IMU-ASSISTED TARGET-FREE EXTRINSIC CALIBRATION OF HETEROGENEOUS LIDARS BASED ON CONTINUOUS-TIME OPTIMIZATION

Zehao Yan[1], *Lin Zhang*[1*], *Zhong Wang*[2], *Shenjie Zhao*[1]

[1] School of Software Engineering, Tongji University, Shanghai, China
[2] Department of Automation, Shanghai Jiao Tong University, Shanghai, China

ABSTRACT

Data fusion of heterogeneous LiDAR systems has gained significant attention due to its potential for providing wide-range sensing and high-density measurements for robots. However, existing LiDAR calibration methods primarily focus on homogeneous LiDAR systems and yield suboptimal outcomes when applied to heterogeneous setups. To this end, this paper proposes an **I**MU-**A**ssisted **He**terogeneous **Li**DAR extrinsics **C**alibration method, namely IA-HeLiC, which is a target-free method based on continuous-time optimization. Specifically, IA-HeLiC utilizes two types of errors, namely geometric constraint error and motion constraint error, and minimizes them within a B-spline-based continuous-time framework to achieve accurate extrinsic calibration. Using a parameter loopback mechanism, this optimization process is performed iteratively to further improve calibration accuracy. IA-HeLiC's performance is validated through experiments using a ground-truth-known handheld device, by which multiple data sequences were collected in diverse real-world scenes. The experiments demonstrate the superior performance of the proposed method compared to its competitors. To make our results reproducible, the source code and the collected dataset have been released at `https://cslinzhang.github.io/IA-HeLiC`.

Index Terms— LiDAR extrinsic calibration, target-free calibration, mechanical spinning LiDAR, solid-state LiDAR, LiDAR data fusion

1. INTRODUCTION

LiDARs enable accurate 3D perception of the surrounding environment and have been widely applied in simultaneous localization and mapping (SLAM), autonomous driving, and autonomous robotics in recent years. At present, the most commonly used LiDARs are the mechanical spinning LiDAR and the solid-state LiDAR. These two types possess complementary characteristics. The mechanical spinning LiDAR has a wide 360-degree FoV, but its measured point cloud is sparse. The solid-state LiDAR provides a denser point cloud despite having a smaller FoV. Additionally, the solid-state LiDAR often comes equipped with an integrated IMU to enhance sensing capabilities. In practice, a heterogeneous LiDAR system can be composed of these two types of LiDARs. Such a system can leverage the wide-area point coverage provided by the mechanical spinning LiDAR, the dense point cloud scanned by solid-state LiDAR and the motion data measured by the IMU.

By fusing the above-mentioned multi-modal data, the heterogeneous LiDAR system can acquire a comprehensive perception. For fusing data from heterogeneous LiDARs, extrinsics are crucial for reprojecting the point clouds from different LiDARs into a common frame. Therefore, obtaining accurate extrinsics between LiDARs is a fundamental requirement for such data fusion. In the extrinsic calibration process of LiDAR systems, the existing methods typically employ geometric constraints and motion constraints. However, on the one hand, the geometric constraints of the existing methods are primarily designed for homogeneous LiDARs, which may not be well adapted to heterogeneous LiDARs. Consequently, these approaches often yield inaccurate calibration results and can even fail when applied to heterogeneous LiDAR systems. On the other hand, for motion constraints, most of the existing methods use discrete-time optimization methods such as graph optimization and Kalman filters. Nevertheless, the LiDAR measurements in a system are often acquired asynchronously, which significantly impacts the calibration accuracy achievable through discrete-time optimization methods.

As an attempt to solve the aforementioned problems, in this paper, we propose an **I**MU-**A**ssisted **He**terogeneous **Li**DAR extrinsics **C**alibration method, namely IA-HeLiC, which is based on continuous-time optimization. The characteristics of IA-HeLiC and our contributions are summarized as follows:

1. To the best of our knowledge, IA-HeLiC is the first heterogeneous LiDAR extrinsic calibration method that considers the geometric constraints of heterogeneous LiDAR point clouds. To generate geometric constraints applicable to both mechanical spinning LiDAR and solid-state LiDAR point clouds, sub-maps of LiDARs are built and then surfels are extracted from the sub-maps. Based on these surfels, we design

[*]Corresponding author. Email: cslinzhang@tongji.edu.cn

a "cross surfel" error by geometrically constraining the distances between LiDAR points and their associated surfels of heterogeneous LiDAR sub-maps. By matching and aligning the LiDAR sub-maps, this novel approach contributes to the overall accuracy of the extrinsic calibration for heterogeneous LiDAR systems.

2. A calibration framework that is based on continuous-time optimization is proposed. This framework leverages B-spline curves to represent the trajectory of the LiDAR system, effectively mitigating the adverse effects of data asynchrony. Under this framework, motion constraints can be constructed on these B-spline curves, and the motion constraint error is jointly minimized with the aforementioned geometric one under continuous-time optimization to accurately calibrate extrinsics.
3. A parameter loopback mechanism is introduced to further improve the accuracy of calibration. This mechanism involves iteratively constructing LiDAR trajectories and sub-maps and subsequently re-optimizing them. Through multiple iterations, the accuracies of the trajectories, sub-maps, and extrinsics are mutually improved, aiming for the global optimum.
4. An adequate heterogeneous LiDAR dataset was established in real-world scenes to validate the performance of IA-HeLiC. This dataset was acquired by a handheld device which was sufficiently motion-activated in all axes of both rotation and translation. This dataset can be employed by other researchers to further study the extrinsic calibration of heterogeneous LiDAR systems.

