pair π_a and π_b, we first compute $h_{ab} = \left|(\mathtt{R}_a\mathbf{n}_a)^\top \mathtt{R}_b\mathbf{n}_b\right|$, which is the absolute value of the dot product of their normal vectors in the global coordinates, where $\mathtt{R}_a$, $\mathtt{R}_b$ are the rotation matrices of their base poses respectively. Then, the two planes are regarded as parallel if h_{ab} is greater than a parallel threshold $h_\parallel$, orthogonal if h_{ab} is less than an orthogonal threshold $h_\perp$, or no specific relationship if none of the above.

Each structural constraint factor is linked to not only the two corresponding landmark planes but also their bases as a quaternary factor (see Fig. 4-b) because of the relative formulation (see Sec. IV-C), which results in faster convergence in the optimization as well. For any pair of π_a and π_b, the error function of their orthogonal constraint factor is

$$e_\perp = \frac{1}{\sigma_\perp^2}\left[(\mathtt{R}_a\mathbf{n}_a)^\top \mathtt{R}_b\mathbf{n}_b\right]^2, \tag{12}$$

and the error function of the parallel constraint factor is

$$e_\parallel = \frac{1}{\sigma_\parallel^2}\left\|[\mathtt{R}_a\mathbf{n}_a]_\times \mathtt{R}_b\mathbf{n}_b\right\|^2. \tag{13}$$

The variances $\sigma_\perp$ and $\sigma_\parallel$ for the orthogonal and parallel factors are set to be small (3×10^{-5}) for strong constraints.

VI. Implementation

Proper implementation is required to make the DPI-SLAM system efficient. The first key idea is to implement the three main parts described in section III in three concurrent threads respectively, which speeds up the system a lot. More implementation details are discussed as follows.

A. Online Data Synchronization

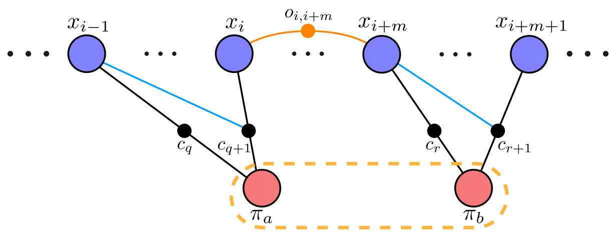
For our online real-time SLAM system, it is crucial to synchronize the input data from the RGB-D and IMU sensors. Our implementation assumes that IMU measurements are available at much higher frequency than RGB-D images, and both of them are measured with timestamps. Even though the temporal offset between RGB-D images and IMU measurements can be calibrated in advance, the image data can still arrive later than the IMU data due to the delay of transmission and preprocessing. So if we preintegrate every input IMU data immediately after it arrives, the "future IMU data" that exceeds the desired preintegration interval might be accidentally added into the current preintegrated IMU factor. A naive solution to avoid this problem is to store all of the input IMU measurements and only preintegrate those within the interval at once at each reference frame. However, in our system, there can be tens of thousands of IMU data to be preintegrated at each reference frame (depends on the sensor motion that affects reference frame selection), which might occasionally slow down the system. As a result, we use a buffer to temporarily store the IMU data stream. Upon the arrival of each new RGB-D frame, only those with earlier timestamps than the timestamp of the new frame are taken out from the buffer and preintegrated. This distributes the work load to each frame and makes the processing time more stable.

B. Gravity Initialization

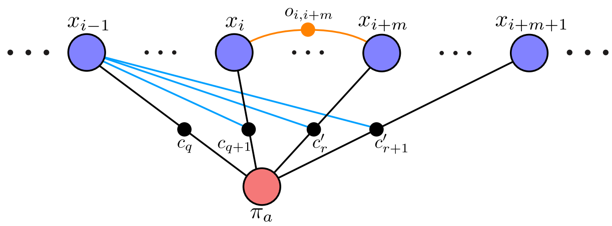
A constant gravity is assumed in the IMU preintegration framework and has to be set in advance. If the IMU and camera are static in the first few frames, and we assume that the initial accelerometer biases are much smaller than the gravity, we can take the average direction of the first few acceleration measurements to be the initial gravity direction $\mathbf{g}_{\mathrm{init}}$. Then, we set the orientation of the prior p (see Fig. 3-c) in a way such that $\mathbf{g}_{\mathrm{init}}$ is rotated and aligned with the z-direction of the global coordinates. Finally, we set the gravity magnitude to $9.81\mathrm{m/s}^2$.

