An Efficient LiDAR-IMU Calibration Method Based on Continuous-Time Trajectory

Jiajun Lyu, Jinhong Xu, Xingxing Zuo, Yong Liu

***Abstract*— Sensor calibration is a prerequisite for multi-sensor fusion system. This paper presents a novel method for calibrating the extrinsic transformation between a multi-beam LiDAR and an Inertial Measurement Unit (IMU) based on continuous-time batch optimization. The continuous-time formulation is well suitable for the problem with a large number of measurements, such as the LiDAR points in this calibration problem. Furthermore, since poses at every time instant are available in this continuous-time formulation, LiDAR distortion even under severe motion can be corrected. Experiments on both simulated and real-world datasets are carried out, which confirms the feasibility and high accuracy of the proposed method.**

I. INTRODUCTION

Sensor calibration is an essential fundamental problem when it comes to multi-sensor fusion. And some sensors, such as LiDAR, camera, Inertial Measurement Units (IMU), etc. are usually used together in a fused way for the sake of robustness and high accuracy. Intensive research has gone into camera-IMU calibration and camera-LiDAR calibration. However, However, few works are focusing on the LiDAR-IMU calibration. Inspired by [1], this paper proposes a method based on a continuous-time batch optimization framework to calibrate the extrinsic transformation between a multi-beam LiDAR and an IMU.

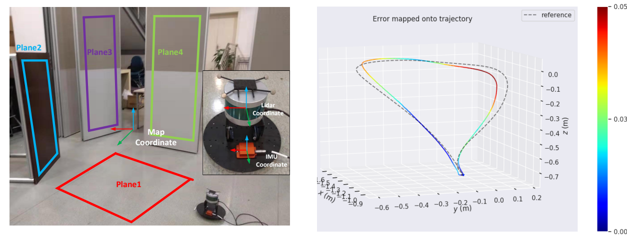
The poses are modeled by a continuous-time trajectory which is defined as B-Spline[2]. LiDAR points will be constrained by point-to-plane distance inside the continuous-time framework. Meanwhile, the continuous-time formulation eases the inclusion of high-rate inertial measurements. Specifically, the main contributions of this paper are as follows:

- We propose a novel method for LiDAR-IMU calibration based on continuous-time batch optimization, which is easy-to-use and efficient.A short time of data collection around 10 Secs is enough for a good calibration result.
- Both simulated and real-world experiments demonstrate the proposed approach can give an accurate extrinsic calibration result.

II. RELATED WORKS

For calibrating a setup of a rigidly connected LiDAR and an IMU, Geiger et al.[3] propose an approach by solving a hand-eye problem[4]. However, it is not easy to get two high-precision trajectories with respect to the IMU and the LiDAR, respectively. Cedric et al.[5] propose an automatic method to calibrate a LiDAR-IMU sensor pair. They use GP regression to interpolate inertial data and based on preintegrated [6] measurements over interpolated IMU readings, the motion distortion of 3D-point clouds can be removed. Furgale et al.[1] propose a continuous-time framework for calibrating the visual-inertial system and Joern et al.[7] extend it to a general approach to calibrate the visual-inertial system with a single beam LiDAR.

All authors are with the Institute of Cyber-Systems and Control, Zhejiang University, Hangzhou, China. Email: {lvjiajun314, xujinhong, xingxingzuo}@zju.edu.cn, yongliu@iipc.zju.edu.cn



(a) (b)

Fig. 1. (a) The environment and the sensor frames in our real-world experiments. The planes play an essential role in the proposed calibration problem. (b) Estimated continuous-time trajectory aligned with the ground-truth. The color indicates ATE.

III. METHOD

Fig.1(a) shows several frames used in this paper. $\{M\}$ denotes the map frame, which is the first IMU reference frame when the calibration is started. The frame of IMU $\{I\}$ is rigidly connected to the LiDAR frame $\{L\}$. We employ B-Spline to parameterize the trajectory in the IMU frame. The transformation from the IMU frame to the map frame at any time t could be expressed as ${}^M_I\boldsymbol{T}(t)$, and ${}^M_I\boldsymbol{T}(t) = \begin{bmatrix} {}^M_I\boldsymbol{R}(t) & {}^M\boldsymbol{p}_I(t) \\ \boldsymbol{0}^T & 1 \end{bmatrix}$. Furthermore, derivatives of the splines with respect to time can be easily computed [8] for generating the linear accelerations ${}^M\boldsymbol{a}(t)$ and angular velocities ${}^M\boldsymbol{\omega}(t)$, which are also reported by IMU sensor in the local IMU reference frame. ${}^L_I\boldsymbol{R}$, ${}^L\boldsymbol{p}_I$ are the extrinsic rotation and translation from IMU to LiDAR frame.