2. RELATED WORK

2.1. Target-based Approaches

Target-based approaches are considered straightforward approaches for LiDAR extrinsic calibration. Such approaches rely on the utilization of objects with specific geometric or physical characteristics as calibration targets. On the one hand, some of the schemes require pre-made calibration targets. The method proposed by Pusztai *et al.* [1] performs calibration with multiple box-like objects placed in the scene. After that, some solutions [2, 3, 4] that conduct calibration with the assistance of planar objects were proposed, which are simpler to set up compared with the Pusztai *et al.*'s work [1]. In a different manner, Gao and Spletzer's work [5] calibrates with retro-reflective targets on a pole. On the other hand, some methods directly use objects with planar features in the scene for calibration [6, 7, 8]. Unlike the previous ones, the calibration process of these methods does not rely on pre-made targets. However, these approaches require high-quality extraction of targets, which is demanding on the calibration scene.

2.2. Target-free Approaches

Target-free approaches extract geometric constraints and motion constraints from the LiDAR point clouds to calibrate the extrinsics of LiDAR systems. Compared with the target-based ones, these approaches have more relaxed requirements for the calibration scene. Schemes [9, 10] focus on establishing geometric constraints for extrinsic calibration of LiDAR systems. However, they do not consider the motion constraints among different frames of the same LiDAR. Unlike them, the solution proposed by Jiao *et al.* [11] and the approach proposed by Lin *et al.* [12] use methods based on LOAM [13] to obtain the motion constraints. In addition, the calibration process can be aided by motion sensors. Das *et al.* [14] use the global navigation satellite system (GNSS) and its built-in IMU to obtain motion constraints. However, it's worth noting that the majority of existing target-free approaches are primarily designed for homogeneous LiDAR systems and do not account for heterogeneous LiDARs.

3. METHOD

3.1. Framework Overview

The IA-HeLiC framework, illustrated in Fig. 1, takes inputs in the form of LiDAR points and IMU measurements from both the mechanical spinning LiDAR and the solid-state LiDAR. It operates in three phases: initialization, data association, and continuous-time optimization. During the initialization phase, initial values of the LiDAR trajectories and extrinsics are established using the hand-eye calibration framework [15]. Moving on to the data association phase, sub-maps are constructed for each LiDAR based on the initialized trajectories. These sub-maps are then used to extract surfels. After that, the associations between LiDAR points and surfels are established from the sub-maps. In the subsequent continuous-time optimization phase, error terms are generated from these associations. These error terms are utilized to optimize both the extrinsics and the trajectory of the LiDAR system. To improve the accuracy of the extrinsics, IA-HeLiC employs a mechanism called parameter loopback. It involves utilizing the optimized extrinsics and the LiDAR system trajectory to reconstruct the sub-maps and re-optimize the parameters iteratively. Through multiple iterations, IA-HeLiC refines the parameters, thereby improving the accuracy of the extrinsics.

3.2. Trajectory Representation

To address the asynchronous nature of the collected LiDAR measurements, IA-HeLiC employs spatio-temporally continuous B-spline curves to represent the system trajectory. The B-spline curves offer quadratic differentiability, ensuring smoothness in the trajectory and enabling direct derivation of the system acceleration and angular velocity. Moreover,

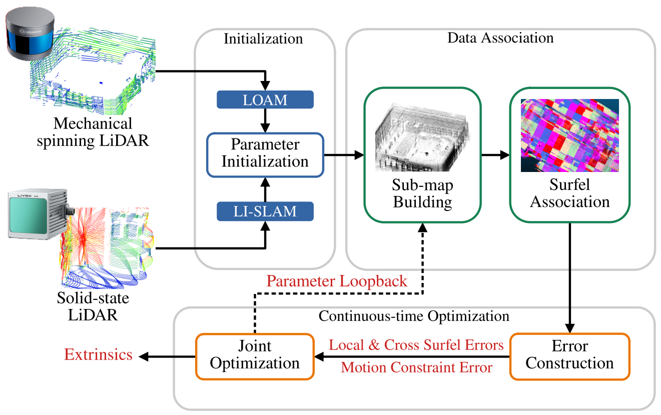


Fig. 1. Workflow of IA-HeLiC. IA-HeLiC takes heterogeneous LiDAR data as input and estimates parameters in the initialization phase. Then parameters are iteratively refined through the data association phase and continuous-time optimization phase.

