Tightly Coupled 3D Lidar Inertial Odometry and Mapping

Preprint · April 2019

3 authors:

Haoyang Ye
The Hong Kong University of Science and Technology
22 PUBLICATIONS   100 CITATIONS

SEE PROFILE

Yuying Chen
The Hong Kong University of Science and Technology
19 PUBLICATIONS   117 CITATIONS

SEE PROFILE

Ming Liu
The Hong Kong University of Science and Technology
326 PUBLICATIONS   3,629 CITATIONS

SEE PROFILE

Some of the authors of this publication are also working on these related projects:

Project heterogeneous sensor calibration View project

Project Autonomous Satellite Orbit Determination View project

Tightly Coupled 3D Lidar Inertial Odometry and Mapping

Haoyang Ye[1], Yuying Chen[1] and Ming Liu[1]

Abstract—**Ego-motion estimation is a fundamental require-ment for most mobile robotic applications. By sensor fusion, we can compensate the deficiencies of stand-alone sensors and pro-vide more reliable estimations. We introduce a tightly coupled lidar-IMU fusion method in this paper. By jointly minimizing the cost derived from lidar and IMU measurements, the lidar-IMU odometry (LIO) can perform well with acceptable drift after long-term experiment, even in challenging cases where the lidar measurements can be degraded. Besides, to obtain more reliable estimations of the lidar poses, a rotation-constrained refinement algorithm (LIO-mapping) is proposed to further align the lidar poses with the global map. The experiment results demonstrate that the proposed method can estimate the poses of the sensor pair at the IMU update rate with high precision, even under fast motion conditions or with insufficient features.**

I. INTRODUCTION

Ego-motion estimation plays a major role in many navi-gation tasks and is one of the key problems for autonomous robots. It offers the knowledge of robot poses and can provide instant feedback to the pose controllers. Besides, together with the various sensors perceiving the environment, it provides crucial information for simultaneous localization and mapping (SLAM). Accurate estimation of the robot pose helps to reduce risks and contributes to successful planning.

The lidar sensor that can provide distance measurements for surrounding environments has been widely used in robotic systems. To be specific, a typical 3D lidar can sense the surroundings at a frequency around 10Hz with a horizontal field of view (FOV) of 360 degrees. Besides, as an active sensor, it is invariant to the illumination. The high reliability and precision make the lidar sensor a popular option for pose estimation.

Despite its advantages, the lidar sensor is not perfect and with several shortcomings. From the lidar itself, it has a low vertical resolution and the sparse point cloud it obtains provides limited features, thus making feature tracking an intractable problem. In addition, lidars mounted on moving robots suffer from motion distortion, which directly affects sensing accuracy. In real-world scenarios, there are some lidar-degraded cases in which the lidar receives few or missing points. For example, 3D lidar receives only few usable points in narrow corridor environments. The received points are mainly from the side walls and only a small portion of points are observed from the ceiling and floor. In this case, the matched lidar features can easily lead to ill-constrained pose estimation. Another shortcoming is its low

update rate. This limits its application for tasks that require fast response such as the control of a robot pose.

In this paper, we present a tightly coupled 3D lidar-IMU pose estimation algorithm to overcome the aforementioned problems. Measurements from both the lidar and IMU are used for a joint-optimization. To achieve real-time and more consistent estimation, fixed-lag smoothing and marginal-ization of old poses are applied, followed by a rotation-constrained refinement. The main contributions of our work are as follows:

- A tightly coupled lidar-IMU odometry algorithm is proposed. It provides real-time accurate state estimation with high update rate.
- Given the prior from the lidar-IMU odometry, a rota-tional constrained refinement method further optimizes the final poses and the generated point-cloud maps. It ensures a consistent and robust estimation, even in some lidar-degraded cases.
- The algorithm is verified with extensive indoor and outdoor tests. It outperforms the state-of-the-art lidar-only or loosely coupled lidar-IMU algorithms.
- The source code is available online [1]. This is the first open-source implementation for tightly coupled lidar and IMU fusion available to the community.

The remainder of the paper is organized as follows. Sec. II presents a review of related works. Sec. III explains the notation and some preliminaries. The proposed odometry and refinement methods are presented in Sec. IV and V, respectively. Implemetation and tests are shown in Sec. VI and VII. Conclusion is presented in Sec. VIII.

II. RELATED WORKS

There are several methods relating to the fusion of IMU and lidar measurements. One important category is loosely coupled fusion. Methods in this category consider the estimation of the lidar and the estimation of the IMU separately. In [1], the lidar odometry with IMU assistance relied on the orientation calculated by the IMU and assumed a zero velocity when using the acceleration. It decoupled the measurements of the lidar and IMU and mainly took the IMU as prior for the whole system, thus it could not utilize IMU measurements for further optimization. In [2], a loosely coupled extended Kalman filter (EKF) was used to fuse an IMU and lidar in the 2D case, but it could not handle 3D or more complex environments. Lynen et al. [3] presented a modular way to fuse IMU measurements with

[1]Haoyang Ye, Yuying Chen and Ming Liu are with RAM-LAB `https://ram-lab.com/`, in the Department of Electronic and Computer Engi-neering, Hong Kong University of Science and Technology, Kowloon, Hong Kong `hy.ye@connect.ust.hk, ychenco,eelium@ust.hk`

[1]`https://sites.google.com/view/lio-mapping` or `https://ram-lab.com/file/hyye/lio-mapping`

other relative pose measurements, e.g., from a camera, lidar or even pressure sensor, by an EKF in the 3D case. This loosely coupled method is of computational efficiency, but is less accurate than tightly coupled methods [4] since it takes the odometry part as a black box and does not update it with measurements from the IMU.