For simplicity, we reconstruct the environment for calibration beforehand by LiDAR odometry, and the RANSAC-based[9] plane-fitting algorithm is used to extract several massive planes from the map. The plane is characterized by its normal unit vector $\boldsymbol{n}$ and its distance d to the origin. The LiDAR odometry also provides the poses in the map at discrete time instants. Let ${}^{L_{t_j}}\boldsymbol{x}$ be a LiDAR point in instantaneous LiDAR frame $\{L_{t_j}\}$ which is collected at time t_j. The NDT [10] algorithm is used to relocate in the map to obtain a pose for each LiDAR scan. With an initial guess

TABLE I
SIMULATION RESULTS

Experment	LiDAR noise	IMU noise	translation error(cm)	rotation error(°)
Exp1	no	no	0.184	0.047
Exp2	yes	no	0.589	0.041
Exp3	no	yes	0.896	0.035
Exp4	yes	yes	1.525	0.088

of ${}^{L}_{I}\boldsymbol{R}$, ${}^{L}\boldsymbol{p}_I$ and the discrete LiDAR poses, we are able to initialize the continuous-time trajectory ${}^{M}_{I}\boldsymbol{T}(t)$. At the same time, we associate some LiDAR points to the planes if the point-to-plane distance is below a certain threshold ϵ:

$$D_{t_j} = \left| [\boldsymbol{n}^T 0] {}^{M}_{I}\boldsymbol{T}(t_j)\boldsymbol{T}^{I}_{L} \begin{bmatrix} {}^{L_{t_j}}\boldsymbol{x} \\ 1 \end{bmatrix} + d \right|, \quad D_{t_j} < \epsilon \quad (1)$$

Only the LiDAR points associated with one of these planes will be considered in the optimization step.

The calibration problem is formulated as a maximum posterior estimation. The state to be estimated is defined as:

$$\mathcal{X} = \{{}^{L}_{I}\boldsymbol{R}, {}^{L}\boldsymbol{p}_I, {}^{M}\boldsymbol{g}, \mathcal{C}_R, \mathcal{C}_p, \mathbf{b}_\omega, \mathbf{b}_\alpha\} \quad (2)$$

where ${}^{M}\boldsymbol{g}$ is the gravity in the map frame. Since the magnitude of gravity is known, we only optimize the direction of gravity. $\mathcal{C}_R$ and $\mathcal{C}_p$ are the control points for the continuous-time trajectory. As only a few seconds datas are needed for the proposed calibration method, the bias of the accelerometer $\mathbf{b}_\omega$ and gyroscope $\mathbf{b}_\alpha$ are considered as constants. The cost function is as follows:

$$J(\mathcal{X}) = \sum_{k\in\mathcal{A}} \left\| {}^{I_k}\boldsymbol{a}_m - {}^{M}_{I}\boldsymbol{R}(t_k)^T \left({}^{M}\boldsymbol{a}(t_k) - {}^{M}\boldsymbol{g}\right) + \boldsymbol{b}_a \right\|^2_{\boldsymbol{\Sigma}_a}$$
$$+ \sum_{k\in\mathcal{W}} \left\| {}^{I_k}\boldsymbol{\omega}_m - {}^{M}_{I}\boldsymbol{R}(t_k)^T {}^{M}\boldsymbol{\omega}(t_k) + \boldsymbol{b}_w \right\|^2_{\boldsymbol{\Sigma}_\omega} + \sum_{j\in\mathcal{L}} \left\| D_{t_j} \right\|^2_{\boldsymbol{\Sigma}_\mathcal{L}} \quad (3)$$

where $\mathcal{A}, \mathcal{W}, \mathcal{L}$ denotes all the linear acceleration, angular velocity, valid LiDAR points measurements, and $\boldsymbol{\Sigma}_a$, $\boldsymbol{\Sigma}_\omega$, $\boldsymbol{\Sigma}_\mathcal{L}$ are the corresponding covariance matrices, respectively. The IMU measurement model could be found in [8] and ${}^{I_k}\boldsymbol{a}_m$, ${}^{I_k}\boldsymbol{\omega}_m$ are the discrete-time raw measurement at time t_k. The optimization framework of Kalibr [1] is employed for solving this nonlinear problem. Since the data sequence for calibration last for only a few seconds and the B-Spline has a good property of local control, this nonlinear least-square problem is sparse and can be solved quickly by Levenberg-Marquardt algorithm.

