

3D Lidar-IMU Calibration based on Upsampled Preintegrated Measurements for Motion Distortion Correction

Cedric Le Gentil, Teresa Vidal-Calleja and Shoudong Huang

Abstract—In this paper, we present a probabilistic framework to recover the extrinsic calibration parameters of a lidar-IMU sensing system. Unlike global-shutter cameras, lidars do not take single snapshots of the environment. Instead, lidars collect a succession of 3D-points generally grouped in scans. If these points are assumed to be expressed in a common frame, this becomes an issue when the sensor moves rapidly in the environment causing motion distortion. The fundamental idea of our proposed framework is to use preintegration over interpolated inertial measurements to characterise the motion distortion in each lidar scan. Moreover, by using a set of planes as a calibration target, the proposed method makes use of lidar point-to-plane distances to jointly calibrate and localise the system using on-manifold optimisation. The calibration does not rely on a predefined target as arbitrary planes are detected and modelled in the first lidar scan. Simulated and real data are used to show the effectiveness of the proposed method.

I. INTRODUCTION

Autonomous cars and Unmanned Aerial Vehicles (UAVs) are currently transforming the transport and service industries. These systems rely on multiple sensors to achieve any level of autonomy. In the past few years, the robotics community has proposed various multi-sensor fusion algorithms for localisation and mapping. From visual-inertial navigation [1] to visual-lidar odometry and mapping [2], all these techniques rely on accurate extrinsic calibration and synchronisation between the sensing devices. To our knowledge, despite the current popularity of lidars and IMUs ([3] [4]), an automatic procedure to directly calibrate a 6-DoF-IMU/3D-lidar sensor pair has not been proposed in the literature. This work presents a probabilistic framework to estimate the extrinsic calibration between a 3D-lidar and a 6-DoF-IMU.

Despite apparent similarities with the problem addressed in [5], [6], and [7], our work considers a simpler set-up. The cited contributions propose techniques to estimate the relative transformation between a 3D-lidar and an inertial navigation system making use of accurate position information provided by a GNSS or odometry. Considering accelerometers and gyroscopes only, accurate localisation information is not readily available. Therefore, when it comes to determining the relative transformation between a lidar and an IMU, a visual sensor is often used. In fact a variety of camera-IMU and camera-lidar extrinsic calibration procedures can be found in the literature.

Authors are with the Centre for Autonomous Systems at the Faculty of Engineering and IT, University of Technology Sydney, Australia. Email: cedric.legentil@student.uts.edu.au, {teresa.vidalcalleja, shoudong.huang}@uts.edu.au

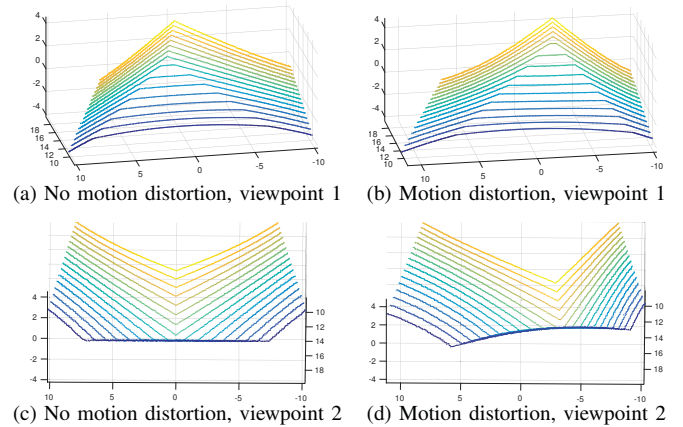


Fig. 1: Lidar point clouds of a room corner with and without motion distortion. In the left column, the lidar is static whereas, in the right one, it moves during the sweep. The motion distortion appears when considering the raw 3D readings as being expressed in a shared “scan frame”. The two rows correspond to different viewpoints.

Over the years, visual-inertial extrinsic calibration became more accurate while getting less restrictive concerning the set-ups needed. Ref. [8] and [9] are examples of complex calibration rigs using actuators and external sensors, such as spinning tables and shaft encoders, to recover the inter-sensor spatial transformation. Free from complex installations, [10] and [11] solve the same problem using only a checkerboard as a calibration target. As for more complex rigs, the accuracy of the estimated calibration parameters depends on the precision with which the target has been built.

The same problem persists when aligning camera images and lidar point clouds as in [12] and [13] for 2D, and in [14] and [15] for 3D scenarios. Accurately built planar visual targets are used for their “visibility” in both modalities. Nonetheless, approaches like [16] and [17] propose target-less calibration pipelines. Both techniques rely on the assumption that high colour gradients in the visual sensor data correspond to high spatial gradients in the 3D-point clouds.

In this paper, we propose a calibration method that does not require a predefined target. Instead, we use a set of arbitrary planes which are automatically extracted and modelled from the first lidar scan to become what we called the *map*. For instance, in our experiments, we opted to use the three planes of a room corner to calibrate our hand-held system. But for outdoor systems, such as large UAVs or cars, a corner of a building and the ground can be used instead.

The first and main issue when it comes to dealing with most of today’s lidars is the sequential nature of the sweeps.

Unlike global-shutter cameras, lidars do not take snapshots of the environment but “progressively scan” the surrounding space. Consequently, every 3D-point is measured from a temporally unique frame. Directly dealing with the lidar points of a scan as being expressed in a shared “scan frame” is equivalent to consider the system static during the sweep. This assumption is commonly made and causes what we know as motion distortion (see Fig. 1). Ideally, each lidar point should be considered as an independent measurement. However, this produces a sensor with an extremely high data rate. Not even the IMU data rate would suffice to characterise motion at the frequency of the ranging measurements.

An additional problem that arises when dealing with multi-sensor systems is the data synchronisation. The work in [11] uses a continuous representation of the state based on basis functions to temporally calibrate a group of sensors. Motivated by this approach, we chose to continuously model our inertial data, instead of the actual state. Gaussian Process (GP) regression [18] is used to interpolate inertial data addressing the issue of “low-frequency” IMU readings compared to the high-frequency point acquisition of the lidar.