B-spline curves facilitate local control and optimization, simplifying the optimization process. In this paper, we denote the global frame by $(\cdot)^G$, and the LiDAR system's pose in this frame at moment t is denoted by $\mathbf{T}_S^G(t)$. Inspired by the work [16], we build two cubic B-spline curves to represent the trajectory: one for the orientation and one for the position. Specifically, $\mathbf{T}_S^G(t)$ can be split into the orientation in the quaternion form $\mathbf{q}_S^G(t) \in \mathbb{R}^4$ and position $\mathbf{t}_S^G(t) \in \mathbb{R}^3$. Both orientation and position B-spline curves are segmented into multiple time intervals of length Δt. The i-th time interval is denoted by $[t_i, t_i + \Delta t)$. At t_i, there is a control point for each B-spline curve. According to [17, 18], at the moment $t \in [t_i, t_i + \Delta t)$, $\mathbf{t}_S^G(t)$ and $\mathbf{q}_S^G(t)$ are represented by their corresponding control points respectively as,

$$\mathbf{t}_S^G(t) = \mathbf{t}_i + \sum_{j=1}^{3} \tilde{B}_{i+j}(t)(\mathbf{t}_{i+j} - \mathbf{t}_{i+j-1}), \tag{1}$$

$$\mathbf{q}_S^G(t) = \mathbf{q}_i \otimes \prod_{j=1}^{3} \mathrm{Exp}\left(\tilde{B}_{i+j}(t)\mathrm{Log}\left(\bar{\mathbf{q}}_{i+j-1} \otimes \mathbf{q}_{i+j}\right)\right), \tag{2}$$

where $\mathbf{t}_i \in \mathbb{R}^3$ denotes the control point of the position B-spline at t_i, quaternion $\mathbf{q}_i \in \mathbb{R}^4$ denotes the control point of the orientation B-spline at t_i, $\otimes$ denotes quaternion multiplication, $\bar{\mathbf{q}}_{i+j-1}$ denotes the conjugate of $\mathbf{q}_{i+j-1}$, $\mathrm{Exp}(\cdot)$ denotes the exponential mapping from $\mathfrak{so}(3)$ to $\mathbb{SO}(3)$ while $\mathrm{Log}(\cdot)$ denotes its inverse mapping, and $\tilde{B}_i(\cdot)$ is the i-th cumulative B-spline basis function of degree 3.

Taking advantage of the quadratic differentiability of B-spline curves, the acceleration $\mathbf{a}^S(t)$ and angular velocity $\boldsymbol{\omega}^S(t)$ of the LiDAR system at moment t in its local frame $(\cdot)^S$ can be derived as,

$$\boldsymbol{\omega}^S(t) = \left(\left(\mathbf{R}_S^G(t)\right)^\top \dot{\mathbf{R}}_S^G(t)\right)^\vee, \tag{3}$$

$$\mathbf{a}^S(t) = \left(\mathbf{R}_S^G(t)\right)^\top (\ddot{\mathbf{t}}_S^G(t) - \mathbf{g}^G), \tag{4}$$

where $\mathbf{R}_S^G(t) \in \mathbb{R}^{3\times 3}$ is the rotation matrix corresponding to $\mathbf{q}_S^G(t)$, $\dot{\mathbf{R}}_S^G(t)$ denotes the first-order derivative of $\mathbf{R}_S^G(t)$ w.r.t. time at t, $\ddot{\mathbf{t}}_S^G(t)$ denotes the second-order derivative of $\mathbf{t}_S^G(t)$ w.r.t. time at t, $\mathbf{g}^G \in \mathbb{R}^3$ denotes the gravity acceleration expressed in the global frame, and $(\cdot)^\vee$ denotes the mapping from 3×3 skew-symmetric matrices to $\mathbb{R}^3$.

3.3. Initialization

Prior to parameter optimization, the trajectory and the extrinsics of the LiDAR system are initialized with rough estimations. First, we introduce the initialization of the system trajectory. On the one hand, the orientation B-spline curve of the trajectory, denoted by $\mathcal{T}_\mathbf{q}$, is initialized using the IMU's gyroscope readings in the system frame, $(\cdot)^S$. Specifically, the gyroscope reading at moment t_i is denoted by $\tilde{\boldsymbol{\omega}}_{(t_i)}^S$, where i is the index of the IMU measurement. We initialize $\mathcal{T}_\mathbf{q}$ by solving the following least-square problem,

$$\mathcal{T}_\mathbf{q} = \underset{\mathcal{T}_\mathbf{q}}{\mathrm{argmin}} \left(\frac{1}{2}\sum_i ||\tilde{\boldsymbol{\omega}}_{(t_i)}^S - \boldsymbol{\omega}^S(t_i)||_2^2\right), \tag{5}$$

where t_i is the moment of gyroscope measurement, $||\cdot||_2^2$ is the square of the $\mathcal{L}_2$ norm, and $\boldsymbol{\omega}^S(\cdot)$ denotes the system's angular velocity given by Eq. (3). On the other hand, as the subsequent optimization is insensitive to the initial trajectory position, we adopt a simple initialization strategy for the position B-spline by setting all its position control points to zero.

Next, we initialize the extrinsics of the LiDAR system, which consist of frame transforms from each LiDAR to the system. To begin the initialization process, an arbitrarily chosen LiDAR, L_0, is taken as the reference LiDAR. After that, during the initialization, we first estimate the frame transforms between L_0 and other LiDARs, and then initialize the frame transforms between the system and L_0.