Tightly coupled methods are another important category. For 2D planar motion estimation, Soloviev et al. [5] proposed a method that extracted and matched lines among the 2D lidar scans, where the tilted lidar was compensated by the predicted orientation from the IMU. A Kalman filter was applied to correct the IMU states in the lidar measurement domain. Hemann et al. [6] proposed a method to tightly couple the IMU propagation and accumulated lidar heightmap in the form of an error-state Kalman filter. The corrections of the states are updated with the matching between the lidar heightmap and a priori digital elevation model (DEM). This method showed the ability in long-range GPS-denied navigation when the environments are known, but it could not work without prior map information. In [7] and [8], the raw measurements directly from the IMU and the predicted IMU measurements from the continuous trajectory were used for calculating the residuals to be optimized. The transition and estimation of the states were not involved in these methods, which made the systems unfeasible under fast motions even with an additional camera [9].

Inspired by other visual-inertial works [10], [11], we design our method under tightly coupled lidar-IMU fusion. We "pre-integrate" and use the raw IMU measurements with the lidar measurements to optimize the states within the whole system, which can work in laser degraded cases or when the motions are rapid. To the best of our knowledge, ours is one of the few 3D lidar-IMU fusion algorithms that are suited for complex 3D environments.

III. NOTATIONS AND PRELIMINARIES

A. Notations

We denote every line of measurement captured by the 3D lidar sensor as *scan* $\mathcal{C}$, and denote *sweep* $\mathcal{S}$ containing all the scans in one measurement. For example, a 16-line 3D lidar contains 16 scans in one sweep.

In the following sections, we denote the transformation matrix as $\mathbf{T}_b^a \in SE(3)$, which transforms the point $\mathbf{x}^b \in \mathbb{R}^3$ in the frame $\mathcal{F}_b$ into the frame $\mathcal{F}_a$. $\bar{\mathbf{T}}_b^a$ is the transform predicted by IMU. $\mathbf{R}_b^a \in SO(3)$ and $\mathbf{p}_b^a \in \mathbb{R}^3$ are the rotation matrix and the translation vector of $\mathbf{T}_b^a$, respectively. The quaternion $\mathbf{q}_b^a$ under Hamilton notation is used, which corresponds to $\mathbf{R}_b^a$. $\otimes$ is used for the multiplication of two quaternions. We use $\hat{\mathbf{a}}_k$ and $\hat{\boldsymbol{\omega}}_k$ to denote the raw measurements of the IMU at timestamp k. The extracted features are denoted as $\mathbf{F}_a$ in the original capture frame $\mathcal{F}_a$, which can be transformed into the frame $\mathcal{F}_b$ as $\mathbf{F}_a^b$.

B. IMU Dynamics

1) States: The body frame $\mathcal{F}_{B_i}$ and $\mathcal{F}_{L_i}$ are the reference of the IMU body and the reference of the lidar center, respectively, while obtaining the lidar sweep $\mathcal{S}_i$ at the discrete timestamp i. The states we will estimate are the IMU state $\mathbf{X}_{B_i}^W$ in the world frame $\mathcal{F}_W$ and the extrinsic parameters $\mathbf{T}_B^L$ between the lidar and the IMU sensors. In detail, we can write the IMU states at i and the extrinsic parameters as

$$
\begin{aligned}
\mathbf{X}_{B_i}^W &= \begin{bmatrix} \mathbf{p}_{B_i}^{W\,T} & \mathbf{v}_{B_i}^{W\,T} & \mathbf{q}_{B_i}^{W\,T} & \mathbf{b}_{a_i}^{\,T} & \mathbf{b}_{g_i}^{\,T} \end{bmatrix}^T \\
\mathbf{T}_B^L &= \begin{bmatrix} \mathbf{p}_B^{L\,T} & \mathbf{q}_B^{L\,T} \end{bmatrix}^T
\end{aligned}, \quad (1)
$$

where $\mathbf{p}_{B_i}^W$, $\mathbf{v}_{B_i}^W$ and $\mathbf{q}_{B_i}^W$ are the position, velocity and orientation of the body frame w.r.t. to the world frame, respectively, $\mathbf{b}_a$ is the IMU acceleration bias and $\mathbf{b}_g$ is the IMU gyroscope bias.

2) Dynamic model: With the inputs from the IMU's accelerator and gyroscope, we can update the preceding IMU state $\mathbf{X}_{B_i}^W$ to current IMU state $\mathbf{X}_{B_j}^W$ by discrete evolution, as shown in Eq. (2), where Δt is the interval between two consecutive IMU measurements, and all the IMU measurements between the lidar sweeps' times $k = i$ and $k = j$ are integrated. With slight abuse of notation, we use $k = j-1$ as the preceding IMU timestamp before $k = j$.

$$
\begin{aligned}
\mathbf{p}_j &= \mathbf{p}_i + \sum_{k=i}^{j-1} \left[\mathbf{v}_k \Delta t + \frac{1}{2}\mathbf{g}^W \Delta t^2 + \frac{1}{2}\boldsymbol{R}_k(\hat{\mathbf{a}}_k - \mathbf{b}_{a_k})\Delta t^2 \right] \\
\mathbf{v}_j &= \mathbf{v}_i + \mathbf{g}^W \Delta t_{ij} + \sum_{k=i}^{j-1} \mathbf{R}_k(\hat{\mathbf{a}}_k - \mathbf{b}_{a_k})\Delta t \\
\mathbf{q}_j &= \mathbf{q}_i \otimes \prod_{k=i}^{j-1} \delta\mathbf{q}_k = \mathbf{q}_i \otimes \prod_{k=i}^{j-1} \begin{bmatrix} \frac{1}{2}\Delta t(\hat{\boldsymbol{\omega}}_k - \mathbf{b}_{g_k}) \\ 1 \end{bmatrix},
\end{aligned}
\tag{2}
$$

where $\mathbf{g}^W$ is the gravity vector in the world frame. We use the shorthands $(\cdot)_i \doteq (\cdot)_{B_i}^W$, $\Delta t_{ij} = \sum_{k=i}^{j-1}\Delta t$ and $\prod_{k=i}^{j-1}$ as the sequences of quaternion multiplications for clarity.