C. Updating Factor Graph with Parallel Subgraphs

As described in Sec. III, the global factor graph $\mathcal{G}$ can be updated by either the RGB-D odometry and preintegrated IMU factors from the first thread with a higher frequency of every reference frame, or the planar or structural constraints from the third thread (see Fig. 3) with a lower frequency of every keyframe. To allow conflict-free and efficient updates from both threads, the new states and factors in the two threads are added into two different *subgraphs* respectively in parallel. Whenever one of the subgraphs is ready, the system will add it into $\mathcal{G}$ and update the entire $\mathcal{G}$ while using mutex lock to avoid adding the other subgraph or accessing the same piece of memory from the other thread in the mean time. This implementation is also based on GTSAM [2].



(a) Before merging two landmark planes



(b) After two landmark planes are merged

Fig. 5: Loop closing with landmark planes merging in the global factor graph $\mathcal{G}$. (a) A pose-to-pose loop closing constraint is added, and two landmark planes π_a and π_b are detected to be representing the same plane. (b) The two landmark planes are merged, and the new factors c'_r and c'_{r+1} are added into $\mathcal{G}$ to replace c_r and c_{r+1}. When all of the factors of the landmark plane π_b are removed, π_b will be automatically removed from $\mathcal{G}$ in the applied GTSAM implementation of iSAM2.

D. Loop Closing

Even though incorporating IMU measurements helps to reduce drift, and the additional structural constraints further reduce the drift in rotation, some small amount of drift still plagues the optimized trajectory and map. As a result, we apply a bag-of-words approach [10] to detect loops and the following algorithm based on iSAM2 [16] to close the loops.

For every keyframe K_j that is detected to be a loop closure candidate with a previous keyframe K_i, we apply the RANSAC-based perspective-*n*-point (P*n*P) algorithm in OpenCV [1] on the SURF [3] feature points extracted from K_i and K_j to estimate the relative transformation first, then apply our fast dense RGB-D odometry method to refine it. The refined output is added into $\mathcal{G}$ as a constraint between the poses of K_i and K_j. Optimizing $\mathcal{G}$ with the keyframe-to-keyframe loop closing constraint usually requires updating a larger part of the underlying Bayes tree in iSAM2, which takes several iterations to converge.

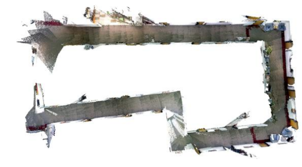
Finally, we check the similarity of landmark planes that are observed in these two keyframes using the same association method, and merge those that are actually representing the same plane to further constrain the solution and avoid the duplication of landmarks. The merging is implemented by relinking the factors of each new landmark plane to the corresponding old one while also updating their base poses (see Fig. 5).

VII. Experimental Results

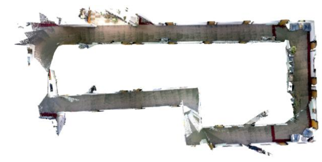
A. Experimental Settings

We evaluate DPI-SLAM on a desktop computer with an Intel Core i7-4790 processor, and GPU being used only for visualization, not computation. There are five separate threads in the system, including the three main threads introduced in Sec. III, an IMU and camera data input thread, and a visualization thread. Our implementation, which has not been optimized, runs at 30 fps.

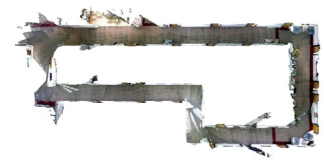
(a)



(b)



(c)



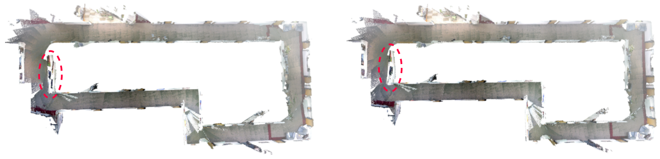
(d)

Fig. 6: The real-time dense 3D reconstructions (from top view) *without loop closure* in four different settings: (a) planar SLAM without using IMU data, (b) planar-inertial SLAM with IMU preintegrated over each two keyframes, (c) DPI-SLAM: planar-inertial SLAM with IMU preintegrated over each two reference frames, and (d) DPI-SLAM with structural constraints.