To estimate the frame transforms between LiDARs, we utilize existing well-established LiDAR odometries to obtain the local trajectory of each LiDAR. For mechanical spinning LiDARs, we use LOAM [13], and for solid-state LiDARs, we adopt the LiDAR-inertial odometry, LI-Init [19]. It is worth mentioning that due to the asynchronous nature of odometry outputs, we align the timestamps of poses using the linear interpolation on Lie algebra presented in [20], and then the time origins of all LiDAR trajectories are aligned with the global time origin. After the local trajectories are obtained, we can then estimate the frame transforms between different LiDARs through hand-eye calibration [15]. Specifically, we denote k-th LiDAR by L_k and denote the pose of L_k at moment t_i in its mapping frame by $\tilde{\mathbf{T}}_{(t_i)}^{L_k} \in \mathbb{SE}(3)$. The relationship between the poses of L_k and L_0 at timestamp t_i can be given as,

$$\tilde{\mathbf{T}}_{(t_i)}^{L_0} \, \mathbf{T}_{L_k}^{L_0} = \mathbf{T}_{L_k}^{L_0} \, \tilde{\mathbf{T}}_{(t_i)}^{L_k}, \tag{6}$$

where t_i is the i-th timestamp of the LiDAR trajectory, and $\mathbf{T}_{L_k}^{L_0}$ is the frame transform between L_k and L_0. With multiple timestamps, $\mathbf{T}_{L_k}^{L_0}$ can then be solved as an $AX = XB$ problem [15].

Once we have obtained the frame transforms between each LiDAR sensor and the reference LiDAR, we proceed to initialize the transform $\mathbf{T}_{L_0}^{S}$ between the frame of L_0 and the system frame. Specifically, the rotation component of $\mathbf{T}_{L_0}^{S}$, denoted by $\mathbf{R}_{L_0}^{S}$, intertwines the orientations of L_0 and the system, which is given by the following equation,

$$\mathbf{R}_{L_0}^{S} \, \tilde{\mathbf{R}}_{(t_i)}^{L_0} = \mathbf{R}_{S}^{G}(t_i) \, \mathbf{R}_{L_0}^{S}, \tag{7}$$

where $\tilde{\mathbf{R}}_{(t_i)}^{L_0}$ is the orientation component of $\tilde{\mathbf{T}}_{(t_i)}^{L_0}$, and $\mathbf{R}_{S}^{G}(t_i)$ denotes the rotation matrix of the LiDAR system at t_i. To initialize $\mathbf{R}_{L_0}^{S}$, we solve Eq. (7) with the method in [21]. Finally, with both $\mathbf{R}_{L_0}^{S}$ and $\mathbf{T}_{L_k}^{L_0}$ solved, we initialize the rotation component of the extrinsics for L_k as $\mathbf{R}_{L_k}^{S} = \mathbf{R}_{L_0}^{S} \mathbf{R}_{L_k}^{L_0}$, where $\mathbf{R}_{L_k}^{L_0}$ represents the orientation component of $\mathbf{T}_{L_k}^{L_0}$. The translation components of all extrinsics are simply set to zero, similar to the initialization of the system position. With these steps, the extrinsics of L_k, $\mathbf{T}_{L_k}^{S}$, are initialized.

3.4. Data Association

Following the initialization stage, we have acquired the discrete trajectory for each LiDAR. In this stage, our objective is to extract surfels and establish data associations for the LiDAR points. To accomplish this, we begin by constructing a sub-map for each LiDAR. This is achieved by projecting the LiDAR points onto its corresponding mapping frame, utilizing the initialized discrete trajectory. It's worth mentioning that, pose interpolation is employed to handle the discreteness of the trajectory. After the construction of LiDAR sub-maps, extraction of geometric features is required to obtain final data associations. Two commonly used geometric features for LiDAR sub-maps are edge features and surfels. However, in the calibration scenes, surfels tend to be more abundant and less influenced by LiDAR noise. Therefore, in IA-HeLiC, we choose to use surfels for data association. To extract surfels, we voxelize each sub-map and then compute the second-order moments of each voxel. Subsequently, based on the work [22], we use the following equation to estimate the likelihood l_p that the point cloud within a voxel represents a surfel, and l_p is given as,

$$l_p = 2 \frac{\lambda_1 - \lambda_0}{\lambda_0 + \lambda_1 + \lambda_2}, \tag{8}$$

where $\lambda_0 \leq \lambda_1 \leq \lambda_2$ are the three eigenvalues of the second-order moment matrix of the voxel. The closer l_p is to 1, the more likely that points in the voxel form a surfel. In our approach, we pick a threshold of $l_p > 0.8$ to select candidate surfels and fit their plane equations under the RANSAC framework. Finally, we establish data associations by associating the points within a short distance (< 0.05m) from the surfel with that particular surfel. Except for associations in the same sub-map, we also establish associations between points and surfels in different sub-maps. To achieve this, the points are projected onto other sub-maps using the initial LiDAR frame transforms, and associations are similarly established with the surfels nearby.