Another approach based on continuous-time state estimation is presented in [19]. An efficient form of GP regression is derived to characterise continuous-time trajectories from discrete measurements. This way the state can be queried at any arbitrary timestamp using GP inference. Although appealing, this method cannot be directly applied to our problem because an accurate trajectory cannot be estimated using only IMU readings.

The main contribution of this work is the probabilistic formulation to jointly calibrate and localise lidar-IMU systems. Based on preintegrated measurements [20] over interpolated IMU readings, our formulation provides a temporally precise procedure to remove motion distortion from 3D-point clouds. Each lidar point is reprojected in the first lidar frame where a point-to-plane distance can be computed. An optimisation on manifold is formulated to recover the calibration parameters as well as the IMU poses, velocities, biases, and time-shift.

The remainder of the paper is as follows. Section II details the development of our novel calibration-localisation framework. Both front-end and back-end techniques are explained here. Section III provides key-points of our implementation while Section IV gives evidence of the proposed method performance through simulated and real data experiments. Conclusions and future work are presented in Section V.

II. METHOD

Let us consider a system with a rigidly mounted 3D lidar and a 6-DoF IMU, where \mathbf{R}_c and \mathbf{p}_c respectively represent the relative rotation and translation between the two sensors. \mathbf{R}_c and \mathbf{p}_c are used to project lidar points to the corresponding IMU frames as later shown in (7). The ranging device moves in the environment and provides N 3D-points \mathbf{x}_i at time t_i ($i = 1, \dots, N$), grouped into M scans. The inertial data include a 3-axis accelerometer and a 3-axis gyroscope, that provide respectively the raw readings \mathbf{f}_q and ω_q at time t_q ($q = 1, \dots, Q$). Based on the first lidar

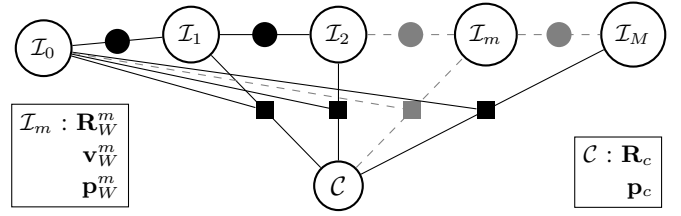


Fig. 2: Factor graph representation of the joint calibration and localisation optimisation. The nodes \mathcal{I}_m represent the IMU poses and velocities at time t_m . The node \mathcal{C} is the set of calibration parameters. The lidar factor, represented by squares, account for reprojected-point-to-target-plane distances. The black circles are the IMU preintegrated factors. Biases and timeshift have been omitted for the sake of readability.

scan, a *map* composed of P planes is built and considered fixed. After the first scan, each lidar point is associated to one of these planes. The plane associated to \mathbf{x}_i is characterised by its normal unit vector \mathbf{n}_i and its distance w_i to the origin. There are only P possible distinct values for \mathbf{n}_i and w_i , in other words, if \mathbf{x}_i and \mathbf{x}_j belong to the same map plane, $\mathbf{n}_i = \mathbf{n}_j$ and $w_i = w_j$. To associate individual lidar points with IMU readings, GP regression is used to independently infer inertial readings on each IMU DoF at any given time t : $\hat{\mathbf{f}}(t)$ and $\hat{\omega}(t)$.

The proposed method aims to estimate the calibration parameters \mathbf{R}_c and \mathbf{p}_c , as well as, the IMU biases, the IMU orientation \mathbf{R}_W^m , position \mathbf{p}_W^m and velocity \mathbf{v}_W^m per lidar scan, and the time-shift δt between the two sensors. The subscript W represents the earth-fixed world frame. The superscript m denotes the m^{th} scan from the lidar and τ_m corresponds to the timestamp of the first lidar point in the m^{th} scan. In the following, \mathcal{S} indicates the state to be estimated: $\mathcal{S} = (\mathbf{R}_c, \mathbf{p}_c, \mathbf{R}_W^0, \dots, \mathbf{R}_W^M, \mathbf{p}_W^1, \dots, \mathbf{p}_W^M, \mathbf{v}_W^0, \dots, \mathbf{v}_W^M, \hat{\mathbf{b}}_f, \hat{\mathbf{b}}_\omega, \delta t)$ with $\hat{\mathbf{b}}_f$, $\hat{\mathbf{b}}_\omega$, and δt the biases and time-shift corrections (more details are given in Section II-D). Note that \mathbf{p}_W^0 is not part of the state as one IMU position needs to be set arbitrarily to define the world frame.

The calibration problem is formulated as a Maximum Likelihood Estimation (MLE):

$$\mathcal{S}^* = \underset{\mathcal{S}}{\operatorname{argmin}} -\log(p(\mathcal{S}|\mathcal{Z})) = \underset{\mathcal{S}}{\operatorname{argmin}} F(\mathcal{S}), \quad (1)$$

with \mathcal{Z} representing the available measurements and F the optimisation cost function. Represented as the factor graph in Fig. 2, and under the assumption of zero-mean Gaussian noise, it can be solved by minimising the point-to-plane distances d_i corresponding to the lidar factors and the residuals \mathbf{r}_I^m associated to the IMU factors. That is

$$F(\mathcal{S}) = \sum_{m=1}^M \|\mathbf{r}_I^m\|_{\Sigma_{r_I^m}}^2 + \sum_{i=1}^N \|d_i\|_{\Sigma_{d_i}}^2. \quad (2)$$

The following notation is utilised in the rest of the paper:

- Σ_\bullet is the covariance matrix of the variable \bullet .
- \mathbf{g} is the known gravity vector in the earth-fixed world frame.
- $\Delta t_k = t_{k+1} - t_k$, $\Delta \tau_m = \tau_{m+1} - \tau_m$ and $\Delta \zeta_m^i = t_i - \tau_m$.