We use an ASUS Xtion Pro Live attached with a Microstrain 3DM-GX4-25 (see Fig. 2) to collect indoor RGB-D datasets with inertial measurements for evaluation. Both color and depth images from Xtion have 640×480 resolution at 30 fps, and both the raw rotational velocity and acceleration measurements from Microstrain are provided at 1000 Hz. We adopt LCM [13] to transmit the RGB-D and inertial data to our system online, or log and replay them to simulate the real-time process. We compare the 3D reconstructions generated by different settings of our system for qualitative evaluation. We also provide a quantitative evaluation by comparing the output model from DPI-SLAM with a ground truth model. The dense 3D ground truth model is obtained with a FARO Focus3D survey lidar scanner from a sequence of stationary 360 degree scans.

B. Results and Discussion

The reconstruction results from the various settings of the system are shown in Figs. 6 and 7. Even though pure planar SLAM without inertial fusion (which is actually KDP-SLAM [12] with its iSAM [15] part replaced by iSAM2) can reduce drift along each corridor, the drift at each corner and along the long corridor are still visible in the output map (see Fig. 6-a). Naively adding preintegrated IMU factors between keyframes does affect the result in some way (see Fig. 6-b), but its drift is still quite large in both rotation (e.g. the upper-right corner) and translation (e.g. the upper long corridor) because the IMU biases are not corrected frequently enough, and also the assumption of constant bias for IMU preintegration might not hold within longer preintegration intervals. The proposed method of preintgerating IMU measurements over each two consecutive reference frames results in a much better reconstruction (see Fig. 6-c), where the drift in rotation at the corners and translation along the corridor are both reduced significantly. Adding structural constraints between



(a) (b)

Fig. 7: The dense 3D reconstructions *with loop closure* (circled in red) in two different system settings. (a) DPI-SLAM without structural constrains (corresponds to Fig. 6-c), which takes about 30 iterations to close the loop. (b) DPI-SLAM with structural constrains (corresponds to Fig. 6-d), which takes only 10 iterations to close the loop.

landmark planes further corrects the drift (see Fig. 6-d) and allows better loop closing results, which is also true when the loop is detected and closed (see Fig. 7). Finally, we can use DPI-SLAM with structural constraints to reconstruct the 3D dense model of the entire floor from the full dataset (see Fig. 1). More reconstruction results of different indoor environments are shown in Figs. 9 and 10.

After registering our output model in Fig. 7-b with the survey lidar model using ICP, we calculate the point-to-plane root-mean-square error (RMSE) and mean absolute error (MAE) between the two models, which are $0.069\mathrm{m}$ and $0.049\mathrm{m}$ respectively. Given that the entire model is about $30\mathrm{m} \times 13\mathrm{m} \times 3\mathrm{m}$, the average error ratio is less than 0.7%. Since we cannot record the RGB-D and IMU data sequence at the same time when the lidar scans are collected, there can be uncontrollable changes in the public environment (see Fig. 8). Therefore, the actual errors between the two methods should be even smaller.

From the experiments, we can conclude that adding inertial measurements improves planar SLAM, and further combining structural constraints can achieve the best reconstruction results. In addition, even though some of the datasets are collected in the same environments shown in our previous paper [12], the sensors are allowed to move about 3 times faster in this work because the odometry estimations from visual-inertial fusion are more robust than pure VO.

Comparing to our previous work KDP-SLAM [12], DPI-SLAM has to estimate more states in the global factor graph and calculate additional IMU preintegration terms and structural constraints. However, these two SLAM systems actually have similar speed because IMU provides better initial pose estimation for the fast dense RGB-D odometry method to converge in fewer iterations. Also, using iSAM2 to update and optimize the global factor graph is more efficient than using iSAM, which requires periodic batch optimization steps. As a result, DPI-SLAM can generally achieve the same real-time performance as KDP-SLAM.

As for closing a loop until convergence, DPI-SLAM sometimes cannot run in real-time because having more IMU states in the global factor graph requires more iterations to converge (e.g. the loop closure process in Fig. 10). Fortunately, with structural constraints, the drift in the trajectory can be much smaller, and therefore the loop closure process can be faster (see Fig. 7). Also, if a converged result is not required immediately, DPI-SLAM can distribute the iteration steps to the later update steps of the global factor graph at each reference frame so that the system will not slow down during loop closing (e.g. the right loop in Fig. 1 gradually converges as the mapping process of the left part continues).