3) Pre-integration: The body motion between the timestamps i and j can be represented via pre-integration measurement $z_j^i = \{\Delta\mathbf{p}_{ij}, \Delta\mathbf{v}_{ij}, \Delta\mathbf{q}_{ij}\}$, which has the covariance $\mathbf{C}_{B_j}^{B_i}$ in the error-state model (see details in the supplementary material [12]).

IV. TIGHTLY COUPLED LIDAR-IMU ODOMETRY

To ensure efficient estimation, many works on lidar mapping, like [7], [1] and [13], separate the task into two parts, the odometry and the mapping. Inspired by these works, the proposed system comprises two parallel parts. The first part, introduced in Sec. IV, is the tightly coupled lidar-IMU odometry, which optimizes all the states within a local window. The second part, presented in Sec. V, is the rotation constrained refinement (leading to a globally consistent mapping process), which aligns the lidar sweeps to the global map using the information from the optimized poses and gravity constraints.

A. Lidar-IMU Odometry Overview

Fig. 1 provides a brief overview of our proposed lidar-IMU odometry. With the previous estimated states, we can use the
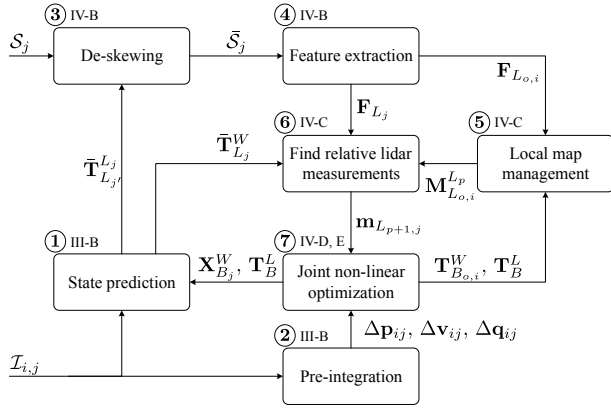
Fig. 1: Lidar-IMU odometry framework. The subsection that discussed each block is given after the circular number.

current lidar raw input $\mathcal{S}_j$, and the IMU raw inputs $\mathcal{I}_{i,j}$, from the last timestamp i to the current timestamp j, to have a new step of optimization for the states. The odometry estimation performs as follows: 1) Before S_j arrives, the IMU states are updated via Eq. (2) iteratively. 2) Meanwhile, these inputs are "pre-integrated" as $\Delta\mathbf{p}_{ij}$, $\Delta\mathbf{v}_{ij}$ and $\Delta\mathbf{q}_{ij}$ to be used in the joint optimization. 3) When the latest lidar sweep $\mathcal{S}_j$ is received, de-skewing is applied on the raw data to obtain the de-skewed lidar sweep $\bar{\mathcal{S}}_j$ (Sec. IV-B). 4) Next, a feature extraction step is applied to reduce the dimension of the data and extract the most important feature points $\mathbf{F}_{L_j}$ (Sec. IV-B). 5) The previous lidar feature points $\mathbf{F}_{L_{o,i}}$ within the local window are merged as a local map $\mathbf{M}_{L_{o,i}}^{L_p}$, according to the previous corresponding optimized states $\mathbf{T}_{B_{o,i}}^W$ and $\mathbf{T}_B^L$ (Sec. IV-C). 6) With the predicted lidar pose for $\mathbf{F}_j$, we can find the relative lidar measurements $\mathbf{m}_{L_{p+1,j}}$ (Sec. IV-C). 7) The final step is joint non-linear optimization, taking the relative lidar measurements and IMU pre-integration to obtain a MAP estimation of the states within the local window (Sec. IV-D and IV-E). The optimized results are applied to update the prediction states in step 1), avoiding the drift from IMU propagation.

B. De-skewing and Feature Extraction

The 3D lidar has rotating mechanism inside to receive data for a whole circle. When the 3D lidar is moving, $\mathcal{S}_j$, the raw data from it, suffers from motion distortion, which makes the point in a sweep different from the true positions. To handle this problem, we use the prediction of lidar motion $\bar{\mathbf{T}}_{L_{j'}}^{L_j}$ from IMU propagation and assume the linear motion model during the sweep. Then, every point $\mathbf{x}(t) \in \mathcal{S}_j \subset \mathbb{R}^3$ is corrected by linear interpolation of $\bar{\mathbf{T}}_{L_{j'}}^{L_j}$ to obtain the deskewed sweep $\bar{\mathcal{S}}_j$ into the ending pose of the sweep, where $t \in (t_{j'}, t_j]$ is the timestamp of the point in the sweep, and $t_{j'}$ and t_j are the timstamps of the sweep start and end, respectively.