IV. EXPERIMENT

A. Simulation

To make the simulation as realistic as possible, the characteristics of the simulated sensors are consistent with the actual sensors used in Section IV-B. The typical measurement accuracy of LiDAR is $\pm 2cm$ and zero-mean Gaussian noise is added to the simulated LiDAR measurements. We assume that the additive noise in acceleration and gyroscope measurements are zero-mean, independent Gaussian and refer to the manufacturer datasheets to determine the noise parameters. Four sets of simulated data are generated to observe the impact of the different noise configurations. Table.I presents the final results which shows that the calibration accuracy is sensitive to IMU noise. The probable reason could be that the Signal-to-noise ratio(SNR) of IMU measurements are lower than LiDAR measurements since the motion in simulation is relatively gentle.

TABLE II
ATE OF THE ESTIMATED TRAJECTORIES IN REAL-WORLD EXPERIMENTS

Data	1	2	3	4	5	6
Duration(sec)	32.6	35.1	24.3	12.1	12.4	22.4
ATE(cm)	2.3	2.5	3.4	1.8	3.2	3.0

TABLE III
CALIBRATION RESULTS IN REAL-WORLD EXPERIMENTS

Statistics	mean	std	min	max	median
Translation error(cm)	2.240	0.511	1.237	3.717	2.299
Rotation error (°)	0.737	0.244	0.407	1.263	0.692

B. Real-world Data

We use Velodyne VLP-16 and MTi-300 for real data experiment as shown in Fig. 1(a). The motion capture system gives the ground-truth of the trajectory. To render all quantities of the calibration observable, we ensure that sufficient linear acceleration and rotational velocity exist in all the 6 sequences. The duration of these sequences is among $12-35$ sec. For every sequence, the whole trajectory is evaluated by the average absolute trajectory error(ATE) using EVO [11]. The results are shown in TABLE II which indicates that the spline is able to model the trajectory accurately. Fig.1(b) shows one of the estimated trajectory which is aligned with the ground-truth.

Then the recorded sequences are randomly split into 17 consecutive segments and the duration of segments are among $4-20$ Secs. The final calibration result is shown in TABLE III. The mean of the rotation error is about $0.737°$ while the translation error is $2.24cm$. Note that we compare the estimated results with the relative pose inferred from CAD assembly drawings. Although this inferred extrinsic transformation includes some minor errors inevitably, it can be used for reference as the ground-truth.

V. CONCLUSIONS

In this paper, we propose an efficient LiDAR-IMU calibration approach in a continuous-time batch optimization framework. There are many possible avenues for the future work, such as improving the data association, estimating the time-offset together with the spatial extrinsic transformation.

[1]https://github.com/ethz-asl/kalibr

References

[1] P. Furgale, T. D. Barfoot, and G. Sibley. "Continuous-time batch estimation using temporal basis functions". In: *2012 IEEE International Conference on Robotics and Automation*. IEEE. 2012, pp. 2088–2095.

[2] C. De Boor, C. De Boor, E.-U. Mathématicien, C. De Boor, and C. De Boor. *A practical guide to splines*. Vol. 27. springer-verlag New York, 1978.

[3] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun. "Vision meets robotics: The KITTI dataset". In: *The International Journal of Robotics Research* 32.11 (2013), pp. 1231–1237.

[4] R. Horaud and F. Dornaika. "Hand-eye calibration". In: *The international journal of robotics research* 14.3 (1995), pp. 195–210.

[5] C. Le Gentil, T. Vidal-Calleja, and S. Huang. "3D Lidar-IMU Calibration based on Upsampled Preintegrated Measurements for Motion Distortion Correction". In: *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2018, pp. 2149–2155.

[6] C. Forster, L. Carlone, F. Dellaert, and D. Scaramuzza. "On-Manifold Preintegration for Real-Time Visual–Inertial Odometry". In: *IEEE Transactions on Robotics* 33.1 (2017), pp. 1–21.

[7] J. Rehder, R. Siegwart, and P. Furgale. "A general approach to spatiotemporal calibration in multisensor systems". In: *IEEE Transactions on Robotics* 32.2 (2016), pp. 383–398.

[8] P. Furgale, C. H. Tong, T. D. Barfoot, and G. Sibley. "Continuous-time batch trajectory estimation using temporal basis functions". In: *The International Journal of Robotics Research* 34.14 (2015), pp. 1688–1710.

[9] P. H. S. Torr and A. Zisserman. "MLESAC: A New Robust Estimator with Application to Estimating Image Geometry". In: *Computer Vision and Image Understanding* 78.1 (2000), pp. 138–156.

[10] M. Magnusson. "The three-dimensional normal-distributions transform: an efficient representation for registration, surface analysis, and loop detection". PhD thesis. Örebro universitet, 2009.

[11] M. Grupp. *evo: Python package for the evaluation of odometry and SLAM*. https://github.com/MichaelGrupp/evo. 2017.