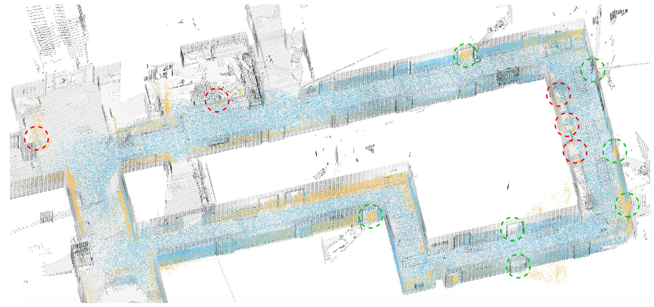
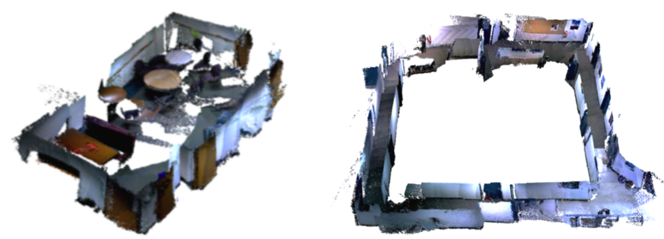


Fig. 8: The registration of our output model (colored) onto the survey lidar ground truth model (black), both downsampled approximately 100 times to save calculation. The points with small deviation from ground truth are shown in blue, while larger RMSE is indicated by yellow. Notice that between recording these two datasets, some doors (green circles) and chairs (red circles) had been moved.



(a) (b)

Fig. 9: Dense 3D reconstruction of various indoor environments using our DPI-SLAM system. (a) An open space with three round tables. (b) Corridors in a loop with lighting changes.

Lastly, adding structural constraints into the pure planar SLAM system might also improve its results. However, without IMU measurements, the drift in rotation can be too large for the system to decide if there should be structural constraints or not. In this case, if a relaxed threshold is chosen, wrong structural constraints might be added into the system and cause more errors.

VIII. Conclusion

We present a novel dense planar-inertial SLAM (DPI-SLAM) approach with structural constraints to reconstruct dense 3D models of indoor environments in real-time using CPU only. The preintegrated IMU measurements improve the fast dense RGB-D tracking as well as the global planar mapping, and the structural constraints further reduce the drift and distortion in the output maps. We demonstrate the advantages of DPI-SLAM through real-world experiments, and its efficiency as a CPU-based dense visual-inertial SLAM system with real-time performance.

In the near future, we would like to publicly share multiple RGB-D and IMU data sequences together with the survey lidar models. As for further improving our SLAM system, keyframe reusing strategies can be adopted to allow long-term localization and mapping in the same environments. We

(a) (b)

Fig. 10: The reconstruction result of a large two-floor dataset with a loop using DPI-SLAM, where the dense model of the corridors and the stairs are clearly shown in (a). The top view (b) shows that the two long corridors on the different floors are well aligned with each other.

also want to extend our current system to SLAM in dynamic environments.

IX. Acknowledgments

We would like to thank Chuck Whittaker for his technical support for the ground truth data collection.

References

[1] Open source computer vision library | OpenCV. [Online]. Available: https://github.com/opencv/opencv

[2] Georgia Tech smoothing and mapping library | GTSAM. [Online]. Available: https://bitbucket.org/gtborg/gtsam

[3] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool, "Speeded-up robust features (SURF)," *Computer vision and image understanding*, vol. 110, no. 3, pp. 346–359, 2008.

[4] P. J. Besl and H. D. McKay, "A method for registration of 3-D shapes," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 14, no. 2, pp. 239–256, Feb 1992.

[5] A. Concha, G. Loianno, V. Kumar, and J. Civera, "Visual-inertial direct SLAM," in *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, May 2016, pp. 1331–1338.

[6] F. Dellaert and M. Kaess, "Factor graphs for robot perception," *Foundations and Trends in Robotics*, vol. 6, no. 1-2, pp. 1–139, Aug. 2017.

[7] C. Erdogan, M. Paluri, and F. Dellaert, "Planar segmentation of RGBD images using fast linear fitting and Markov chain Monte Carlo," in *Proceedings of the 2012 Ninth Conference on Computer and Robot Vision*, ser. CRV '12, Washington, DC, USA, 2012, pp. 32–39.