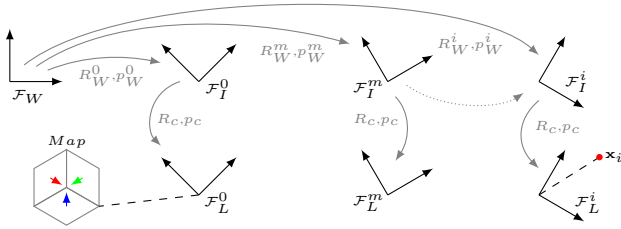


Fig. 3: Frames and transformation during a sequence of measurements. \mathcal{F}_I^0 and \mathcal{F}_L^0 represent respectively the IMU and lidar frames at time t_0 . The plain line arrows are the transformations between the different frames. \mathcal{F}_W is the world fixed frame. The map (set of plane equations) is expressed in \mathcal{F}_L^0 . The dotted line shows the preintegrated measurements used to reproject the point \mathbf{x}_i .

- $\text{Exp}(\cdot)$ is the exponential mapping from axis-angle representations ($\mathfrak{so}(3)$) to rotation matrices ($SO(3)$)¹.
- $\text{Log}(\cdot)$ is the logarithmic mapping from rotation matrices ($SO(3)$) to axis-angle representations ($\mathfrak{so}(3)$)¹.

In order to avoid repetitive integration when the linearisation point changes, the proposed method utilises the IMU preintegrated measurements defined in [1] and [20] as

$$\begin{aligned}\Delta \mathbf{p}_m^i &= \sum_{k=\kappa}^{i-1} \Delta \mathbf{v}_m^k \Delta t_k + \Delta \mathbf{R}_m^k (\mathbf{f}(t_k - \delta t) - \mathbf{b}_f) \Delta t_k^2 \\ \Delta \mathbf{v}_m^i &= \sum_{k=\kappa}^{i-1} \Delta \mathbf{R}_m^k (\mathbf{f}(t_k - \delta t) - \mathbf{b}_f) \Delta t_k \\ \Delta \mathbf{R}_m^i &= \prod_{k=\kappa}^{i-1} \text{Exp}((\boldsymbol{\omega}(t_k - \delta t) - \mathbf{b}_\omega) \Delta t_k),\end{aligned}\quad (3)$$

with $\{\kappa \in \mathbb{N} | t_\kappa = \tau_m\}$. Then

$$\mathbf{p}_W^i = \mathbf{p}_W^m + \Delta \zeta_m^i \mathbf{v}_W^m + \frac{1}{2} \Delta \zeta_m^i^2 \mathbf{g} + \mathbf{R}_W^m \Delta \mathbf{p}_m^i \quad (4)$$

$$\mathbf{v}_W^i = \mathbf{v}_W^m + \Delta \zeta_m^i \mathbf{g} + \mathbf{R}_W^m \Delta \mathbf{v}_m^i \quad (5)$$

$$\mathbf{R}_W^i = \mathbf{R}_W^m \Delta \mathbf{R}_m^i, \quad (6)$$

where \mathbf{b}_f and \mathbf{b}_ω are the accelerometer and gyroscope biases respectively, and δt is the time-shift between the lidar and IMU data. Note that both biases and time-shift are considered as constant along the sequence of readings.

Sections II-A to II-D describe the method's back-end. The data association detailed in Section II-E represents our method's front-end.

A. Lidar factors

The reprojection error of the lidar points into the *map*, defined by point-to-plane distances, is used in the lidar factors. As a point \mathbf{x}_i is expressed in the lidar frame \mathcal{F}_L^i , it needs to be reprojected in the first lidar frame \mathcal{F}_L^0 according to \mathcal{S} and the IMU preintegrated measurements (Fig. 3).

First, \mathbf{x}_i is projected into the IMU frame \mathcal{F}_I^i

$$\mathbf{x}_I^i = \mathbf{R}_c \mathbf{x}_i + \mathbf{p}_c. \quad (7)$$

¹The expressions for these mapping transformations can be found in [1].

Then with (4) and (6), \mathbf{x}_I^i is projected into the world frame

$$\mathbf{x}_W^i = \mathbf{R}_W^i \mathbf{x}_I^i + \mathbf{p}_W^i, \quad (8)$$

with $\tau_m \leq t_i < \tau_{m+1}$. Finally,

$$\mathbf{x}_{L_0}^i = \mathbf{R}_c^\top (\mathbf{R}_W^{0\top} (\mathbf{x}_W^i - \mathbf{p}_W^0) - \mathbf{p}_c) \quad (9)$$

expresses the point in \mathcal{F}_L^0 . And the point-to-plane distances

$$d_i = \mathbf{n}_i^\top \mathbf{x}_{L_0}^i + w_i \quad (10)$$

for $i = 1, \dots, N$, are used as residuals.

B. IMU factors

The IMU factors can be seen as constraints on the IMU poses and velocities. The associated residuals are obtained directly by manipulating (4), (5), and (6):

$$\begin{aligned}\mathbf{r}_p^m &= \mathbf{R}_W^{m\top} (\mathbf{p}_W^{m+1} - \mathbf{p}_W^m - \Delta \tau_m \mathbf{v}_W^m - \frac{\Delta \tau_m^2}{2} \mathbf{g}) - \Delta \mathbf{p}_m^{m+1} \\ \mathbf{r}_v^m &= \mathbf{R}_W^{m\top} (\mathbf{v}_W^{m+1} - \mathbf{v}_W^m - \Delta \tau_m \mathbf{g}) - \Delta \mathbf{v}_m^{m+1} \\ \mathbf{r}_R^m &= \text{Log}(\Delta \mathbf{R}_m^{m+1\top} \mathbf{R}_W^{m\top} \mathbf{R}_W^{m+1}) \\ \text{and } \mathbf{r}_I^m &= [\mathbf{r}_R^m; \mathbf{r}_v^m; \mathbf{r}_p^m].\end{aligned}\quad (11)$$