For computational efficiency, lidar feature extraction is required. Here, we are only interested in the points which are most alike on a plane or on an edge [1], [7], since these points can be extracted from every scan of a lidar sweep. Such feature points $\mathbf{F}_{L_j}$ in $\bar{\mathcal{S}}_j$ are selected by the curvature
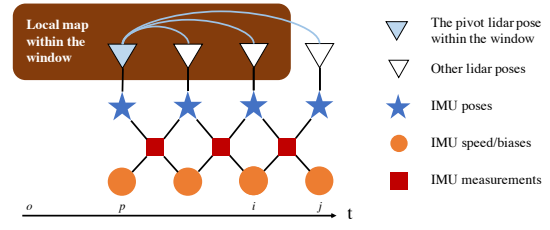

Fig. 2: Local window. The local map consists of the previous point-clouds before j and starting from o. The optimization window contains the frame between p and j.

and distance changes, as are those in [1]; i.e., the most planar or edged points are selected.

C. Relative Lidar Measurements

With the fusion of the IMU and another sensor, which is capable to provide the relative pose of the sensor pair, the states to be estimated, $\mathbf{X}_B^W$ and $\mathbf{T}_B^L$, will be locally observable if we fix the first reference frame [14]. To properly incorporate pre-integration from the IMU, we propose using the relative lidar measurements, between sweeps to constrain the lidar poses, as Algorithm 1. Before finding the point correspondences, we build a local map, because the points in a single sweep are not dense enough to calculate accurate correspondences.

The local map contains the lidar feature points from N_m discrete timestamps $\{o, \cdots, p, \cdots, i\}$, where o, p and i are the timestamps of the *first* lidar sweep within the window, the *pivot* lidar sweep and the *last* processed lidar sweep, respectively, as shown in Fig. 2. The local map $\mathbf{M}_{L_{o,i}}^{L_p}$ is built in the frame of the *pivot* lidar sweep from the features $\mathbf{F}_{L_\gamma}^{L_p}, \gamma \in \{o, \cdots, i\}$, which is transformed via the previous optimized lidar poses $\mathbf{T}_{L_\gamma}^{L_p}$ [2]. The to-be-estimated states are the ones at the N_s timestamps $\{p+1, \cdots, i, j\}$, where $p+1$ and j are the timestamp of the lidar sweep next to the *pivot* one and the current lidar sweep in the window.

Algorithm 1: Relative Lidar Measurements

Input: $\mathbf{F}_{L_\gamma}$ and $\mathbf{T}_{L_\gamma}^W$, $\gamma \in \{o, \cdots, i, j\}$
Output: $\mathbf{m}_{L_\alpha}$, $\alpha \in \{p+1, \cdots, i, j\}$

1 **for** $\gamma \leftarrow o$ **to** j **do**
2   Transform $\mathbf{F}_{L_\gamma}$ into $\mathcal{F}_{L_p}$ as $\mathbf{F}_{L_\gamma}^{L_p}$ by $\mathbf{T}_{L_\gamma}^{L_p}$;
3   **if** $\gamma \neq j$ **then**
4   Merge $\mathbf{F}_{L_\gamma}^{L_p}$ into $\mathbf{M}_{L_{o,i}}^{L_p}$;
5   **end**
6 **end**
7 **for** $\alpha \leftarrow p+1$ **to** j **do**
8   Find KNN($\mathbf{F}_{L_\alpha}^{L_p}$) in $\mathbf{M}_{L_{o,i}}^{L_p}$;
9   For each point $\mathbf{x} \in \mathbf{F}_{L_\alpha}$, use $\pi(\mathbf{x}^{L_p})$ to obtain the relative measurement model, which forms $\mathbf{m}_{L_\alpha}$;
10 **end**

---

[2] For simplification, we denote the predicted transform $\bar{\mathbf{T}}_{L_j}^{L_p}$ as $\mathbf{T}_{L_j}^{L_p}$ in this section, and $\mathbf{F}_{L_j}^{L_p}$ is transformed via $\bar{\mathbf{T}}_{L_j}^{L_p}$.

With the built local map, the correspondences can be found between $\mathbf{M}_{L_{o,i}}^{L_p}$ and the original $\mathbf{F}_{L_\alpha}, \alpha \in \{p+1, \cdots, j\}$. We define such correspondence as relative lidar measurements, since they are relative to the *pivot* pose, and the *pivot* pose will change with the sliding window. The original features we extracted in Sec. IV-B are the most planar or edged points in $\mathcal{F}_{L_\alpha}$. In practice, we found that the edged points cannot improve the results of the lidar-IMU odometry. Thus, in the following, we only discuss the planar features. KNN is used for each transformed feature point $\mathbf{x}^{L_p} \in \mathbf{F}_{L_\alpha}^{L_p}$ to find the $\mathbf{K}$ nearest points $\pi(\mathbf{x}^{L_p})$ in $\mathbf{M}_{L_{o,i}}^{L_p}$. Then for the planar points, we fit these neighbor points into a plane in $\mathcal{F}_{L_p}$. The coefficient of a planar point can be solved by the linear equation defined by $\boldsymbol{\omega}^T \mathbf{x}' + d = 0, \mathbf{x}' \in \pi(\mathbf{x}^{L_p})$, where $\boldsymbol{\omega}$ is the plane normal direction and d is the distance to the origin of $\mathcal{F}_{L_p}$. We denote $m = [\mathbf{x}, \boldsymbol{\omega}, d] \in \mathbf{m}_{L_\alpha}$ for each planar feature point $\mathbf{x} \in \mathbf{F}_{L_\alpha}$ as one of the relative lidar measurements. To be mentioned, in each relative lidar measurement $m \in \mathbf{m}_{L_\alpha}$, $\mathbf{x}$ is defined in $\mathcal{F}_{L_\alpha}$, and $\boldsymbol{\omega}$ and d are defined in $\mathcal{F}_{L_p}$.

