PGO-LIOM: Tightly-coupled LiDAR-Inertial Odometry and Mapping via Parallel and Gradient-free Optimization

Hongming Shen, Qun Zong, Bailing Tian, Xuewei Zhang, and Hanchen Lu

Abstract—Real-time localization and mapping for micro aerial vehicles (MAV) is a challenging problem, due to the limitation of the onboard computational power. In this paper, a tightly coupled light detection and ranging (LiDAR)inertial odometry is developed, which achieves high accuracy, real-time trajectories estimation for MAV utilizing only onboard sensors and a low-power onboard computer. The key idea of the proposed method is to integrate the IMU measurements, correct LiDAR matching measurements, Li-DAR matching outliers into one nonlinear and noncontinuous objective function, and formulate the localization and mapping problem as a stochastic optimization problem. To deal with the nonlinear and noncontinuous objective function, a gradient-free optimization method is proposed to solve the stochastic optimization problem with a single parallel iteration. The novel constructed objective function and gradient-free optimization algorithm enable the proposed LiDAR-inertial odometry to achieve high accuracy and low time consumption. The effectiveness of the proposed method is demonstrated through various scenarios, including public datasets and real-world flight experiments.

Index Terms—LiDAR-based perception, Aerial robots, Sensor fusion, State estimation, Parallel processing.

I. INTRODUCTION

L OCALIZATION and mapping is a fundamental task in developing autonomous MAV. With the deployment of the autonomous MAV, it has the potential to enable surveillance, information gathering, and exploration. Both of these missions rely on reliable state estimation for MAV. During the last decade, the global navigation satellite system (GNSS) [1], motion capture systems [2], and ultra-wideband (UWB) [3] are well-applied for MAV-based applications. However, these navigation systems rely on external sensings are only functional under certain environmental conditions. For the real-world application, MAVs have to travel through a variety of diverse environments or operate in previously unknown unstructured

This work was supported in part by the National Key Research and Development Program of China 2018AAA0102401, in part by the National Natural Science Foundation of China under Grant 62073234, Grant 62022060, Grant 61873340, Grant 62273249, and in part by the Integration of Industry, Education and Research Innovation Fund of China University Grant 2021ZYA08009. (Corresponding Author: Bailing Tian)

Hongming Shen, Qun Zong, Bailing Tian, Xuewei Zhang and Hanchen Lu are with the School of Electrical and Information Engineering, Tianjin University, Tianjin 300072, China (e-mail: shenhm@tju.edu.cn; zongqun@tju.edu.cn; bailing_tian@tju.edu.cn; zhangxuewei@tju.edu.cn; luhanchen_1993@tju.edu.cn). rough terrain where external sensors are difficult to set up. With the goal of achieving a fully autonomous application for MAV in different scenarios flexibly, it is necessary to develop a real-time localization and mapping system that can work without external sensors. Thanks to the continuously reduced cost and weight of LiDAR in recent years, it has become a viable onboard sensing payload for MAV. Compared with visual navigation systems (VNS), LiDAR-based localization and mapping systems can provide more reliable state estimation thanks to the direct depth measurement, which is immune to scene illumination and texture changes. The aforementioned character is critical for the aerial platforms since they are easily destroyed when state estimation failures occur. Despite the above advantages, the large number point cloud generated by LiDAR worsens the computational complexity. Different from the ground vehicles, which are easy to take laptop, or even desktop onboard, a low-power onboard computer is the preference for MAV to improve its duration of the flight. Hence, a LiDAR-based navigation system that can provide real-time localization and mapping with a low-power onboard computer is crucial for autonomous MAV applications.

A. Related Work

Real-time localization and mapping is a challenging problem for LiDAR navigation systems (LNS). In [4], a portable LiDAR-based localization and mapping system is developed through the point cloud registration technology represented by iterative closest point (ICP) [5], generalized iterative closest point (GICP) [6], and normal distribution transformation (NDT) [7], which can provide pose estimation by the dense point cloud. Processing the dense point cloud is very computationally costly, with the aim of reducing the computational cost of [4], a voxelization strategy is adopted in LiTAMIN [8] and LiTAMIN2 [9] to downsample a large number of the raw point cloud, and a high-performance graphics processing unit (GPU) is used to accelerate the localization and mapping procedure. However, different from the low-power GPU preferred by MAVs, the high-performance GPU is difficult to take onboard, due to the power and weight limitation of MAVs. In DLO [10], a lightweight LNS is developed through the GICP matching method in the context of the DARPA Subterranean Challenge. DLO enables the use of dense point cloud to provide realtime pose estimation for MAV, through the pruning of GICP matching with an adaptive keyframe strategy. Nonetheless,

IEEE TRANSACTIONS ON INDUSTRIAL ELECTRONICS

the keyframe strategy is a trade-off between accuracy and real-time performance. Feature-based LNS extracts geometric features from LiDAR scans to improve the real-time performance without sacrificing localization and mapping accuracy. LOAM [11] is a popular localization and mapping algorithm, which can provide low-drift pose estimation and mapping. Simple geometric feature extraction strategies are introduced in LOAM, where the feature points are segmented into edge features and planar features based on local smoothness, and the robot pose is estimated by doing edge-to-edge and planar-toplanar matching procedure. However, LiDAR-only localization and mapping methods will struggle to perform robustly during high-speed motion. Therefore, LiDAR is typically fused with an IMU for robust localization and mapping. LIMO [12], a tightly-coupled LiDAR-inertial odometry, introduced a bundle adjustment (BA) into LNS to jointly optimize measurements from LiDAR and IMU. Although BA has good performance in the VNS field, for LiDAR odometry which has 10 to 100 times of feature points than the visual odometry, BA is designed to process all the LiDAR features in the local window, the realtime performance is difficult to achieve by onboard computing power. With the aim of improving the real-time performance of the BA process performed in LIOM [12], LINS is presented in [13], which uses an error-state Kalman filter to fuse the measurements from LiDAR and IMU in a tightly-coupled manner. In LIO-SAM [14], one of the state-of-the-art LiDARinertial odometry is proposed by fusing the IMU preintegration factor and scan-to-map matching factor into a factor graph, which can achieve real-time with onboard computing power thanks to the keyframe strategy. In CLINS [15], a continuous time LiDAR-inertial system is proposed by adopting a nonrigid registration using B-splines. With the aim of achieving localization and mapping for autonomous MAV, FAST-LIO is presented in [16], a Sherman-Morrison-Woodbury equation is adopted to reduce the computation complexity of the iterated extended Kalman filter (IEKF) update procedure. To further improve the real-time performance of FAST-LIO, an incremental KD-tree [17] is implemented in FAST-LIO2 [18], which significantly improves the efficiency of LiDAR matching and mapping procedure.