C. Preintegration on upsampled IMU measurements

As per (8), the proposed method requires preintegrated measurements $\Delta \mathbf{R}_m^i$, $\Delta \mathbf{v}_m^i$, and $\Delta \mathbf{p}_m^i$ for each 3D-point. Because the lidar points have a higher data rate than the IMU, and because the sensors are not synchronised, we need to interpolate the raw IMU readings \mathbf{f}_q and $\boldsymbol{\omega}_q$. GP regression [18] is used independently on each IMU's DoF to obtain $\hat{\mathbf{f}}(t)$ and $\hat{\boldsymbol{\omega}}(t)$ at any time t . Constant mean functions and isotropic Matern covariance functions have been chosen for our application. GP regression is not only useful for estimating upsampled measurements with their associated uncertainty, but also for filtering high-frequency noises from the actual readings due to the chosen smooth kernel. The six GPs are trained in temporal windows as the number of inertial readings in a sequence can be very high.

This interpolation allows computation of synchronised preintegrated measurements, as per (3), with very fine time increments. The preintegrated measurements uncertainty is calculated iteratively as derived in [21] (Section 1.1). Our method uses timestamp-dependent, uncorrelated covariance matrices built from the 6 independent GP-inferred variances at given times. This is different from [21], where a constant covariance matrix is used to represent the noise of the IMU readings at any given time.

D. Biases and time-shift

The preintegration terms in (3) are pre-processed assuming perfect knowledge of the biases and time-shift before any optimisation. This assumption does not hold in general. To address this problem, we adopted the first-order expansion presented in [1] to include a bias correction into our method.

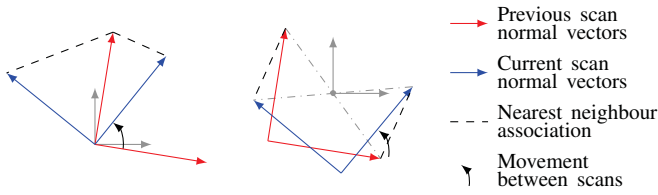


Fig. 4: Nearest neighbour plane matching without/with (left/right) subtraction of centroid for orthogonal normal vectors. Subtraction of the centroid makes the data association more robust.

We extended it to consider a variable time-shift. With $\mathbf{b}_f = \bar{\mathbf{b}}_f + \hat{\mathbf{b}}_f$, $\mathbf{b}_\omega = \bar{\mathbf{b}}_\omega + \hat{\mathbf{b}}_\omega$, and $\delta t = \bar{\delta t} + \hat{\delta t}$,

$$\begin{aligned} \Delta \mathbf{R}_m^i(\mathbf{b}_\omega, \delta t) &\approx \Delta \mathbf{R}_m^i(\bar{\mathbf{b}}_\omega, \bar{\delta t}) \text{Exp} \left(\frac{\partial \Delta \mathbf{R}_m^i}{\partial \mathbf{b}_\omega} \hat{\mathbf{b}}_\omega + \frac{\partial \Delta \mathbf{R}_m^i}{\partial t} \hat{\delta t} \right) \\ \Delta \mathbf{v}_m^i(\mathbf{b}_f, \mathbf{b}_\omega, \delta t) &\approx \Delta \mathbf{v}_m^i(\bar{\mathbf{b}}_f, \bar{\mathbf{b}}_\omega, \bar{\delta t}) + \frac{\partial \Delta \mathbf{v}_m^i}{\partial \mathbf{b}_f} \hat{\mathbf{b}}_f + \frac{\partial \Delta \mathbf{v}_m^i}{\partial \mathbf{b}_\omega} \hat{\mathbf{b}}_\omega + \frac{\partial \Delta \mathbf{v}_m^i}{\partial t} \hat{\delta t} \\ \Delta \mathbf{p}_m^i(\mathbf{b}_f, \mathbf{b}_\omega, \delta t) &\approx \Delta \mathbf{p}_m^i(\bar{\mathbf{b}}_f, \bar{\mathbf{b}}_\omega, \bar{\delta t}) + \frac{\partial \Delta \mathbf{p}_m^i}{\partial \mathbf{b}_f} \hat{\mathbf{b}}_f + \frac{\partial \Delta \mathbf{p}_m^i}{\partial \mathbf{b}_\omega} \hat{\mathbf{b}}_\omega + \frac{\partial \Delta \mathbf{p}_m^i}{\partial t} \hat{\delta t}, \end{aligned} \quad (12)$$

Here $\bar{\bullet}$ denotes the prior knowledge of the value and $\hat{\bullet}$ represents the correction. The state to be estimated includes $\hat{\mathbf{b}}_f$, $\hat{\mathbf{b}}_\omega$, and $\hat{\delta t}$. The derivation of the bias-related Jacobians, as an iterative calculation, can be found in [21]. The time-shift Jacobians can be calculated numerically using a finite difference method.

E. Plane segmentation and data association

The RANSAC-based plane-fitting algorithm [22] is used at each lidar scan to detect the P planes that constitute the calibration target. The distance-threshold between a plane and an inlier point must be large enough to account for the motion distortion and the sensor noise. It is assumed that the sensors system is static during the first scan. This is to ensure the accuracy of the plane equations that will form the *map* of our joint calibration and localisation problem. The point-to-plane data association is an iterative process dealing with the normal vectors of the planes detected in two consecutive scans. For a limited movement between scans, a simple nearest neighbour search links the corresponding planes together. We propose to use a room corner or a building as a calibration target. Exploiting the map geometry for a set of three near-orthogonal planes (Fig. 4), we reinforce the tracking process by subtracting the centroid of the normal vectors before looking for the nearest neighbour.