[8] A. Flint, C. Mei, I. Reid, and D. Murray, "Growing semantically meaningful models for visual SLAM," in *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, June 2010, pp. 467–474.

[9] C. Forster, L. Carlone, F. Dellaert, and D. Scaramuzza, "On-manifold preintegration for real-time visual–inertial odometry," *IEEE Trans. Robotics*, vol. 33, no. 1, pp. 1–21, Feb 2017.

[10] D. Galvez-Lopez and J. D. Tardos, "Bags of binary words for fast place recognition in image sequences," *IEEE Trans. Robotics*, October 2012.

[11] D. Holz, S. Holzer, R. Rusu, and S. Behnke, "Real-time plane segmentation using RGB-D cameras," in *RoboCup 2011: Robot Soccer World Cup XV*, ser. Lecture Notes in Computer Science, vol. 7416. Springer, 2012, pp. 306–317.

[12] M. Hsiao, E. Westman, G. Zhang, and M. Kaess, "Keyframe-based dense planar SLAM," in *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, Singapore, May 2016.

[13] A. S. Huang, E. Olson, and D. C. Moore, "LCM: Lightweight communications and marshalling," in *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, Oct 2010, pp. 4057–4062.

[14] M. Kaess, "Simultaneous localization and mapping with infinite planes," in *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, May 2015, pp. 4605–4611.

[15] M. Kaess, A. Ranganathan, and F. Dellaert, "iSAM: Incremental smoothing and mapping," *IEEE Trans. Robotics*, vol. 24, no. 6, pp. 1365–1378, Dec. 2008.

[16] M. Kaess, H. Johannsson, R. Roberts, V. Ila, J. J. Leonard, and F. Dellaert, "iSAM2: Incremental smoothing and mapping using the Bayes tree," *Intl. J. of Robotics Research*, vol. 31, no. 2, pp. 216–235, 2012.

[17] N. Keivan, A. Patron-Perez, and G. Sibley, "Asynchronous adaptive conditioning for visual-inertial SLAM," in *Experimental Robotics*. Springer, 2016, pp. 309–321.

[18] K. Konolige, M. Agrawal, and J. Sola, "Large-scale visual odometry for rough terrain," in *Robotics research*. Springer, 2010, pp. 201–212.

[19] S. Leutenegger, S. Lynen, M. Bosse, R. Siegwart, and P. Furgale, "Keyframe-based visual–inertial odometry using nonlinear optimization," *Intl. J. of Robotics Research*, vol. 34, no. 3, pp. 314–334, 2015.

[20] L. Ma, C. Kerl, J. Stueckler, and D. Cremers, "CPA-SLAM: Consistent plane-model alignment for direct RGB-D SLAM," in *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, May 2016.

[21] L. Ma, J. M. Falquez, S. McGuire, and G. Sibley, "Large scale dense visual inertial SLAM," in *Field and Service Robotics*. Springer, 2016, pp. 141–155.

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

[23] V. Nguyen, A. Harati, A. Martinelli, R. Siegwart, and N. Tomatis, "Orthogonal SLAM: a step toward lightweight indoor autonomous navigation," in *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Oct 2006, pp. 5007–5012.

[24] V. Nguyen, A. Harati, and R. Siegwart, "A lightweight SLAM algorithm using orthogonal planes for indoor mobile robotics," in *2007 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Oct 2007, pp. 658–663.

[25] B. Peasley, S. Birchfield, A. Cunningham, and F. Dellaert, "Accurate on-line 3D occupancy grids using manhattan world constraints," in *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, Oct 2012, pp. 5283–5290.

[26] F. Steinbrucker, J. Sturm, and D. Cremers, "Real-time visual odometry from dense RGB-D images," in *Computer Vision Workshops (ICCV Workshops), 2011 IEEE International Conference on*, Nov 2011, pp. 719–722.

[27] Y. Taguchi, Y.-D. Jian, S. Ramalingam, and C. Feng, "Point-plane SLAM for hand-held 3D sensors," in *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, May 2013, pp. 5182–5189.

[28] S. Weiss, M. W. Achtelik, S. Lynen, M. Chli, and R. Siegwart, "Real-time onboard visual-inertial state estimation and self-calibration of MAVs in unknown environments," in *IEEE Intl. Conf. on Robotics and Automation (ICRA)*. IEEE, 2012, pp. 957–964.