B. Motivations and Contributions

In view of the aforementioned analysis, LiDAR-based localization and mapping generally adopt the voxelization method [8], [9], keyframe strategy [10], [14], effective data structures [18] to improve LiDAR matching efficiency and solve the localization and mapping problem through nonlinear optimization or the Kalman filter. These methods are the Gaussian approximation of the maximum a posteriori (MAP) problem solved through the iterative least-squares optimization. Only correct LiDAR matchings are used in the aforementioned methods to make observations satisfying the Gaussian distribution assumption. However, discarded incorrect LiDAR matchings may help improve the accuracy, and the iterative process is time-consuming due to the data association procedure in each iterative. With the goal of designing a computationally efficient and high-accuracy localization and mapping system, in this work, a tightly-coupled LiDARinertial odometry is developed with a single parallel-iteration solver. The key idea of the proposed method is to integrate the IMU measurements, correct LiDAR matching measurements, LiDAR matching outliers into one nonlinear and noncontinuous objective function, and formulate the localization and mapping problem as a stochastic optimization problem. With gradient-free optimization, the nonlinear and noncontinuous stochastic optimization problem is solved through a Monte-Carlo sampling in real-time. The main contributions of this paper are listed as follow:

- 1) A tightly-coupled localization and mapping approach is proposed which considers IMU measurements, the likelihood of correct LiDAR matchings, and the number of incorrect LiDAR matchings simultaneously.
- 2) To deal with the nonlinear and noncontinuous localization and mapping problem, a gradient-free optimization algorithm is proposed to solve the optimization with a single parallel iteration, which significantly improves the realtime performance of the localization and mapping process.
- 3) The proposed method is fully paralleled, which can provide real-time localization and mapping with low-power onboard computation power. Moreover, a real-world autonomous flight is performed to demonstrate the application of the proposed LiDAR-inertial odometry.

II. SYSTEM OVERVIEW

A system overview of the proposed LiDAR-inertial odometry is given in Fig. 1, which takes advantage of measurements from LiDAR and IMU in a tightly-coupled manner. The LiDAR point cloud measurements are fed into the feature extraction algorithm proposed in LOAM [11], which extracts features on edge and planer by calculating the curvature of each LiDAR point. In order to perform the LiDAR matching process, the extracted features are registered to a voxel map. The voxel map management module is in charge of maintaining a local map in a Hash table through the adaptive voxelization method [19] and the singular value decomposition (SVD) technique (Section III-B). The LiDAR matching cost is constructed by formulating the LiDAR matching problem as a maximum likelihood estimation (MLE) problem (Section IV-A2). Thanks to the acceptance-rejection sampling [20], which provides a criterion for judging whether a LiDAR matching is correct, the LiDAR matching cost considers the likelihood of correct matchings and the number of incorrect matchings simultaneously. To deal with the nonlinear and noncontinuous cost functions, a gradient-free optimizationbased sensor fusion method is proposed to approximate the optimal solution through a fully paralleled Monte-Carlo sampling (Section IV-B).

III. PROBLEM FORMULATION

Consider a localization and mapping system which is modeled by taking advantage of measurements from a LiDAR, and an IMU. The full state vector is defined as:

$$\mathbf{x} = [\mathbf{t}^{\top}, \mathbf{v}^{\top}, \mathbf{R}^{\top}, \mathbf{b}_{a}^{\top}, \mathbf{b}_{g}^{\top}]^{\top}$$
(1)

IEEE TRANSACTIONS ON INDUSTRIAL ELECTRONICS



Fig. 1. System overview of the proposed LiDAR-inertial odometry.

where t is the translation vector, v is the velocity vector, $\mathbf{R} \in SO(3)$ is the rotation matrix, \mathbf{b}_a and \mathbf{b}_g are IMU accelerometer and gyro biases, respectively.

Different from traditional LiDAR-inertial odometry, the present work solves the localization and mapping problem as a stochastic optimization problem.

$$\hat{\mathbf{u}} = \arg\min \mathbb{E}_{\boldsymbol{\tau} \sim \mathcal{Q}} \left[\mathbf{S}(\mathbf{x}, \boldsymbol{\tau}) \right]$$
(2)

where τ is treated as a random vector generated by a Gaussian noise process $\tau \sim Q(\mathbf{u}, \sigma_u)$, $\mathbf{u} = [\boldsymbol{\omega}^{\top}, \mathbf{a}^{\top}]^{\top}$, which includes the angular velocity $\boldsymbol{\omega}$ and local liner acceleration \mathbf{a} , is the mean vector of τ , and σ_u is the covariance matrix of τ . $\mathbb{E}_{\tau \sim Q}[\cdot]$ denotes the expectation operation over the random vector τ with respect to Q, which is abbreviated as $\mathbb{E}_Q[\cdot]$ in the rest of this paper. $\mathbf{S}(\mathbf{x}, \tau)$ is the cost function which is constructed through IMU measurements residual $\mathbf{r}_I(\mathbf{x}, \tau)$ and LiDAR measurements residual $\mathbf{r}_L(\mathbf{x}, \tau)$.

$$\mathbf{S}(\mathbf{x}, \boldsymbol{\tau}) = \mathbf{r}_{I}(\mathbf{x}, \boldsymbol{\tau}) + \mathbf{r}_{L}(\mathbf{x}, \boldsymbol{\tau})$$
(3)

At time t_k , $\mathbf{S}(\mathbf{x}, \tau)$ is a function of the state vector $\mathbf{x}_{t_{k-1}}$ and the random vector $\tau_{t_{k-1}}$. $\mathbf{x}_{t_{k-1}}$ is the state vector estimated at time t_{k-1} , which is a constant vector at time t_k . $\tau_{t_{k-1}}$ is a random vector with a mean vector of $\mathbf{u}_{t_{k-1}}$. At time t_k , if the mean vector $\mathbf{u}_{t_{k-1}}$ can be solved from (2), the state vector \mathbf{x}_{t_k} can be estimated by substituting $\hat{\mathbf{u}}_{t_{k-1}}$ and $\hat{\mathbf{x}}_{t_{k-1}}$ into the kinematic model (5).