This iterative procedure relates each lidar point to one of the map planes as shown in Fig. 5. We denote \mathbf{n}_i and w_i the parameters of the initial plane associated to the point \mathbf{x}_i . Having the point-to-plane associations for each scan allows

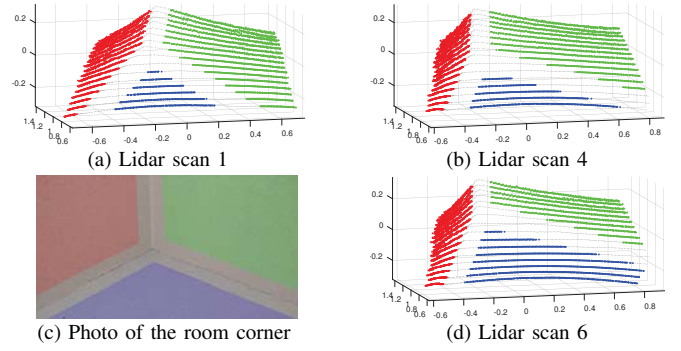


Fig. 5: Plane association in real data. The proposed method associates lidar points from each scan to the map planes. The photo (c) has been coloured to show the data association.

a rough estimation of the lidar poses assuming motionless sweeps. Minimising

$$\mathbf{R}_{L_0}^m, \mathbf{p}_{L_0}^m = \underset{\mathbf{R}_{L_0}^m, \mathbf{p}_{L_0}^m}{\text{argmin}} \sum_{i=\kappa}^{\varrho} \left(\mathbf{n}_i^\top (\mathbf{R}_{L_0}^m \mathbf{x}_i + \mathbf{p}_{L_0}^m) + w_i \right)^2 \quad (13)$$

with $\{\kappa, \varrho \in \mathbb{N} | t_\kappa = \tau_m, t_{\varrho+1} = \tau_{m+1}\}$, returns the lidar pose expressed in the first lidar frame $\mathcal{F}_{L_0}^0$. This information is used to compute the initial guess for the optimisation in (2).

III. IMPLEMENTATION

A. Noise propagation on lidar factors

In standard MLE problems [23], the error function explicitly calculates the difference between the predicted measurement at the current state estimate and the actual noisy measurement. However, the proposed lidar factors' point-to-plane distance function is not bijective. Therefore we cannot utilise an explicit measurement model that provides lidar and IMU preintegrated measurements as a function of the state. Consequently, d_i provides the implicit error function that relates the state and the measurements.

To comply with the zero-mean Gaussian noise assumption from our MLE framework, the noise on d_i is computed by propagating the measurement uncertainties ($\Sigma_{\mathbf{x}_i}$, $\Sigma_{\Delta \mathbf{R}_m^i}$, and $\Sigma_{\Delta \mathbf{p}_m^i}$) through the Jacobians of the reprojection chain and the distance function. Therefore, Σ_{d_i} depends on the state \mathcal{S} . To prevent extensive Jacobian derivations and save processing time, we assume that Σ_{d_i} is locally constant. We first solve the problem with all the measurements covariance matrices set to identity to provide a good initial guess for the noise propagation and to reduce the global optimisation time. In the end, the optimisation pipeline starts with this rough estimation followed by refinement steps in which: Σ_{d_i} is computed with the state estimate from the previous step and several optimisation iterations are executed.

B. Factor balance

Perfect knowledge of the IMU nodes provides very accurate estimates of the calibration parameters. However, wrongly fixed IMU poses affect the calibration accuracy greatly. In practice, the IMU preintegrated measurements

uncertainty is very small compared to the point-to-plane distance uncertainty. Therefore, if there are not enough points per scan, the optimisation in (2) will trust the preintegrated measurements between the IMU nodes over the lidar factors. To a certain extent it is similar to use inexact fixed IMU poses (as the preintegration measurements are not flawless: measurements and integration noise). This leads to poor calibration parameter estimates.

To “naturally balance” the optimisation problem, a large number of points per scan would be needed. Commonly used 3D-lidars do not have the required density, and that number of points would come along with prohibitive processing times. On the other hand, lowering down the IMU factors importance reduces the observability of the calibration parameters. We overcame this problem by independently weighting the IMU residuals covariance matrices in (2):

$$F(\mathcal{S}) = \sum_{m=1}^M \lambda_R \|\mathbf{r}_R^m\|_{\Sigma_{r_R}^m}^2 + \lambda_v \|\mathbf{r}_v^m\|_{\Sigma_{r_v}^m}^2 + \lambda_p \|\mathbf{r}_p^m\|_{\Sigma_{r_p}^m}^2 + \sum_{i=1}^N \|d_i\|_{\Sigma_{d_i}}^2, \quad (14)$$

with λ_\bullet empirically defined as $3 \times N \times M^{-1} \times \text{mean}(\Sigma_{d_i}^{-1}) \times \text{mean}(\text{tr}(\Sigma_{r_\bullet}^m))$.

C. Other implementation choices

The optimisation problem was implemented using *Manopt*'s trust-region solver [24]. This Matlab toolbox provides tools to implement non-linear optimisation on manifolds.

We used *gpml* toolbox [18] to upsample the IMU measurements. The six GP models are trained on M temporal windows as mentioned above. Each of these windows is centred around one lidar scan with an overlap of half a lidar scan period on both the previous and the next scan.

To shorten the processing time we also reduced the number of points used in the optimisation by setting a maximum number of points used per plane per scan (referred as pt/pl/sc in the rest of the paper).

The lidar field-of-view has been reduced to $\pm 45^\circ$ as our real system is hand-held. Thus, the person carrying the sensor suite does not interfere with the lidar data.

IV. EXPERIMENTS AND RESULTS

To demonstrate the performance of the proposed method and to confirm the rightfulness of the assumptions made, experiments on simulated and real data have been conducted.

A. Simulation

The parameters of the simulated data used in this section have been chosen to represent the real system used in Section IV-B:

- 16-channel ($\pm 15^\circ$) rotating lidar providing scans at 10Hz with a density of 240k point per second.
- 3-axis accelerometer and 3-axis gyroscope providing readings at 100Hz.