D. Lidar Sweep Matching

The relative lidar measurements can provide relative constraints between the *pivot* lidar pose and the following lidar poses. Our method optimizes all the poses in the optimization window, including the first pose $\mathbf{T}_{L_p}^W$, i.e., $\mathcal{F}_{L_p}$ is not fixed. Thus, each item in the lidar cost function involves the poses of two lidar sweeps, $\mathbf{T}_{L_p}^W$ and $\mathbf{T}_{L_\alpha}^W, \alpha \in \{p+1, \cdots, j\}$. Optimizing the *pivot* pose will help to minimize the pre-integration error better and ensure the sensor pair align with gravity. The states we estimate are those of the IMU; thus we need to introduce extrinsic parameters to represent the lidar constraints by IMU states. The relative transformation from the latter lidar poses to the *pivot* one in the window can be defined as

$$\mathbf{T}_{L_\alpha}^{L_p} = \mathbf{T}_B^L \mathbf{T}_{B_p}^{W}{}^{-1} \mathbf{T}_{B_\alpha}^W \mathbf{T}_B^{L}{}^{-1} = \begin{bmatrix} \mathbf{R}_{L_\alpha}^{L_p} & \mathbf{p}_{L_\alpha}^{L_p} \\ \mathbf{0} & 1 \end{bmatrix}. \quad (3)$$

With the previous correspondences, the residual for each relative lidar measurement $m = [\mathbf{x}, \boldsymbol{\omega}, d] \in \mathbf{m}_{L_\alpha}, \alpha \in \{p+1, \cdots, j\}$ can be represented as point-to-plane distance

$$\mathbf{r}_\mathcal{L}(m, \mathbf{T}_{L_p}^W, \mathbf{T}_{L_\alpha}^W, \mathbf{T}_B^L) = \boldsymbol{\omega}^T(\mathbf{R}_{L_\alpha}^{L_p} x + \mathbf{p}_{L_\alpha}^{L_p}) + d. \quad (4)$$

E. Optimization

To obtain the optimized states, a fixed-lag smoother and marginalization are applied. The fixed-lag smoother keeps N_s IMU states in the sliding window, from $\mathbf{X}_{B_p}^W$ to $\mathbf{X}_{B_j}^W$, as shown in Fig. 2. The sliding window helps to bound the amount of computation. When new measurement constraints come, the smoother will include the new states and marginalize the oldest states in the window. The whole of the states to be estimated in detail is

$$\mathbf{X} = \left[\mathbf{X}_{B_p}^W, \cdots, \mathbf{X}_{B_j}^W, \mathbf{T}_B^L \right]. \quad (5)$$
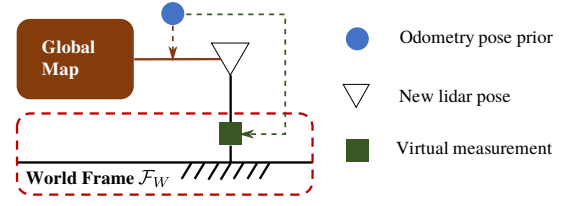

Fig. 3: Rotationally constrained mapping. The odometry pose first serves as a prior for the global point-cloud registration. Then the rotational component from the odometry is applied as a virtual measurement to constrain the optimization.

Then the following cost funtion with a Mahalanobis norm is minimized to obtain the MAP estimation of the states $\mathbf{X}$,

$$\min_{\mathbf{X}} \frac{1}{2} \left\{ \|\mathbf{r}_\mathcal{P}(\mathbf{X})\|^2 + \sum_{\substack{m \in \mathbf{m}_{L_\alpha} \\ \alpha \in \{p+1, \cdots, j\}}} \|\mathbf{r}_\mathcal{L}(m, \mathbf{X})\|_{\mathbf{C}_{L_\alpha}^m}^2 \\ + \sum_{\beta \in \{p, \cdots, j-1\}} \left\| \mathbf{r}_\mathcal{B}(z_{\beta+1}^\beta, \mathbf{X}) \right\|_{\mathbf{C}_{B_{\beta+1}}^{B_\beta}}^2 \right\}, \quad (6)$$

where $\mathbf{r}_\mathcal{P}(\mathbf{X})$ is the prior items from marginalization. $\mathbf{r}_\mathcal{L}(m, \mathbf{X})$ is the residual of the relative lidar constraints and $\mathbf{r}_\mathcal{B}(z_{\beta+1}^\beta, \mathbf{X})$ is the residual of the IMU constraints. The cost function in the form of a non-linear least square can be solved by the Gauss-Newton algorithm, which takes the form $\mathbf{H}\delta\mathbf{X} = -\mathbf{b}$. We use Ceres Solver [15] to solve the problem.

The lidar constraint $\mathbf{r}_\mathcal{L}(m, \mathbf{X})$ can be derived from Eq. (4) for each relative lidar measurement. The covariance matrix $\mathbf{C}_{L_\alpha}^m$ is determined by the lidar accuracy. Similar to the one in [10], IMU constraint $\mathbf{r}_\mathcal{B}(z_{\beta+1}^\beta, \mathbf{X})$ can be obtained from the states and IMU pre-integration, $\|\mathbf{r}_\mathcal{P}(\mathbf{X})\|^2 = \mathbf{b}_\mathcal{P}^T \mathbf{H}_\mathcal{P}^+ \mathbf{b}_\mathcal{P}$ can be obtained by Schur complement (detailed in [12]).