A. Kinematic Model

The kinematic model is defined as:

$$\mathbf{t} = \mathbf{v}, \dot{\mathbf{v}} = \mathbf{R}\mathbf{a}, \mathbf{R} = \mathbf{R}[\boldsymbol{\omega}]_{\wedge}$$

$$\dot{\mathbf{b}}_{\boldsymbol{\omega}} = \mathbf{0}_{3\times 1}, \dot{\mathbf{b}}_{\mathbf{a}} = \mathbf{0}_{3\times 1}$$
(4)

where $[\cdot]_{\wedge}$ denotes the skew-symmetric cross-product matrix. The state estimated at time t_k can be represented as a discretetime kinematic model of the form

$$\mathbf{x}_{t_k} = F(\mathbf{x}_{t_{k-1}}, \mathbf{u}_{t_{k-1}}) \tag{5}$$

where $F(\cdot)$ denotes the state-transition function.

B. Map Representation

Following our previous work [21], an adaptive voxelization method [19] is adopted to maintain a local map for LiDAR-inertial odometry. The local map is divided into a set of voxels, each voxel containing a group of LiDAR points $\mathbf{p}_i (i = 1, ..., N)$ indexed in a Hash table. Assuming that each

voxel m in the local map is subject to Gaussian distributions $\mathbf{m} \sim \mathcal{N}(\mathbf{m}^{\mu}, \boldsymbol{\sigma}_{\mathbf{m}}).$

$$\mathbf{m}^{\mu} = \frac{1}{N} \sum_{i=1}^{N} \mathbf{p}_{i}$$

$$\boldsymbol{\sigma}_{m} = \frac{1}{N-1} \sum_{i=1}^{N} (\mathbf{p}_{i} - \mathbf{m}^{\mu}) (\mathbf{p}_{i} - \mathbf{m}^{\mu})^{\top}$$
(6)

where \mathbf{m}^{μ} and $\boldsymbol{\sigma}_{m}$ are mean and covariance matrix of the Gaussian distribution. The voxels in the local map can be divided into different shapes depending on the relationships between the eigenvalues of the covariance matrix $\boldsymbol{\sigma}_{m}$. A SVD is performed to calculate the eigenvalues of the covariance matrix $\boldsymbol{\sigma}_{m}$.

$$\boldsymbol{\sigma}_{\mathbf{m}} = \begin{bmatrix} \mathbf{V}_1, \mathbf{V}_2, \mathbf{V}_3 \end{bmatrix} \begin{bmatrix} \lambda_1 & 0 & 0\\ 0 & \lambda_2 & 0\\ 0 & 0 & \lambda_3 \end{bmatrix} \begin{bmatrix} \mathbf{V}_1, \mathbf{V}_2, \mathbf{V}_3 \end{bmatrix}^{\top} \quad (7)$$

where λ_1 , λ_2 , and λ_3 are eigenvalues of the covariance matrix σ_m in descending order. $\mathbf{V}_1, \mathbf{V}_2, \mathbf{V}_3$ are eigenvectors that correspond to $\lambda_1, \lambda_2, \lambda_3$, respectively.

Hence, voxels in the local map can be divided into 3 categories:

- Plane voxel: A voxel that satisfies λ₂ ≫ λ₃. A plane voxel is represented by a centroid m^{μ_p} and a normal vector m^{n_p} = V₃ of the plane.
- 2) Edge voxel: A voxel that is not a plane voxel and satisfies λ₁ ≫ λ₂. An Edge voxel is represented by a centroid m^{μ_e} and a normal vector m^{n_e} = V₁ of the edge.
- Candidate voxel: A voxel that is neither a plane voxel nor edge voxel. Candidate voxel is not considered in cost function construction and needs more points to cluster plane voxel or edge voxel.

IV. TIGHTLY COUPLED LIDAR-INITIAL ODOMETRY

A. Construction of the Cost Function

The state of the LiDAR-inertial odometry is estimated based on sensors observation. For the proposed tightly coupled LiDAR-inertial odometry, IMU measurements cost and LiDAR matching cost are considered, which are described in more detail below.

1) *IMU measurements cost:* The IMU measurements model is defined as:

$$\mathbf{a}_m = \mathbf{a} + \mathbf{R}^\top \mathbf{g} + \mathbf{b}_a + \mathbf{n}_a$$

$$\boldsymbol{\omega}_m = \boldsymbol{\omega} + \mathbf{b}_g + \mathbf{n}_g$$
(8)

where \mathbf{a}_m and $\boldsymbol{\omega}_m$ are raw IMU measurements, g is the constant gravity vector, \mathbf{n}_a and \mathbf{n}_g are the Gaussian white noise of acceleration and gyroscope measurements.

Thanks to the novel formulation in (2), the IMU measurements residual is defined in (9) instead of the complicated IMU preintegration formulation [22].

$$\mathbf{r}_{I}(\mathbf{x},\boldsymbol{\tau}) = \left\| \begin{aligned} \mathbf{a}_{m} - \hat{\mathbf{a}} - \hat{\mathbf{R}}^{\top} \mathbf{g} - \hat{\mathbf{b}}_{a} \\ \boldsymbol{\omega}_{m} - \hat{\boldsymbol{\omega}} - \hat{\mathbf{b}}_{g} \end{aligned} \right\|_{2} \tag{9}$$

For the gradient-free optimization described in Section IV-B, $\hat{\omega}$ and \hat{a} are approximated through the Monte-Carlo sampling over τ .