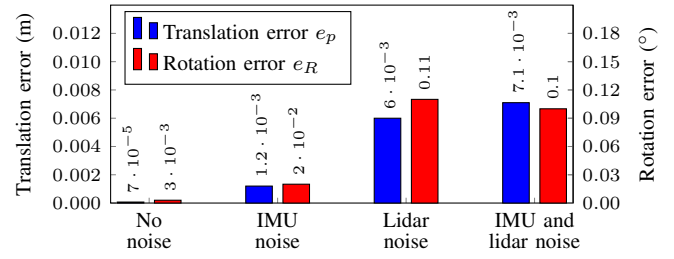


Fig. 6: Calibration accuracy for different noise configurations of simulated data. IMU and lidar can be simulated with either no noise or realistic noise.

- System motion built from sine functions between 0.2 and 0.53Hz with a peak-to-peak amplitude of $\pm 12^\circ$.
- Simulated sequences of 20 seconds.

In the following, the term *realistic noise* refers to the actual specification of the sensor noise from the manufacturer datasheets. The results are evaluated over a 10-runs Monte Carlo simulation. The relative rotation \mathbf{R}_{c_i} and translation \mathbf{p}_{c_i} represent the calibration parameters obtained from the i^{th} run. The evaluation metrics are the mean error on translation and rotation between the ground-truth and the estimated calibration parameters:

$$e_R = \frac{1}{10} \sum_{i=1}^{10} \sqrt{\text{Log}(\mathbf{R}_{c_i}^* \mathbf{R}_c^{GT\top})^\top \text{Log}(\mathbf{R}_{c_i}^* \mathbf{R}_c^{GT\top})},$$

$$e_p = \frac{1}{10} \sum_{i=1}^{10} \sqrt{(\mathbf{p}_{c_i}^* - \mathbf{p}_c^{GT})^\top (\mathbf{p}_{c_i}^* - \mathbf{p}_c^{GT})} \quad (15)$$

with the superscript GT representing the ground-truth values. For convenience, in the rest of the paper, the metric e_R is converted into degrees.

1) *Noise sensitivity*: This set-up studies the impact that different sensor noise combinations have on the overall accuracy of the approach. In this set of experiments, each device is either perfectly deterministic or contains realistic noisy data, depending on the experiment. Fig. 6 presents the results of the different combinations of noise. This figure shows how the method loses much of its accuracy when the lidar uncertainty is considered even if the IMU noise is null. As a matter of fact, GP regression and high frequency preintegration reduce the impact of the inertial noise on the estimated calibration parameters.

2) *Lidar noise sensitivity*: This set-up aims to quantify the lidar noise impact on the estimated calibration parameters. These series of experiments have been run using different values for the lidar's range standard deviation. The results, shown in Fig. 7, are obtained using 100pt/pl/sc. For small standard deviation values, the curve suggests a linear relationship between the proposed method accuracy and the lidar noise.

3) *IMU pose sensitivity*: This set-up aims to analyse the factor balance by studying the impact of errors in the IMU poses. As \mathbf{R}_c and \mathbf{p}_c do not intervene in the IMU residuals, strongly unbalanced factors (e.g. small IMU uncertainty with small pt/pl/sc) is somewhat equivalent to have (almost) fixed

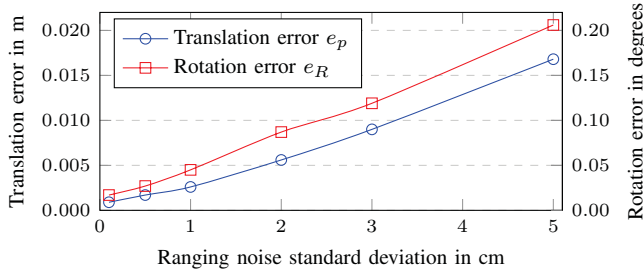


Fig. 7: Calibration accuracy for different lidar noise amplitudes of simulated data. In these experiments, the inertial data is simulated with realistic noise.

	Translational error e_p (m)		Rotational error e_R (°)	
	Normal motion	Fast motion	Normal motion	Fast motion
(i)	5.7e-4	9.5e-4	0.016	0.013
(ii)	1.0e-2	7.97e-2	0.23	0.34

TABLE I: Calibration accuracy with exact known poses and noise-free measurements. The model (i) uses the upsampled preintegrated measurements whereas the model (ii) doesn't.

IMU poses and velocities. We modified our framework to consider the IMU poses and velocities as fixed (i.e. \mathbf{p}_W^m , \mathbf{v}_W^m , and \mathbf{R}_W^m excluded from \mathcal{S}). We compare two cases: one with exact values and the other with approximate values of the IMU poses and velocities. Both experiments used 100pt/pl/sc and realistic noise for the range and inertial measurements. In the first experiment, the ground-truth IMU poses and velocities are used. The second experiment adds a relatively small uniform noise to each ground-truth position, velocity, and orientation axis: $\pm 0.01\text{m}$, $\pm 0.01\text{m/s}$, and $\pm 1.15^\circ$.

Perfect knowledge of the IMU poses and velocities allows an accurate estimation of the calibration parameters with $e_p = 0.0057\text{m}$ and $e_R = 0.02^\circ$. However, a slight perturbation in the IMU states produces large errors $e_p = 0.34\text{m}$ and $e_R = 0.96^\circ$ in the calibration, showing the need for accurate estimation of the IMU nodes.

4) *Upsampled-preintegrated measurements*: This set-up aims to demonstrate the importance of using the upsampled-preintegrated measurements to correct the motion distortion. These series of experiments have been run using fixed exact values for the IMU poses and noise-free sensor data. Two versions of our method are opposed: (i) uses the preintegrated measurements as explained in the previous section, (ii) does not. The values presented in Table I demonstrate better accuracy when using upsampled preintegrated measurements. We also highlight that without the upsampled preintegrated measurements, the system accuracy severely suffers from fast motion (normal motion has been described above, fast motion differs only by its higher frequency: 1.53Hz).