3.5. Continuous-time Optimization

In this stage, we perform a joint optimization of all control points in the B-spline curves $\mathcal{T}$ representing the system trajectory as well as the extrinsics of the heterogeneous LiDAR system $\mathcal{E} = \{\mathbf{T}_{L_k}^{S}\}$. This optimization allows us to accurately calibrate the extrinsics. In the continuous-time framework, we build geometric constraints from the aforementioned point-surfel associations, obtaining the local surfel error $e_i^{\mathrm{LS}} \in \mathbb{R}$ and the cross surfel error $e_i^{\mathrm{CS}} \in \mathbb{R}$. Also, we derive the angular velocity and acceleration of the LiDAR system from the trajectory B-spline curves to build motion constraints, resulting in two sorts of motion constraint error: the angular velocity error $\mathbf{e}_i^{\boldsymbol{\omega}} \in \mathbb{R}^3$ and the acceleration error $\mathbf{e}_i^{\mathbf{a}} \in \mathbb{R}^3$. An illustration of the errors is given in Fig. 2. Based on these errors, we formulate the continuous-time optimization as the following minimization problem,

$$\begin{aligned}\mathcal{E}, \mathcal{T} = & \underset{\mathcal{E},\mathcal{T}}{\operatorname{argmin}} \Big(\frac{w_{\mathrm{LS}}}{2} \sum_i \left(e_i^{\mathrm{LS}}\right)^2 + \frac{w_{\mathrm{CS}}}{2} \sum_j \left(e_j^{\mathrm{CS}}\right)^2 \\ & + \frac{w_{\boldsymbol{\omega}}}{2} \sum_k ||\mathbf{e}_k^{\boldsymbol{\omega}}||_2^2 + \frac{w_{\mathbf{a}}}{2} \sum_l ||\mathbf{e}_l^{\mathbf{a}}||_2^2 \Big),\end{aligned} \tag{9}$$

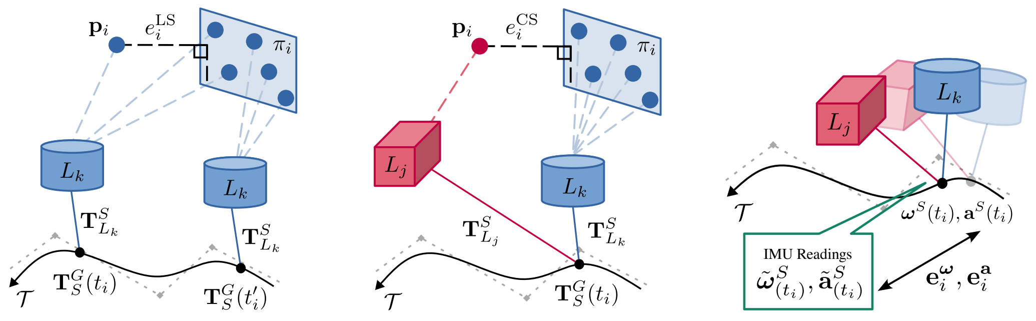
where w_{LS}, w_{CS}, $w_{\boldsymbol{\omega}}$, and $w_{\mathbf{a}}$ are the weights of the corresponding errors, respectively. Subsequently, we introduce each of these errors individually.

Local surfel error. To establish the geometric constraint that a surfel point lies on its associated surfel within the same sub-map, e_i^{LS} is constructed as the point-plane distance between the surfel point and the surfel within the sub-map, which is given as,

$$e_i^{\mathrm{LS}} = \operatorname{dist}\left(\mathbf{p}_i^{L_k}, \pi_i^{L_k}\right), \tag{10}$$

$$\mathbf{p}_i^{L_k} = \left(\mathbf{T}_{L_k}^{S}\right)^{-1} \mathbf{T}_{S}^{G}(t_i) \, \mathbf{T}_{L_k}^{S} \, \tilde{\mathbf{p}}_i^{L_k(t_i)}, \tag{11}$$

where $\tilde{\mathbf{p}}_i^{L_k(t_i)} \in \mathbb{R}^3$ denotes the raw LiDAR measurement of i-th surfel point, L_k denotes the measuring LiDAR, t_i denotes the measurement timestamp, $\mathbf{p}_i^{L_k}$ denotes the projection of $\tilde{\mathbf{p}}_i^{L_k(t_i)}$ in the mapping frame of L_k, $\mathbf{T}_{S}^{G}(t_i)$ denotes the system pose at t_i given by Eq. (1) and Eq. (2), $\operatorname{dist}(\cdot,\cdot)$ denotes the point-plane Euclidean distance, and $\pi_i^{L_k}$ denotes of the



(a) Local surfel error (b) Cross surfel error (c) Motion constraint error

Fig. 2. Illustrations of the error terms involved in IA-HeLiC. The local surfel error and cross surfel error are constructed as distances from LiDAR points to corresponding surfels, while the motion constraint error is constructed based on motion estimations from the IMU.

surfel associated with $\tilde{\mathbf{p}}_i^{L_k(t_i)}$ in the mapping frame of L_k. e_i^{LS} characterizes the internal roughness of each sub-map.