IEEE TRANSACTIONS ON INDUSTRIAL ELECTRONICS

2) LiDAR matching cost: The objective of the LiDAR matching process is to construct the state-dependent cost through geometric constraints between feature points and the voxel map. When a new LiDAR scan arrives, edge features and planer features are extracted by the algorithm proposed in [11]. The LiDAR matching problem can be formulated as a maximum likelihood estimation (MLE) problem given below.

$$\hat{\mathbf{T}} = \underset{\mathbf{T}}{\operatorname{arg\,max}} \prod_{i=1}^{n} L(\mathbf{T}^{B}\mathbf{f}_{i}|^{I}\mathbf{M})$$

$$= \underset{\mathbf{T}}{\operatorname{arg\,min}} \sum_{i=1}^{n} - \log(L(^{I}\mathbf{f}_{i}|^{I}\mathbf{M}))$$
(10)

where $\mathbf{T} \in SE(3)$ is the transformation matrix. *n* is the number of feature points in the current LiDAR scan. $L({}^{I}\mathbf{f}_{i}|^{I}\mathbf{M})$ represents the likelihood of observing a feature point ${}^{I}\mathbf{f}_{i}$ generated from the voxel map \mathbf{M} .

Inspired by [23], the likelihood function $L({}^{I}\mathbf{f}_{i}|^{I}\mathbf{M})$ can be simplified through the Gaussian mixture model (GMM).

$$L({}^{I}\mathbf{f}_{i}|{}^{I}\mathbf{M}) = L({}^{I}\mathbf{f}_{i}|{}^{I}\mathbf{m}_{j})$$
(11)

where ${}^{I}\mathbf{m}_{j}$ is the voxel correspondence to feature ${}^{I}\mathbf{f}_{i}$, which is found through the feature-to-map matching (FMM) according to the hash indexing.

However, the hash indexing is a simple feature corresponding strategy, which may lead to a fraction of incorrect matchings. Introducing matching outliers into (10) can compromise the correctness of the resulting estimate. Hence, an acceptancerejection algorithm [20] is adopted to refine the matching results. With the assumption that each voxel is subject to a Gaussian distribution, the target distribution is defined as

$$f({}^{I}\mathbf{f}_{i}|{}^{I}\mathbf{m}_{j}) = \frac{1}{\left[(2\pi)^{3}|\boldsymbol{\sigma}_{m,j}|\right]^{\frac{1}{2}}} \exp\left[-\frac{1}{2}({}^{I}\mathbf{f}_{i}-\boldsymbol{m}_{j}^{\mu})^{\top}\boldsymbol{\sigma}_{m,j}^{-1}({}^{I}\mathbf{f}_{i}-\boldsymbol{m}_{j}^{\mu})\right] \quad (12)$$

The upper bound of the target distribution $\bar{f}({}^{I}\mathbf{f}_{i}|{}^{I}\mathbf{m}_{i})$ can be defined as

$$f({}^{I}\mathbf{f}_{i}|{}^{I}\mathbf{m}_{j}) \leq \bar{f}({}^{I}\mathbf{f}_{i}|{}^{I}\mathbf{m}_{j}) = \frac{1}{\left[\left(2\pi\right)^{3}|\boldsymbol{\sigma}_{m,j}|\right]^{\frac{1}{2}}}$$
(13)

Using the acceptance-rejection sampling for each feature point, the criterion for judging whether a feature point ${}^{I}\mathbf{f}_{i}$ is corresponding with a voxel ${}^{I}\mathbf{m}_{i}$ is defined as

$$c \le \frac{f({}^{I}\mathbf{f}_{i}|{}^{I}\mathbf{m}_{j})}{\bar{f}({}^{I}\mathbf{f}_{i}|{}^{I}\mathbf{m}_{j})}$$
(14)

where $c \in [0, 1]$ is the acceptance-rejection threshold.

According to the MLE problem (10) and the LiDAR matching outlier criterion (14), the LiDAR matching cost is constructed based on the following twofold: 1) The likelihood of correct matchings, 2) the number of matching outliers. As noted in our previous work [21], the negative log-likelihood function $-\log(L({^I}\mathbf{f}_i|{^I}\mathbf{m}_j))$ can be simplified as point-to-voxel distance d_{p2v} using such a Gaussian approximation and principal component analysis (PCA). The LiDAR matching cost can be noted as

$$\mathbf{r}_{L}(\mathbf{x}, \boldsymbol{\tau}) = \begin{cases} d_{p2v} &, \frac{f({}^{I}\mathbf{f}|{}^{I}\mathbf{m})}{f({}^{I}\mathbf{f}|{}^{T}\mathbf{m})} \ge c \\ K_{m}n_{m} &, otherwise \end{cases}$$
(15)

where n_m is the number of incorrect matchings, and K_m is the weighted value that trade-off the importance of the likelihood of correct matchings and the number of incorrect matchings. The LiDAR matching cost $\mathbf{r}_L(\mathbf{x}, \tau)$ defined in (15) can be expressed as a function of \mathbf{x} and τ due to the ${}^I\mathbf{f}_i$ in the likelihood function $L({}^I\mathbf{f}_i | {}^I\mathbf{m}_j)$ can be expressed as ${}^I\mathbf{f}_i = \mathbf{R}^B\mathbf{f}_i + \mathbf{t}$, where \mathbf{R}_{t_k} and \mathbf{t}_{t_k} are a part of state vector $\mathbf{x}_{t_{k-1}}$. The state vector \mathbf{x}_{t_k} can be expressed as a function of $\mathbf{x}_{t_{k-1}}$ and $\mathbf{u}_{t_{k-1}}$ through the kinematic model (5), and $\mathbf{u}_{t_{k-1}}$ is the mean vector of $\tau_{t_{k-1}}$.

The point-to-voxel distance d_{p2v} in (15) can be split into two cases:

 For edge features, the point-to-voxel distance between edge feature and edge voxel is defined as:

$$d_{p2v}^{e} = \sum_{i=1}^{n_{c}^{e}} ({}^{I}\mathbf{f}_{i}^{e} - \mathbf{m}_{j}^{\mu_{e}})^{\top} (\mathbf{I} - \mathbf{m}_{j}^{n_{e}}\mathbf{m}_{j}^{n_{e}}^{\top}) ({}^{I}\mathbf{f}_{i}^{e} - \mathbf{m}_{j}^{\mu_{e}})$$
(16)

where n_c^e is the number of correct edge point-to-voxel matchings.