B. Real data

The hardware used for the real data experiments comprises a Velodyne VLP-16 and a Xsens MTi3 IMU (Fig. 8 top left). ROS Xsens driver² and the *snark* driver³ were used

²http://wiki.ros.org/xsens_driver

³<https://github.com/acfr/snark>

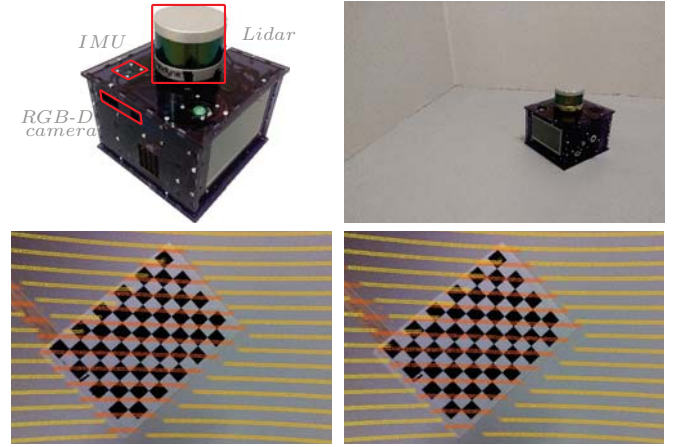


Fig. 8: Top: Experimental set-up and room corner. The camera is not used in the proposed method. Real data have been collected moving the sensor suite approximately a meter away from the room corner. Bottom left: Reprojection of lidar points in the camera image after lidar-camera extrinsic calibration. Bottom right: Reprojection of the lidar points through the chain lidar-IMU (proposed method), IMU-camera.

to collect the IMU and lidar data respectively. Lidar points and IMU measurements were logged with their associated timestamps. Note that these experimental settings do not ensure synchronisation between lidar and IMU data.

A 60-second dataset has been recorded moving the system in front of a room corner (Fig. 8 top right). Particular attention has been given to stimulate the three rotation-axis of the sensing suite. Even though the recording included 577 scans, the optimisation used only 442 scans. This is due to partial observations of the target, not leaving enough lidar points on each plane for the plane segmentation to succeed.

Given the absence of a direct lidar-IMU calibration method in the literature, we decided to benchmark our results with a “chained” calibration: IMU-camera/camera-lidar. The toolbox *Kalibr* [11] was used to estimate the IMU-camera calibration. Although solutions for the camera-lidar calibration exist in the literature, [14] and [15], the low vertical resolution of the VLP-16 makes very challenging the direct use of these solutions. Instead, we opted for a straightforward and simple implementation of a point-to-plane optimisation across multiple lidar-camera static snapshots of a checkerboard. The checkerboard plane equations in the camera frame were estimated thanks to the *RADOCC* toolbox [25].

The proposed lidar-IMU approach using 150pt/pl/sc estimated $\mathbf{R}_c = [2.90 -1.11 179.9]^\circ$ (Euler angles) and $\mathbf{p}_c = [-0.080 0.089 -0.053]\text{m}$. The “chained” calibration computed $\mathbf{R}_c = [1.85 -0.50 -179.3]^\circ$ and $\mathbf{p}_c = [-0.068 0.118 -0.034]\text{m}$. The difference between the two pipelines is summarised with $e_p = 0.036\text{m}$ and $e_R = 1.45^\circ$. The estimated mean linear and angular velocities were respectively 0.7m/s and $26^\circ/\text{s}$.

To provide a qualitative result, Fig. 8 (bottom) shows a comparison of the projected lidar points into the camera images using two different paths: direct lidar-camera and chained lidar-IMU/IMU-camera transformations. These results show that the proposed method can provide accurate

extrinsic calibration with real data. It also demonstrates the robustness of the calibration pipeline regarding partial observation of the target and unsynchronised sensor readings.

Regarding the absence of ground-truth and the difference between the two pipelines, it is difficult to draw strong conclusions from this experiment. The reprojection image, Fig. 8 (bottom left), shows a relatively accurate lidar-camera calibration. Because the motion distortion phenomenon is more pronounced in the lidar data than in the camera images (the Realsense colour camera is a rolling-shutter, not a global-shutter), the camera-IMU calibration is potentially more precise than our lidar-IMU method. Nonetheless, even though two different calibration results have been compounded, Fig. 8 (bottom right) shows a relatively good performance of the proposed method.

V. CONCLUSION

This paper proposes a novel extrinsic calibration framework for 3D lidars and 6 DoF IMUs. This method models the motion distortion in lidar point clouds using preintegration over upsampled IMU readings. The spatial transformation between the two asynchronous sensors and the IMU poses, velocities, biases, and time-shift are jointly estimated by minimising point-to-plane distances between reprojected 3D-points and a set of planes that represents the calibration target. Our implementation with a hand-held sensing suite uses a room corner as target. We conducted a range of simulated experiments to analyse the impact of the sensor noise on the calibration accuracy. On real data using a low-cost IMU, our framework compares with a chained calibration using a camera in the loop.

The proposed calibration procedure can easily be extended to a lidar-IMU pair mounted under a UAV or on a vehicle. For such systems, the calibration target could be the floor, ceiling and walls (indoor or outdoor) of a warehouse or an underground car park. Such systems could also benefit from using the full field-of-view of the lidar.

Moreover, the approach also considers the use of Gaussian process regression on IMU readings to provide high-frequency preintegrated measurements. The interpolation-preintegration pipeline could benefit other techniques such as visual inertial navigation by filtering the raw data noise, reducing the integration noise thanks to the artificially generated higher frequency, and, in general, for synchronising other sensors' measurements.