Cross surfel error. Drawing upon the geometric constraint that a surfel point in one LiDAR point cloud lies on its associated surfel in the heterogeneous LiDAR point cloud, e_i^{CS} is formed as the point-plane distance between the projection of the surfel point to the heterogeneous sub-map and the corresponding surfel, which is given as,

$$e_i^{\mathrm{CS}} = \operatorname{dist}\left(\mathbf{p}_i^{L_m}, \pi_i^{L_m}\right), \tag{12}$$

$$\mathbf{p}_i^{L_m} = \left(\mathbf{T}_{L_m}^S\right)^{-1} \mathbf{T}_S^G(t_i)\, \mathbf{T}_{L_k}^S\, \tilde{\mathbf{p}}_i^{L_k(t_i)}, \tag{13}$$

where m denotes the index of the sub-map, $\mathbf{p}_i^{L_m}$ denotes the projection of $\tilde{\mathbf{p}}_i^{L_k(t_i)}$ in the mapping frame of L_m, and $\pi_i^{L_m}$ denotes the surfel associated with $\tilde{\mathbf{p}}_i^{L_k(t_i)}$ in that mapping frame. The cross surfel error e_i^{CS} assesses the degree of alignment between the heterogeneous LiDAR sub-maps when reprojected with the extrinsics, thereby serving as a measure of the extrinsic accuracy.

Motion constraint error. $\mathbf{e}_i^{\boldsymbol{\omega}}$ and $\mathbf{e}_i^{\mathbf{a}}$ are defined as the difference between the motion measurements obtained from the IMU and the motion estimates derived from the trajectory. After transforming IMU readings to the LiDAR system frame, $\mathbf{e}_i^{\boldsymbol{\omega}}$ and $\mathbf{e}_i^{\mathbf{a}}$ are given as,

$$\mathbf{e}_i^{\boldsymbol{\omega}} = \tilde{\boldsymbol{\omega}}_{(t_i)}^S - \boldsymbol{\omega}^S(t_i), \tag{14}$$

$$\mathbf{e}_i^{\mathbf{a}} = \tilde{\mathbf{a}}_{(t_i)}^S - \mathbf{a}^S(t_i), \tag{15}$$

where i is the index of IMU measurement, t_i is the measurement timestamp, $\tilde{\boldsymbol{\omega}}_{(t_i)}^S$ denotes the angular velocity measured by the IMU's gyroscope at t_i, $\tilde{\mathbf{a}}_{(t_i)}^S$ denotes the acceleration measured by the IMU's accelerometer at t_i, $\boldsymbol{\omega}^S(\cdot)$ denotes system angular velocity given by Eq. (3), and $\mathbf{a}^S(\cdot)$ denotes system acceleration given by Eq. (4). These error terms describe the tracking accuracy of the system trajectory. Given that the IMU typically has a higher measurement frequency compared to LiDARs, optimizing with these error terms can establish a fine-grained control of the trajectory.

Once all the aforementioned errors have been constructed, we proceed to solve the non-linear least-square problem presented in Eq. (9). By solving this problem, we can calibrate the extrinsics of the LiDAR system and refine the continuous system trajectory.

Parameter loopback. The sub-maps in Sec. 3.4 are built using the discrete LiDAR trajectories, whose accuracy is affected by the employed odometries. Considering that they are not optimal when building geometric constraints, we take the optimized extrinsics and system trajectory obtained thus far and feed them back to the data association stage (see Sec. 3.4) to rebuild the sub-maps. Since LiDAR trajectories are represented by continuous B-spline curves now, no pose interpolation is further required in sub-map construction. Utilizing the optimized parameters as initial values, the re-optimization is then performed on these new geometric constraints to achieve a more accurate calibration. Noticing that the above process can be looped, we iterate multiple times to obtain the final extrinsics of the heterogeneous LiDAR system.

4. EXPERIMENT

4.1. Experimental Setup

To evaluate the performance of IA-HeLiC, data from heterogeneous LiDAR systems are required for extrinsic calibration. However, existing heterogeneous LiDAR datasets do not have enough pose changes and motion activations, making them suboptimal for calibration problems. To this end, we built a handheld data collection device (pictured) to collect our calibration dataset. The device is equipped with a ROBOSENSE RS-LiDAR-16 mechanical spinning LiDAR and a Livox Avia solid-state LiDAR. The Livox Avia LiDAR has a built-in 6-axis 200 Hz IMU. These sensors are rigidly attached to the handheld device and their orientations and positions can be obtained directly from the device's readings. Our dataset collected with this device contains multiple sequences of sensor data in various real-world scenes, each lasting more than 1 minute. Included in each sequence is data captured through complete rotation and motion of the device in all axes.

IA-HeLiC was implemented using ROS Noetic and PCL libraries in C++17 and executed on a computer with an Intel Xeon CPU E5-2630 v3 @ 2.40GHz $\times$ 32. In all experiments, $w_{\text{LS}} = 2$, $w_{\text{CS}} = 10$, $w_{\omega} = 18.5$, $w_{\mathbf{a}} = 28$, and the time interval of B-spline control points Δt was set to 0.2s.

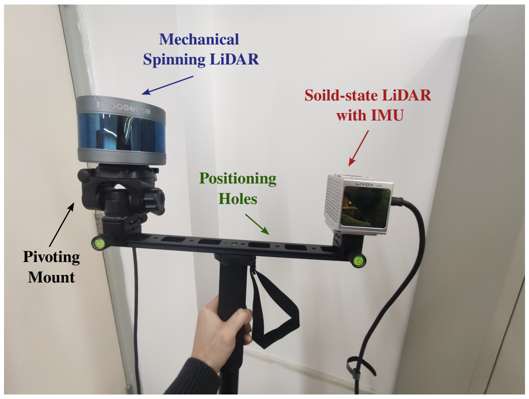


Fig. 3. The handheld device used to collect the test dataset for all experiments. Positions and orientations can be read from the positioning holes and the pivoting mount on the device.