2) For planar features, the point-to-voxel distance between planar feature and planar voxel is defined as:

$$d_{p2v}^{p} = \sum_{i=1}^{n_{c}^{\nu}} ({}^{I}\mathbf{f}_{i}^{p} - \mathbf{m}_{j}^{\mu_{p}})^{\top} \mathbf{m}_{j}^{n_{p}} \mathbf{m}_{j}^{n_{p}}^{\top} ({}^{I}\mathbf{f}_{i}^{p} - \mathbf{m}_{j}^{\mu_{p}})$$
(17)

where n_c^p is the number of correct planer point-to-voxel matchings.

Remark 1: Different from traditional LiDAR-based localization and mapping systems [11], [14], [18], the proposed LiDAR matching cost (15) considers the likelihood of correct matchings and the number of incorrect matchings simultaneously. For the novel LiDAR matching cost defined in (15), it penalizes states that produce incorrect matchings which could otherwise help improve the quality of localization and mapping.

B. Gradient-free Optimization

Based on the above derivation, the problem solved in the present work can be summarized as:

$$\hat{\mathbf{u}} = \operatorname*{arg\,min}_{\mathbf{u}} \mathbb{E}_{\mathcal{Q}} \left[\mathbf{r}_{I}(\mathbf{x}, \boldsymbol{\tau}) + \mathbf{r}_{L}(\mathbf{x}, \boldsymbol{\tau}) \right]$$
(18)

The cost function defined in (18), constitutes the IMU measurements residual $\mathbf{r}_I(\mathbf{x}, \boldsymbol{\tau})$ and the LiDAR matching residual $\mathbf{r}_L(\mathbf{x}, \boldsymbol{\tau})$, composed of (9)-(17). The stochastic optimization (18) is difficult to solve using traditional gradient-based methods, due to the nonlinear and noncontinuous LiDAR matching residual, which considers the likelihood of correct LiDAR matchings and the number of incorrect LiDAR matchings in a piecewise function formulation. In this paper, a samplingbased and gradient-free optimization is adopted to solve the stochastic optimization problem (18), which is fully parallel and suitable for performing on GPU.

Assume that Q^* is the optimal distribution of τ , which makes $\mathbb{E}_{Q^*}[\mathbf{S}(\mathbf{x},\tau)]$ provides the lower bound of $\mathbb{E}_Q[\mathbf{S}(\mathbf{x},\tau)]$.

$$\mathbb{E}_{\mathcal{Q}^*}\left[\mathbf{S}(\mathbf{x}, \boldsymbol{\tau})\right] \le \mathbb{E}_{\mathcal{Q}}\left[\mathbf{S}(\mathbf{x}, \boldsymbol{\tau})\right] \tag{19}$$

IEEE TRANSACTIONS ON INDUSTRIAL ELECTRONICS

Hence, the stochastic optimization problem (18) can be converted into:

$$\hat{\mathbf{u}} = \operatorname*{arg\,min}_{\mathbf{u}} \mathbb{E}_{\mathcal{Q}} \left[\mathbf{S}(\mathbf{x}, \boldsymbol{\tau}) \right]$$
(20)

$$\triangleq \mathbb{E}_{\mathcal{Q}^*}(\boldsymbol{\tau}) = \int q^*(\mathbf{u}) \mathbf{u} d\mathbf{u}$$
(21)

where q^* is the density function correspondence to the optimal distribution Q^* . The step from (20) to (21) follows because $\mathbb{E}_{Q}[\mathbf{S}(\mathbf{x}, \tau)]$ achieves the lower bound by sampling from the optimal distribution Q^* . Hence, if the optimal distribution Q^* can be defined, the stochastic optimization problem (18) can be solved through calculate the expected value of τ sampled from the optimal distribution Q^* .

1) Optimal distribution: The definition of the optimal density function q^* is crucial to solving (21), which can be derived from the lower bound of $\mathbb{E}_{Q^*}[\mathbf{S}(\mathbf{x}, \tau)]$. As noted in [24], the lower bound of $\mathbb{E}_{Q^*}[\mathbf{S}(\mathbf{x}, \tau)]$ can be defined as follows

$$\mathbb{E}_{\mathcal{Q}^{*}}\left(\mathbf{S}\left(\mathbf{x},\boldsymbol{\tau}\right)\right) \geq -\lambda \log \left(\mathbb{E}_{\mathcal{Q}}\left[\exp\left(-\frac{1}{\lambda}\mathbf{S}\left(\mathbf{x},\boldsymbol{\tau}\right)\right)\right]\right) - \lambda \mathbb{D}(\mathcal{Q}^{*}||\mathcal{Q})$$
(22)

which establishes a relationship between the lower bound of $\mathbb{E}_{\mathcal{Q}^*}[\mathbf{S}(\mathbf{x}, \boldsymbol{\tau})]$ and the distribution $\mathcal{Q}, \lambda \in \mathbb{R}^+$ is a value similar to the kernel bandwidth of the Gaussian kernel function [25], and $\mathbb{D}(\mathcal{Q}^*||\mathcal{Q})$ is the KL-divergence between distributions \mathcal{Q}^* and \mathcal{Q} , which can be expanded as:

$$\mathbb{D}(\mathcal{Q}^*||\mathcal{Q}) = \mathbb{E}_{\mathcal{Q}^*}\left[\log\left(\frac{q^*}{q}\right)\right]$$
(23)

where q^* and q are density functions correspondence to distributions Q^* and Q, respectively.

Substituting (23) into (22) yields

. . . .

$$\lambda \mathbb{E}_{Q^{*}} \left[\log \left(\frac{q^{*}}{q} \right) \right] \geq -\lambda \log \left(\mathbb{E}_{Q} \left[\exp \left(-\frac{1}{\lambda} \mathbf{S} \left(\mathbf{x}, \tau \right) \right) \right] \right) - \mathbb{E}_{Q^{*}} \left[\mathbf{S} \left(\mathbf{x}, \tau \right) \right]$$
(24)

which leads to

$$\lambda \mathbb{E}_{\mathcal{Q}^{*}} \left[\log \left(q^{*} \right) \right] - \lambda \mathbb{E}_{\mathcal{Q}^{*}} \left[\log \left(q \right) \right] \geq \\ - \lambda \log \left(\mathbb{E}_{\mathcal{Q}} \left[\exp \left(-\frac{1}{\lambda} \mathbf{S} \left(\mathbf{x}, \boldsymbol{\tau} \right) \right) \right] \right) - \mathbb{E}_{\mathcal{Q}^{*}} \left[\mathbf{S} \left(\mathbf{x}, \boldsymbol{\tau} \right) \right]$$
(25)