V. REFINEMENT WITH ROTATIONAL CONSTRAINTS

Registering the feature points to a global map, instead of local maps, can constrain the lidar poses to a consistent world frame $\mathcal{F}_W$. Our refinement method uses the relative lidar measurements $\mathbf{m}_L$ as the ones in Sec. IV. Since the global map is a by-product of the refinement, we also refer to it as a mapping method. A cost function to align the latest lidar feature points with the global map can be formed as

$$\mathbf{C}_\mathcal{M} = \sum_{m \in \mathbf{m}_L} \left\| \mathbf{r}_\mathcal{M}(m, \mathbf{T}_L^W) \right\|^2, \\ \mathbf{r}_\mathcal{M}(m, \mathbf{T}) = \boldsymbol{\omega}^T(\mathbf{R}\mathbf{x} + \mathbf{p}) + d \quad (7)$$

where $\mathbf{T} = \mathbf{T}_L^W$ is the latest to-be-estimated lidar pose, and m is the relative lidar measurement with the feature point $\mathbf{x}$ in $\mathcal{F}_L$ and the coefficients $\boldsymbol{\omega}, d$ defined in $\mathcal{F}_W$. Then we can use a similar Gauss-Newton method to minimize $\mathbf{C}_\mathcal{M}$. The optimization is carried out by the residual $\mathbf{C}_\mathcal{M}$ and Jacobians $\mathbf{J}_\mathbf{p}^\mathbf{C}$ and $\mathbf{J}_\theta^\mathbf{C}$, where θ is the error state of the corresponding quaternion $\mathbf{q}$. However, with the accumulated rotation error and after long-term operation, the merged global map cannot align with gravity accurately. This can lead further mapping to wrongly align with a tilted map. Inspired by [16] which

optimizes SE(3) with SE(2)-constraints, we propose a constrained mapping strategy. This strategy utilizes the rotational constraints from the lidar-IMU odometry, which ensures the final map always aligns with gravity. Fig. 3 illustrates the structure of the rotationally constrained mapping.

Given the property that the orientation along the z-axis has higher uncertainty, and that the other two DoF of the orientation are much more close to the true value, we can constrain the cost function by modifying the Jacobian of the orientation as (detailed derivations in [12]),

$$\mathbf{J}^{\mathbf{C}}_{\boldsymbol{\theta}_z} = \mathbf{J}^{\mathbf{C}}_{\boldsymbol{\theta}} \cdot (\breve{\mathbf{R}})^T \cdot \breve{\boldsymbol{\Omega}}_z$$

$$\breve{\boldsymbol{\Omega}}_z = \begin{bmatrix} \epsilon_x & 0 & 0 \\ 0 & \epsilon_y & 0 \\ 0 & 0 & 1 \end{bmatrix}, \tag{8}$$

where $(\breve{\cdot})$ denotes the estimation of the state in the last iteration, and $\breve{\boldsymbol{\Omega}}_z$ is an approximiation of the information matrix of the orientation w.r.t $\mathcal{F}_W$, and ϵ_x and ϵ_y can be obtained by the information ratio of the x- and y-axes orientation to the z-axis orientation in $\mathcal{F}_W$.

After that, we use $\mathbf{J}^{\mathbf{C}}_{\mathbf{p}}$ and $\mathbf{J}^{\mathbf{C}}_{\boldsymbol{\theta}_z}$ as the Jacobians instead, and these are needed for the optimization step. The incremental lidar poses can be obtained as $\delta\boldsymbol{\theta}_z$ and $\delta\mathbf{p}$, which lead to the updated lidar states $\tilde{\mathbf{p}}$ and $\tilde{\mathbf{q}}$

$$\tilde{\mathbf{p}} = \breve{\mathbf{p}} + \delta\mathbf{p}$$
$$\tilde{\mathbf{q}} = \begin{bmatrix} \frac{1}{2}\delta\boldsymbol{\theta}_z \\ 1 \end{bmatrix} \otimes \breve{\mathbf{q}}. \tag{9}$$

VI. Implementation

Different sensor configurations, system initialization and different parameters for indoor and outdoor tests are introduced in this section.

A. Different Sensor Configurations

Sensor pairs can be configured differently. For a hand-held sensor pair, such as the one in Fig. 4a, the lidar and IMU are close to each other. Thus the pipeline remains the same as what was introduced before. But for sensor pairs mounted on cars, the two sensors are usually farther away from each other. For example, Fig. 4b shows an IMU mounted above the car's base link, while the lidar is mounted at the front of the car. Instead of auto-calibrating all translation parameters, a prior item for the extrinsic translational parameters is added to Eq. (6) for the tests on the cars.

B. Initialization

At first, there are no estimated poses for the lidar feature points. Therefore, roughly accurate matching algorithms are needed. We adopt [1]'s lidar odometry in the initialization step. With the provided lidar poses, sufficient motions of the sensor pair are required to make the IMU states observable [14]. Then the poses of the lidar and the IMU measurements are used to initialize the IMU states, which can be solved by the methods introduced in [10] and [17]. For the tests in Sec. VII, we follow the initialization method in [10], which also linearly initializes the extrinsic parameters. Next, with
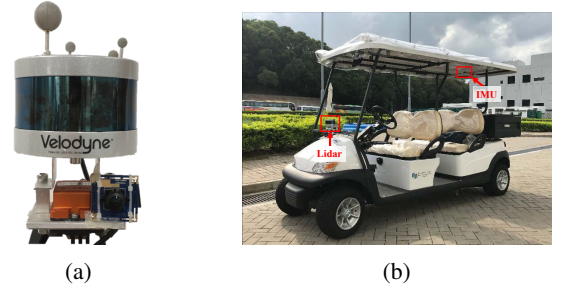

(a)   (b)

Fig. 4: (a) The sensor configuration for quantitative analysis and indoor tests. Note that the attached camera is only used to record the test scenes. (b) Outdoor golf car configuration.