4.2. Qualitative Results

To have an intuitive understanding of the calibration accuracy of IA-HeLiC, we reprojected the heterogeneous LiDAR data into a unified frame using the calibrated extrinsics and system trajectory, as shown in Fig. 4. The results reveal a high level of consistency in the heterogeneous LiDAR point clouds when reprojected using the calibrated extrinsics, indicating that the extrinsics calibrated by IA-HeLiC exhibit a significant degree of accuracy. Moreover, upon examination of the reprojected point clouds, it shows that planar objects such as whiteboards appear flat, and smaller objects like chairs are distinguishable, which further attests to the superior calibration accuracy achieved by IA-HeLiC.

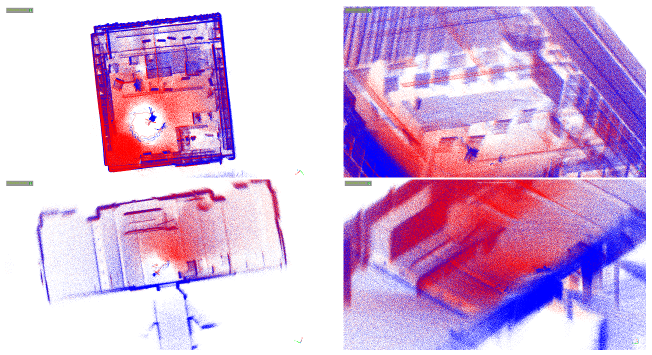


Fig. 4. Reprojection results. Blue points are from the mechanical spinning LiDAR, while red points are from the solid-state LiDAR. The reprojected maps are clear and point clouds of different LiDARs have a good consistency.

4.3. Quantitative Results

Given the lack of existing extrinsic calibration methods specifically designed for heterogeneous LiDAR systems, we chose to compare IA-HeLiC with two alternative approaches: a state-of-the-art extrinsic calibration method for homogeneous LiDAR systems [10] and a method based on normal distributions transform (NDT) [23]. These competitors, along with IA-HeLiC, were executed on all available data sequences to obtain calibrated extrinsics. To evaluate the calibration accuracy, we measured the calibration errors by calculating the difference between calibrated extrinsics and the ground-truth ones. The ground-truth was directly obtained from the readings of our handheld device. The mean rotational error in each axis $e_{\text{roll}}, e_{\text{pitch}}, e_{\text{yaw}}$ and the mean translational error in each axis e_x, e_y, e_z of all compared methods are presented in Table 1. The results indicate that IA-HeLiC exhibits a significant accuracy advantage compared to its competitors, both in terms of rotation and translation, suggesting the effectiveness of our IA-HeLiC in the extrinsic calibration of heterogeneous LiDAR systems.

Table 1. Calibration errors of IA-HeLiC and compared methods on our collected dataset.

Error	Liu *et al.* [10]	NDT [23]	IA-HeLiC
e_{roll} ($^\circ$)	4.37	1.18	**0.75**
e_{pitch} ($^\circ$)	5.36	1.01	**0.47**
e_{yaw} ($^\circ$)	4.81	1.39	**0.44**
e_x (m)	0.244	0.034	**0.022**
e_y (m)	0.356	0.056	**0.016**
e_z (m)	0.332	0.039	**0.015**

Table 2. Calibration errors under different configurations of IA-HeLiC in ablation study.

Error	WoPL	WoCSE	WoMCE	IA-HeLiC
e_{roll} ($^\circ$)	0.98	1.47	**0.62**	0.75
e_{pitch} ($^\circ$)	0.53	0.79	0.48	**0.47**
e_{yaw} ($^\circ$)	0.45	0.86	0.60	**0.44**
e_x (m)	0.027	0.060	**0.022**	**0.022**
e_y (m)	0.021	0.040	0.018	**0.016**
e_z (m)	0.028	0.087	0.016	**0.015**

4.4. Ablation Studies

In IA-HeLiC, three key mechanisms, namely the cross surfel error, the motion constraint error, and the parameter loopback, all play crucial roles in ensuring calibration accuracy. To verify our claims, we try to justify the effectiveness of these three components, respectively. We mainly compared IA-HeLiC against three baselines, which were 1) WoPL: The parameter loopback is deactivated; 2) WoCSE: The cross surfel error is removed; and 3) WoMCE: The motion constraint error is excluded. These baselines and IA-HeLiC were all evaluated on our collected dataset, and relative quantitative experimental results are presented in Table 2. From Table 2, it can be found that IA-HeLiC overperforms all other counterparts in calibration accuracy, corroborating the effectiveness of aforementioned three key mechanisms involved in IA-HeLiC.

5. CONCLUSION

In this paper, we studied an emerging problem in the field of robotics, extrinsic calibration of heterogeneous LiDARs, and proposed a solution namely IA-HeLiC. IA-HeLiC constructs both geometric constraints and motion constraints and then the errors are jointly minimized under the continuous-time framework. Moreover, IA-HeLiC incorporates a parameter loopback mechanism to further improve calibration accuracy. The superiority of IA-HeLiC in the extrinsic calibration accuracy is verified by extensive experimental results on our own collected real-world dataset. Future work involves the extrinsic calibration of systems with heterogeneous LiDARs and other types of sensors.