With the aim of finding the optimal density function q^* , (25) can be rewritten as:

$$\mathbb{E}_{\mathcal{Q}^{*}}\left[\log\left(q^{*}\right)\right] \geq \mathbb{E}_{\mathcal{Q}^{*}}\left[-\log\left(\mathbb{E}_{\mathcal{Q}}\left[\exp\left(-\frac{1}{\lambda}\mathbf{S}\left(\mathbf{x},\boldsymbol{\tau}\right)\right)\right]\right)\right] \\ +\mathbb{E}_{\mathcal{Q}^{*}}\left[\log\left(\exp\left(-\frac{1}{\lambda}\mathbf{S}\left(\mathbf{x},\boldsymbol{\tau}\right)\right)\right) + \log\left(q\right)\right]$$
(26)

which can be simplified as

$$\mathbb{E}_{\mathcal{Q}^{*}}\left[\log\left(q^{*}\right)\right] \geq \mathbb{E}_{\mathcal{Q}^{*}}\left[\log\left(\frac{\exp\left(-\frac{1}{\lambda}\mathbf{S}\left(\mathbf{x},\boldsymbol{\tau}\right)\right)q}{\mathbb{E}_{\mathcal{Q}}\left[\exp\left(-\frac{1}{\lambda}\mathbf{S}\left(\mathbf{x},\boldsymbol{\tau}\right)\right)\right]}\right)\right] \quad (27)$$

Hence, the optimal density function q^* can be defined as

$$q^{*} = \frac{\exp\left(-\frac{1}{\lambda}\mathbf{S}\left(\mathbf{x},\boldsymbol{\tau}\right)\right)q}{\mathbb{E}_{\mathcal{Q}}\left(\exp\left(-\frac{1}{\lambda}\mathbf{S}\left(\mathbf{x},\boldsymbol{\tau}\right)\right)\right)}$$
(28)

According to the derivation in (23)-(27), the optimal density function defined in (28) can make $\mathbb{E}_{\mathcal{Q}^*}[\mathbf{S}(\mathbf{x}, \tau)]$ achieve

Algorithm 1: Gradient-free LiDAR-inertial odometry

Input : ^BF: Newly obtained feature set, ^IM: The voxel map, m: Number of samples, u₀: The initial guess of u.

Output: $\hat{\mathbf{x}}$: The estimated state vector.

for
$$i \leftarrow 0$$
 to $m - 1$ do
 $\mathbf{u}_i \leftarrow Sample \mathcal{Q}(\mathbf{u}_0, \sigma_u);$
Get \mathbf{T}_i from kinematic model (5) and $\mathbf{u}_i;$
Compute the IMU measurment residual $\mathbf{r}_I(\mathbf{x}, \mathbf{u}_i);$
for each ${}^B\mathbf{f}_j \in {}^B \mathbf{F}$ do
Transform ${}^B\mathbf{f}_j$ into the inertial frame ${}^I\mathbf{f}_j = \mathbf{T}_i^B\mathbf{f}_j$
Find ${}^I\mathbf{m}_k = \text{FMM}({}^I\mathbf{f}_j, {}^I\mathbf{M});$
if $IsCorrectMatching({}^I\mathbf{f}_j, {}^I\mathbf{m}_k)$ is true then
 $\mathbf{f}_i f_j \in \mathbf{F}^e$ then
 $\mathbf{r}_L(\mathbf{x}, \mathbf{u}_i) = \mathbf{r}_L(\mathbf{x}, \mathbf{u}_i) +$
 $\left[{}^{(I}\mathbf{f}_j - \mathbf{m}_k^{\mu_e})^\top (\mathbf{I} - \mathbf{m}_k^{n_e} \mathbf{m}_k^{n_e\top})({}^I\mathbf{f}_j - \mathbf{m}_k^{\mu_e}) \right]$
else if $\mathbf{f}_j \in \mathbf{F}^p$ then
 $\mathbf{r}_L(\mathbf{x}, \mathbf{u}_i) = \mathbf{r}_L(\mathbf{x}, \mathbf{u}_i) +$
 $\left[{}^{(I}\mathbf{f}_j - \mathbf{m}_k^{\mu_p})^\top \mathbf{m}_k^{n_p} \mathbf{m}_k^{n_p\top} ({}^I\mathbf{f}_j - \mathbf{m}_k^{\mu_p}) \right]$
else
 $\sum_{i=0}^{m-1} \mathbf{r}_I(\mathbf{x}, \mathbf{u}_i) + \mathbf{r}_L(\mathbf{x}, \mathbf{u}_i)$
 $\hat{\mathbf{u}} = \sum_{i=0}^{m-1} \frac{\exp(-\frac{1}{\lambda}\mathbf{S}(\mathbf{x}, \mathbf{u}_i))\mathbf{u}_i}{\sum_{j=0}^{m-1} \exp(-\frac{1}{\lambda}\mathbf{S}(\mathbf{x}, \mathbf{u}_j))}$
Get $\hat{\mathbf{x}}$ from kinematic model (5) and $\hat{\mathbf{u}};$

the lower bound in (22), which means that the stochastic optimization problem (18) can be solved by substituting (28) into (21).

2) Importance Sampling: With the definition of the optimal distribution, (21) can be solved through the importance sampling technique [26], which compute expectations with respect to Q^* by sampling from Q.

$$\hat{\mathbf{u}} = \int \underbrace{\frac{q^*(\boldsymbol{\tau})}{q(\boldsymbol{\tau})}}_{w(\boldsymbol{\tau})} q(\boldsymbol{\tau}) \boldsymbol{\tau} d\boldsymbol{\tau} = \mathbb{E}_{\mathcal{Q}} \left[w(\boldsymbol{\tau}) \boldsymbol{\tau} \right]$$
(29)

where $w(\boldsymbol{\tau})$ is the weight of importance sampling.