TABLE I: Translational and rotational errors w.r.t. ground-truth. The motions in the 6 sequences vary from fast to slow.

Errors	Sequence	LOAM	LIO-raw	LIO-no-ex	LIO	LIO-mapping
Translation RMSE (m)	fast 1	0.4469	0.2464	0.0957	**0.0949**	**0.0529**
	fast 2	0.2023	0.4346	0.1210	**0.0755**	**0.0663**
	med 1	0.1740	0.1413	0.1677	**0.1002**	**0.0576**
	med 2	**0.1010**	0.2460	0.3032	0.1308	**0.0874**
	slow 1	**0.0606**	0.1014	0.0838	0.0725	**0.0318**
	slow 2	**0.0666**	0.1016	0.0868	0.1024	**0.0435**
Rotation RMSE (rad)	fast 1	0.1104	0.1123	0.0547	**0.0545**	**0.0537**
	fast 2	0.0763	0.1063	0.0784	**0.0581**	**0.0574**
	med 1	0.0724	0.0620	0.0596	**0.0570**	**0.0523**
	med 2	0.0617	0.0886	0.0900	**0.0557**	**0.0567**
	slow 1	**0.0558**	0.0672	0.0572	0.0581	**0.0496**
	slow 2	0.0614	0.0548	0.0551	**0.0533**	**0.0530**

the initial states and the newly coming measurements, the non-linear optimization within local windows will be carried out iteratively to estimate the states.

VII. Tests and Analyses

Indoor and outdoor tests are conducted to evaluate our method. The quantitative and qualitative results are provided in the following sections.

A. Quantitative Analysis

To quantitatively analyze our method, the sensor pair shown in Fig. 4a is used. A Velodyne VLP-16 lidar with 16 lines is mounted above an Xsens MTi-100 IMU. The reflective markers can provide the ground-truth poses using the motion capture system. The lidar is configured to have a 10Hz update rate, and IMU updates at 400Hz. The estimated trajectories from different methods are aligned with the ground-truth using [18].

1) Tests under Different Motion Conditions: Table I shows the root mean square error (RMSE) results under different motion speeds and different methods, where LOAM [1] is regarded as the baseline. LIO is our local window optimized odometry method. LIO-raw and LIO-no-ex are the same as LIO expect that the motion compensation or the online extrinsic parameter estimation is cut off, respectively. LIO-mapping is from the results of the mapping with rotaional constraints. The two best results are shown in bold.

From the results, we see that LIO-mapping can always provide accurate estimation of translational (position) and

rotational (orientation) states in all cases. LIO has better performance when motion is faster, which produces more IMU excitation. But it suffers from drift if the motion is slow, since the local map is relatively sparse at this time. The table also shows that with motion compensation and online extrinsic parameter estimation, LIO can provide better performance, especially when motions are rapid.

2) Tests of Drift over Time: To evaluate how the error changes with time, we test the algorithms in a longer test. The first 50 estimated poses are aligned with the ground-truth. The final trajectories from the different methods are shown in Fig. 5, and the translational and rotational errors are shown in Fig. 6. The results show that LIO can provide relatively accurate poses and constrain $roll$ and $pitch$ close to the ground-truth, but it suffers from drift. Neither the method without IMU fusion (LOAM) nor the one with loosely coupled fusion (LOAM+IMU) can provide robust estimation when the motion becomes rapid (in the latter half of the test). LIO-mapping benefits from rotational constraints provided by LIO, and further registers the current sweep to the global map. Thus, it results in less drift of the trajectory and greater consistency of the state estimation.
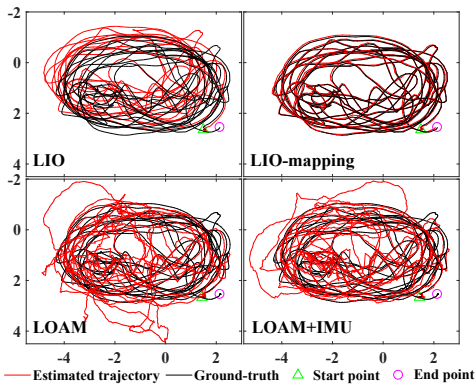

Fig. 5: Trajectories from different methods. LIO can provide relatively accurate poses. Due to the small local window it uses, it has drift when run long-term. LIO-mapping can eliminate the drift with the help of a consistent map. Neither LOAM nor LOAM+IMU can work when the motion becomes rapid (in the latter half of the test).

B. Qualitative Results

Several tests with different sensor configurations and environments are carried out in order to show the improvements in challenging scenarios, including indoor hand-held and outdoor campus golf cart tests, with the configurations as shown in Fig. 4, and tests on the KAIST Urban dataset [19]. Due to the limited space, these pose estimation and mapping results are shown in the supplementary video.