6. REFERENCES

[1] Z. Pusztai, I. Eichhardt, and L. Hajder, "Accurate calibration of multi-LiDAR-multi-camera systems," *Sensors*, vol. 18, no. 7, pp. 2139, 2018.

[2] D.-H. Kim and G.-W. Kim, "Efficient calibration method of multiple LiDARs on autonomous vehicle platform," in *Int'l Conf. Big Data Smart Comput.*, 2020, pp. 446–447.

[3] J. Kim, C. Kim, and H. J. Kim, "Robust extrinsic calibration for arbitrarily configured dual 3D Lidars using a single planar board," in *Int'l Conf. Control Automat. Systems*, 2020, pp. 576–580.

[4] W. Lee, C. Won, and J. Lim, "Unified calibration for multi-camera multi-LiDAR systems using a single checkerboard," in *IEEE/RSJ IROS*, 2020, pp. 9033–9039.

[5] C. Gao and J. R. Spletzer, "On-line calibration of multiple LIDARs on a mobile vehicle platform," in *IEEE ICRA*, 2010, pp. 279–284.

[6] D.-G. Choi, Y. Bok, J.-S. Kim, and I. S. Kweon, "Extrinsic calibration of 2-D Lidars using two orthogonal planes," *IEEE Trans. Robotics*, vol. 32, no. 1, pp. 83–98, 2016.

[7] J. Jiao, Q. Liao, Y. Zhu, T. Liu, Y. Yu, R. Fan, L. Wang, and M. Liu, "A novel dual-Lidar calibration algorithm using planar surfaces," in *IEEE Intell. Vehicles Symp.*, 2019, pp. 1499–1504.

[8] J. M. Maroli, Ü. Özgüner, K. Redmill, and A. Kurt, "Automated rotational calibration of multiple 3D LIDAR units for intelligent vehicles," in *IEEE ITSC*, 2017, pp. 1–6.

[9] X. Liu and F. Zhang, "Extrinsic calibration of multiple LiDARs of small FoV in targetless environments," *IEEE RA-L*, vol. 6, no. 2, pp. 2036–2043, 2021.

[10] X. Liu, C. Yuan, and F. Zhang, "Targetless extrinsic calibration of multiple small FoV LiDARs and cameras using adaptive voxelization," *IEEE Trans. Instrum. Meas.*, vol. 71, pp. 1–12, 2022.

[11] J. Jiao, Y. Yu, Q. Liao, H. Ye, R. Fan, and M. Liu, "Automatic calibration of multiple 3D LiDARs in urban environments," in *IEEE/RSJ IROS*, 2019, pp. 15–20.

[12] J. Lin, X. Liu, and F. Zhang, "A decentralized framework for simultaneous calibration, localization and mapping with multiple LiDARs," in *IEEE/RSJ IROS*, 2020, pp. 4870–4877.

[13] J. Zhang and S. Singh, "Low-drift and real-time lidar odometry and mapping," *Auton. Robots*, vol. 41, no. 2, pp. 401–416, 2017.

[14] S. Das, L. Klinteberg, M. Fallon, and S. Chatterjee, "Observability-aware online multi-Lidar extrinsic calibration," *IEEE RA-L*, vol. 8, no. 5, pp. 2860–2867, 2023.

[15] R. Y. Tsai and R. K. Lenz, "A new technique for fully autonomous and efficient 3D robotics hand/eye calibration," *IEEE Trans. Robot. Automat.*, vol. 5, no. 3, pp. 345–358, 1989.

[16] H. Ovrén and P.-E. Forssén, "Trajectory representation and landmark projection for continuous-time structure from motion," *Int'l J. Robot. Res.*, vol. 38, no. 6, pp. 686–701, 2019.

[17] A. Haarbach, T. Birdal, and S. Ilic, "Survey of higher order rigid body motion interpolation methods for keyframe animation and continuous-time trajectory estimation," in *Int'l Conf. 3D Vis.*, 2018, pp. 381–389.

[18] C. Sommer, V. Usenko, D. Schubert, N. Demmel, and D. Cremers, "Efficient derivative computation for cumulative B-splines on Lie groups," in *IEEE/CVF CVPR*, 2020, pp. 11145–11153.

[19] F. Zhu, Y. Ren, and F. Zhang, "Robust real-time LiDAR-inertial initialization," in *IEEE/RSJ IROS*, 2022, pp. 3948–3955.

[20] S. Bansal and A. Tatu, "Affine interpolation in a Lie group framework," *ACM ToG*, vol. 38, no. 4, pp. 1–16, 2019.

[21] Z. Yang and S. Shen, "Monocular visual–inertial state estimation with online initialization and camera–IMU extrinsic calibration," *IEEE Trans. Automat. Sci. Eng.*, vol. 14, no. 1, pp. 39–51, 2017.

[22] M. Bosse and R. Zlot, "Continuous 3D scan-matching with a spinning 2D laser," in *IEEE ICRA*, 2009, pp. 4312–4319.

[23] P. Biber and W. Straßer, "The normal distributions transform: A new approach to laser scan matching," in *IEEE/RSJ IROS*, 2003, pp. 2743–2748.