In practice, the expectation $\mathbb{E}_{\mathcal{Q}}[w(\tau)\tau]$ is difficult to perfectly evaluate. The optimal solution defined in (29) is approximated through a Monte-Carlo sampling.

$$\hat{\mathbf{u}} = \sum_{i=0}^{m-1} \frac{\exp\left(-\frac{1}{\lambda}\mathbf{S}\left(\mathbf{x},\mathbf{u}_{i}\right)\right)\mathbf{u}_{i}}{\sum_{j=0}^{m-1}\exp\left(-\frac{1}{\lambda}\mathbf{S}\left(\mathbf{x},\mathbf{u}_{j}\right)\right)}$$
(30)

where *m* is the number of Monte-Carlo sampling, and \mathbf{u}_i is a vector obtained from *i*-th sampling on the Gaussian distribution $\mathcal{Q}(\mathbf{u}, \boldsymbol{\sigma}_u)$. The mean vector **u** is calculated by substituting raw IMU measurements into (8), and the covariance $\boldsymbol{\sigma}_u$ is a predefined matrix based on the IMU measurement noise. $\hat{\mathbf{u}}$ is the estimated vector of the angular velocity $\boldsymbol{\omega}$ and local linear acceleration **a**. The state vector can be estimated by substituting (30) into the kinematic model defined in (5). The process of the proposed gradient-free LiDAR-inertial odometry is given in Algorithm 1.

IEEE TRANSACTIONS ON INDUSTRIAL ELECTRONICS

Remark 2: Different from traditional LiDAR-based localization and mapping systems [11], [14], [18], the LiDARinertial odometry described in Algorithm 1 is fully paralleled, which can be performed efficiently with a GPU. In [11], [14], [18], the localization and mapping problem is solved through gradient optimization or IEKF, which is time-consuming due to the LiDAR matching process for each iteration. For the proposed method, a Monte-Carlo sampling is adopted to solve the optimization with a single parallel iteration, which significantly improves the real-time performance of the LiDARinertial odometry. Moreover, the computation of Jacobians can be problematic for complex systems, the gradient-free optimization introduced in this paper does not require to deriving the Jacobians, which makes the algorithm flexible to deal with noncontinuous constraints.

C. Practical Implementation

The proposed LiDAR-inertial odometry build with the Robots Operating System (ROS) under Ubuntu. The feature extraction and voxel mapping process are implemented on the CPU, and the gradient-free optimization process described in Algorithm 1 is implemented on a GPU with Nvidia's CUDA architecture. It is worth noting that, there exists a nested loop in Algorithm 1. In order to deploy Algorithm 1 on GPU parallelly, a dynamic-parallelism-like strategy [27] is adopted. In our implementation, Algorithm 1 uses $m \times k$ CUDA threads, where m is the number of Monte-Carlo sampling and k is the number of features. For example, the *l*-th CUDA thread, where $0 \leq l < m \times k$, is in charge of calculating LiDAR measurements residual corresponding to mod(l, k)-th feature and l/k-th Monte-Carlo sampling. All the residuals calculated in 0 to $m \times k - 1$ CUDA threads are saved in a matrix with a dimension of $m \times k$. The cost function $\mathbf{S}(\mathbf{x}, \mathbf{u}_i)$, $i = 0, \ldots, m - 1$, is calculated by summing all the column vectors of the matrix through a reduction sum operation in CUDA Thrust [28].

Predefined parameters of Algorithm 1 are set to c = 0.98, and $\sigma_u = \text{diag}(3.0, 3.0, 3.0, 0.1, 0.1, 0.1)$ for all experiments evaluated in Section V. As noted in [24], the negative exponentiation required by (30) is numerically sensitive to the range of the input values. For this reason, we shift the range of the costs **S** so that the best cost sampled has a value of 0. Define \mathbf{S}_{min} as the minimum cost of $\mathbf{S}(\mathbf{x}, \mathbf{u}_i), i = 0, \dots, m - 1$, where *m* is the number of Monte-Carlo sampling. Multiply $\exp(\frac{1}{\lambda}\mathbf{S}_{\min})/\exp(\frac{1}{\lambda}\mathbf{S}_{\min})$ to (30) results in

$$\hat{\mathbf{u}} = \sum_{i=0}^{m-1} \frac{\exp\left(-\frac{1}{\lambda} \left(\mathbf{S}\left(\mathbf{x}, \mathbf{u}_{i}\right) - \mathbf{S}_{\min}\right)\right) \mathbf{u}_{i}}{\sum_{j=0}^{m-1} \exp\left(-\frac{1}{\lambda} \left(\mathbf{S}\left(\mathbf{x}, \mathbf{u}_{j}\right) - \mathbf{S}_{\min}\right)\right)}$$
(31)

which can prevent numerical overflow or underflow. The $\exp\left(-\frac{1}{\lambda}\left(\mathbf{S}\left(\mathbf{x},\mathbf{u}_{i}\right)-\mathbf{S}_{\min}\right)\right)$ in (31) is similar to the Gaussian kernel function formulation [25], where the parameter λ is corresponding to the kernel bandwidth. As noted in [25], the spatial extent of the Gaussian kernel ranges from $-\infty$ to $+\infty$, but in practice, it has negligible values for $(\mathbf{S}(\mathbf{x},\mathbf{u}_{i})-\mathbf{S}_{\min})$ larger than a few (say 5) λ . Hence, if $\lambda \ll (\mathbf{S}(\mathbf{x},\mathbf{u}_{i})-\mathbf{S}_{\min})$

the solution of (31) places all its mass on a single Monte-Carlo sampling, whereas as $\lambda \gg (\mathbf{S}(\mathbf{x}, \mathbf{u}_i) - \mathbf{S}_{\min})$ all the Monte-Carlo sampling have close equal weight. To prevent the numerical instability of (31), the parameter λ is set as a self-adapting value according to the average cost \mathbf{S}_{mean} and the minimum cost \mathbf{S}_{\min} . We set $\lambda = (\mathbf{S}_{\text{mean}} - \mathbf{S}_{\min})/5$ for all experiments evaluated in Section V.

V. RESULTS

In this section, the effectiveness of the proposed LiDARinertial odometry is demonstrated through various scenarios, including public datasets and real-world experiments. For more details, a video demonstration is available online¹.