C. Running Time Analysis

We run this test with an Intel i7-7700K CPU at 4.20GHz, 16GB RAM. The lidar intervals vary indoors (0.2s) and outdoors (0.3s), depending on the number of feature points in a sweep (typically more feature points outdoors, around 3000, than indoors, 1000). These intervals help to build
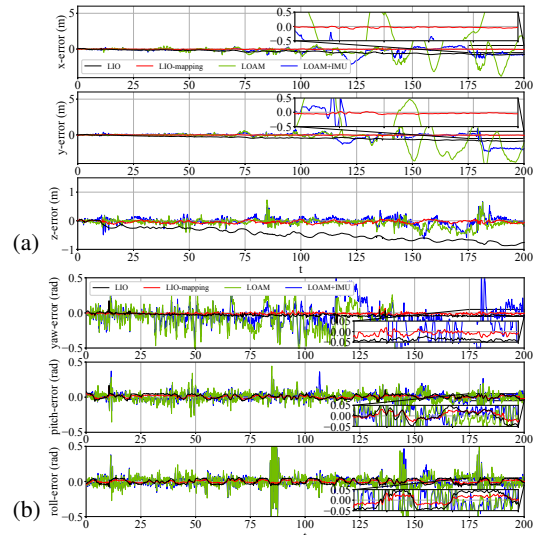

Fig. 6: (a) Translation errors. (b) Rotational errors. In general, our methods (both LIO and LIO-mapping) can provide much smoother results than their counterparts.

TABLE II: Mean running time analyses for 16-line 3D lidar.

Scenarios	Time (ms)		
	Prediction	Odometry	Mapping
Indoor	0.0127	128.7	108.3
Outdoor	0.0102	213.5	167.6

larger maps and skip some of the lidar sweeps to fulfill real-time computation. The mean running time of our method can be found in Table II using the data from a 16-line 3D lidar. The time stands for the processing time of each of the new inputs for a module, i.e., raw IMU measurements, lidar measurements and odometry outputs. Note that the odometry and mapping are in different threads. The mapping thread processes the outputs from the odometry thread. The prediction of the IMU is operated based on the optimized states, alongside the optimization. Thus, it can run as fast as the output rate of the IMU.

VIII. CONCLUSION

A novel tightly coupled lidar-IMU fusion method was presented. It comprised the state optimization for the odometry and the refinement with rotational constraints. The results showed that our method outperformed the state-of-the-art lidar-only method and loosely coupled methods.

Despite the limitation that the proposed method requires initialization, our method indeed showed robust pose estimations results with fast update rate, even under challenging test scenarios, e.g. fast-motion cases, lidar-degraded cases and lidar sweeps with limited overlapping, empowered by sufficient IMU excitation.

REFERENCES

[1] J. Zhang and S. Singh, "Loam: Lidar odometry and mapping in real-time." in *Robotics: Science and Systems*, vol. 2, 2014.

[2] J. Tang, Y. Chen, X. Niu, L. Wang, L. Chen, J. Liu, C. Shi, and J. Hyyppä, "Lidar scan matching aided inertial navigation system in gnss-denied environments," *Sensors*, vol. 15, no. 7, pp. 16 710–16 728, 2015.

[3] S. Lynen, M. W. Achtelik, S. Weiss, M. Chli, and R. Siegwart, "A robust and modular multi-sensor fusion approach applied to mav navigation," in *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on*. IEEE, 2013, pp. 3923–3929.

[4] M. Li, B. H. Kim, and A. I. Mourikis, "Real-time motion tracking on a cellphone using inertial sensing and a rolling-shutter camera," in *Robotics and Automation (ICRA), 2013 IEEE International Conference on*. IEEE, 2013, pp. 4712–4719.

[5] A. Soloviev, D. Bates, and F. Van Graas, "Tight coupling of laser scanner and inertial measurements for a fully autonomous relative navigation solution," *Navigation*, vol. 54, no. 3, pp. 189–205, 2007.

[6] G. Hemann, S. Singh, and M. Kaess, "Long-range gps-denied aerial inertial navigation with lidar localization," in *Intelligent Robots and Systems (IROS), 2016 IEEE/RSJ International Conference on*. IEEE, 2016, pp. 1659–1666.

[7] M. Bosse and R. Zlot, "Continuous 3d scan-matching with a spinning 2d laser," in *Robotics and Automation, 2009. ICRA'09. IEEE International Conference on*. IEEE, 2009, pp. 4312–4319.

[8] C. Park, P. Moghadam, S. Kim, A. Elfes, C. Fookes, and S. Sridharan, "Elastic lidar fusion: Dense map-centric continuous-time slam," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2018, pp. 1206–1213.

[9] T. Lowe, S. Kim, and M. Cox, "Complementary perception for handheld slam," *IEEE Robotics and Automation Letters*, vol. 3, no. 2, pp. 1104–1111, 2018.

[10] 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, 2018.

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

[12] H. Ye, Y. Chen, and M. Liu, "Supplementary material to: Tightly coupled 3d lidar inertial odometry and mapping," Tech. Rep. [Online]. Available: https://sites.google.com/view/lio-mapping

[13] J. Behley and C. Stachniss, "Efficient surfel-based slam using 3d laser range data in urban environments," in *Proc. of Robotics: Science and Systems (RSS)*, 2018.

[14] E. S. Jones and S. Soatto, "Visual-inertial navigation, mapping and localization: A scalable real-time causal approach," *The International Journal of Robotics Research*, vol. 30, no. 4, pp. 407–430, 2011.

[15] S. Agarwal, K. Mierle, and Others, "Ceres Solver," (accessed 22-Aug-2018). [Online]. Available: http://ceres-solver.org

[16] F. Zheng, H. Tang, and Y.-H. Liu, "Odometry-vision-based ground vehicle motion estimation with se (2)-constrained se (3) poses," *IEEE Transactions on Cybernetics*, 2018.

[17] 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, 2017.

[18] S. Umeyama, "Least-squares estimation of transformation parameters between two point patterns," *IEEE Transactions on Pattern Analysis & Machine Intelligence*, no. 4, pp. 376–380, 1991.

[19] J. Jeong, Y. Cho, Y.-S. Shin, H. Roh, and A. Kim, "Complex urban lidar data set," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2018, pp. 6344–6351.