Future work includes a C++ implementation, with a solver such as *Ceres*⁴, to speed-up the optimisation. Allowing the use of more lidar points, our method would gain in accuracy as the ranging noise is currently the largest source of error. We are also interested in extending this work to a tripartite extrinsic calibration including visual, ranging, and inertial sensors in a single optimisation.

REFERENCES

- [1] C. Forster, L. Carlone, F. Dellaert, and D. Scaramuzza, "IMU preintegration on manifold for efficient visual-inertial maximum-a-posteriori estimation," *Robotics: Science and Systems*, pp. 6–15, 2015.
- [2] J. Zhang and S. Singh, "Visual-lidar Odometry and Mapping: Low-drift, Robust, and Fast," *IEEE International Conference on Robotics and Automation*, pp. 2174–2181, 2015.
- [3] J. Zhang and S. Singh, "LOAM : Lidar odometry and mapping in real-time," *Robotics: Science and Systems*, 2014.
- [4] M. Bosse, R. Zlot, and P. Flick, "Zebedee : Design of a spring-mounted 3-D range sensor with application to mobile mapping," *IEEE Transactions on Robotics*, vol. 28, no. October, pp. 1–15, 2012.
- [5] J. Levinson and S. Thrun, "Unsupervised calibration for multi-beam lasers," *Springer Tracts in Advanced Robotics*, vol. 79, pp. 179–193, 2014.
- [6] Z. Taylor and J. Nieto, "Automatic Markerless Calibration of Multi-Modal Sensor Arrays," *IEEE International Conference on Robotics and Automation*, no. August, pp. 4843–4850, 2015.
- [7] C. Glennie, "Calibration and Kinematic Analysis of the Velodyne HDL-64E S2 Lidar Sensor," *Photogrammetric Engineering & Remote Sensing*, vol. 78, no. 4, pp. 339–347, 2012.
- [8] J. Alves, J. Lobo, and J. Dias, "Camera-inertial sensor modelling and alignment for visual navigation," *International Conference on Advanced Robotics*, vol. 5, no. 3, pp. 103–111, 2003.
- [9] J. Lobo and J. Dias, "Relative pose calibration between visual and inertial sensors," *The International Journal of Robotics Research*, vol. 26, pp. 561–575, 2007.
- [10] J. Kelly and G. S. Sukhatme, "Fast Relative Pose Calibration for Visual and Inertial Sensors," *Springer Tracts in Advanced Robotics*, vol. 54, pp. 515–524, 2009.
- [11] P. Furgale, J. Rehder, and R. Siegwart, "Unified temporal and spatial calibration for multi-sensor systems," *IEEE International Conference on Intelligent Robots and Systems*, pp. 1280–1286, 2013.
- [12] Q. Zhang and R. Pless, "Extrinsic Calibration of a Camera and Laser Range Finder (improves camera calibration)," *IEEE International Conference on Intelligent Robots and Systems*, vol. 3, pp. 2301–2306, 2004.
- [13] S. Sim, J. Sock, and K. Kwak, "Indirect correspondence-based robust extrinsic calibration of LiDAR and camera," *Sensors (Switzerland)*, vol. 16, no. 6, 2016.
- [14] Y. Park, S. Yun, C. S. Won, K. Cho, K. Um, and S. Sim, "Calibration between color camera and 3D LIDAR instruments with a polygonal planar board," *Sensors (Switzerland)*, vol. 14, no. 3, pp. 5333–5353, 2014.
- [15] M. Velas, M. Spanel, Z. Materna, and A. Herout, "Calibration of RGB camera with velodyne LiDAR," *WSCG 2014 Communication Papers Proceedings*, pp. 135–144, 2014.
- [16] Z. Taylor and J. Nieto, "Parameterless automatic extrinsic calibration of vehicle mounted lidar-camera systems," *International Conference on Robotics and Automation: Long Term Autonomy Workshop*, no. October, pp. 3–6, 2014.
- [17] J. Castorena, U. S. Kamilov, and P. T. Boufounos, "Autocalibration of LIDAR and optical cameras via edge alignment," *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 2862–2866, 2016.
- [18] C. E. Rasmussen, C. K. I. Williams, G. Processes, M. I. T. Press, and M. I. Jordan, *Gaussian Processes for Machine Learning*, 2006.
- [19] S. Anderson and T. D. Barfoot, "Full STEAM ahead: Exactly sparse Gaussian process regression for batch continuous-time trajectory estimation on SE(3)," *IEEE International Conference on Intelligent Robots and Systems*, vol. 2015-Decem, no. 3, pp. 157–164, 2015.
- [20] T. Lupton and S. Sukkari, "Visual-inertial-aided navigation for high-dynamic motion in built environments without initial conditions," *IEEE Transactions on Robotics*, vol. 28, no. 1, pp. 61–76, 2012.
- [21] C. Forster, L. Carlone, F. Dellaert, and D. Scaramuzza, "Supplementary Material to : IMU Preintegration on Manifold for Efficient Visual-Inertial Maximum-a-Posteriori Estimation IMU Preintegration : Noise Propagation and Bias Updates," *Technical Report GT-IRIM-CP&R-2015-001*, 2015.
- [22] P. H. S. Torr and A. Zisserman, "MLESAC: A new robust estimator with application to estimating image geometry," *Computer Vision and Image Understanding*, vol. 78, no. 1, pp. 138–156, 1996.
- [23] G. Grisetti, R. Kummerle, C. Stachniss, and W. Burgard, "A Tutorial on Graph-Based SLAM," *IEEE Intelligent Transportation Systems Magazine*, vol. 2, no. 4, pp. 31–43, 2010.
- [24] N. Boumal, B. Mishra, P.-A. Absil, and R. Sepulchre, "Manopt , a Matlab Toolbox for Optimization on Manifolds," *Journal of Machine Learning Research* 15, vol. 15, pp. 1455–1459, 2014.
- [25] A. Kassir and T. Peynot, "Reliable Automatic Camera-Laser Calibration," *Australasian Conference on Robotics and Automation*, 2010.

⁴<http://ceres-solver.org/>