A. Quantitative Evaluation on the NTU VIRAL Datasets

The proposed LiDAR-inertial odometry is evaluated quantitatively on the NTU VIRAL dataset [29] which provides point clouds captured by two 16-channel lightweight LiDAR OS1 (horizontal LiDAR measurements are used in experiments), high-frequency inertial data, and ground truth from the Leica laser tracker. Different from datasets with 3D LiDAR mounted on ground vehicles, the NTU VIRAL dataset is aerial vehicle viewpoint, which has much more complex motion in 3D space with frequent aggressive rotational and translational motions. We compare our work with current state-of-the-art LiDAR-based odometry such as LOAM² [11], LIO-SAM³ [14], and FAST-LIO2⁴ [18]. For a fair comparison, all methods are implemented without loop closure. Moreover, an ablation study on the proposed method is performed to understand the influence of the sampling number, we run the algorithm in various Monte-Carlo sampling numbers of 512, 1024, 2048, so-called proposed 512, proposed 1024, proposed 2048, respectively. Each experiment is conducted on a computer equipped with an Intel Core i7-7700 and an NVIDIA GeForce GTX 1080.

1) Accuracy evaluation: To quantitatively evaluate the accuracy of the LiDAR-inertial odometry, the absolute trajectory error (ATE) results of each method over NTU VIRAL datasets are shown in Table I. An ablation study on the parameter K_m is also performed to understand the effect of incorrect LiDAR matchings considered in (15). For $K_m = 0$, which means only correct LiDAR matchings are considered in the LiDAR matching cost, the proposed method converges into a sub-optimal local minimum that leads to system failure. As an extreme example, assuming that there exists a transformation T that matches all the features to the incorrect voxels. Obviously, the transformation $\hat{\mathbf{T}}$ is a sub-optimal local minimum when $K_m = 0$. For $K_m > 0$, the novel LiDAR matching cost defined in (15) penalizes states that produce incorrect matchings which could otherwise help improve the quality of localization and mapping and attempts to escape the local minimum. From the results shown in Table I, for $K_m > 0$, better accuracy can be achieved when a higher

¹https://b23.tv/1bMhFY5

²https://github.com/brytsknguyen/A-LOAM

³https://github.com/brytsknguyen/LIO-SAM

⁴https://github.com/shenhm0516/FAST_LIO

IEEE TRANSACTIONS ON INDUSTRIAL ELECTRONICS

number of samples is adopted for the proposed method, and the proposed method with 2048 samples achieves the best accuracy in all data sequences. This is because the gradientfree optimization algorithm introduced in Section IV-B is a suboptimal estimator, which approximates the optimal solution through the Monte-Carlo sampling, and the approximation performance is determined by the number of samples. In Table I, we also compare the ATE of the proposed method at $K_m = 0.5$ and $K_m = 1.0$. In all sequences, the accuracy of the proposed method with $K_m = 0.5$ performs similarly to $K_m = 1.0$, and we choose $K_m = 0.5$ for all the rest of the experiments performed in this section. The comparison between the proposed method with 2048 samples and ground truth is shown in Fig. 2(a).

2) Running time evaluation: Table II shows the average running time for process one scan of the proposed method with different sampling numbers, and each state-of-the-art method. The time consumption of each individual component, including feature extraction ("FE" in Table II), state estimation ("SE" in Table II), and mapping ("Map" in Table II), are compared across all 9 sequences. For the feature extraction, FAST-LIO2 achieves the best real-time performance, due to FAST-LIO2 registering raw points to the map without extracting features. Thanks to the fully paralleled optimization algorithm introduced in Section IV-B, which can be accelerated through massive parallel sampling on a GPU, the proposed method with 512 samples achieves the best real-time performance in state estimation, and the proposed method with 1024 samples (state estimated by the scan-to-map matching) achieves a similar state estimation time with LOAM (state estimated by the scan-to-scan matching). For the mapping time, the proposed method maintains a voxel map in a hash table instead of a KD-tree used in LIO-SAM and LOAM or the iKDtree used in FAST-LIO2. It is worth noting that, similar to FAST-LIO2, the proposed method maintains the voxel map incrementally without re-build the local map in KD-Tree (the time complexity of re-building the KD-Tree is $O(n \log n)$ [17]). Furthermore, thanks to the effective data insertion operation of hash data structure, which takes a constant time complexity of O(1), features can insert into the voxel map more efficiently, when compared with the iKD-tree adopted in FAST-LIO2 (the time complexity of point insertion on iKDtree is $O(\log n)$ [18]). The total time for feature extraction, state estimation, and mapping ultimately affects the real-time performance of the localization and mapping system, which is also summarized in Table II ("Total" in Table II). As the results show in Table I and Table II, for the proposed method, the number of samples is a tradeoff between accuracy and realtime performance. The proposed method with 2048 samples outperforms each state-of-the-art method in accuracy with an acceptable processing time. Compared with optimizationbased methods, LOAM and LIO-SAM, the proposed method with 1024 samples achieves a comparative accuracy and a significant efficiency improvement, which saves 70% to 90%processing time, and the proposed method with 512 samples achieves the best real-time performance in all data sequences. Fig. 2(b) shows the frames per second (FPS) on nya 03 sequence. The result indicated that the proposed LiDAR-inertial



Fig. 2. Comparison results of under the nya 03 sequence.



Fig. 3. The aerial platform used in real-world experiments.

odometry reaches the speed of over 200 Hz and achieves 1.5 to 2 times FPS compared with FAST-LIO2.

B. Real-world Experiments

To attest the practicality of the proposed localization and mapping method, a variety of real-world experiments is performed with a quadrotor aerial platform. As shown in Fig. 3, the aerial platform is equipped with a Pixhawk4 flight controller, a Velodyne VLP-16 LiDAR, a MicroStrain 3DM-GX5-25 IMU, and a low-power onboard computer (Nvidia Xavier worked with the 30W 6 Core power model). The realworld experiments are divided into two parts: 1) Verification of real-time performance and global consistency in a large-scale urban environment. 2) Autonomous flight in a GPS-denied environment.

1) High precision map building in real-time: To evaluate the performance of the proposed LiDAR-inertial odometry, a MAV running the proposed method with 512 samples onboard is flown manually to reconstruct a dense 3D, high-precision, large-scale map of campus in real-time. The mapping result is shown in Fig. 4, which is merged with the Google Earth image to examine the accuracy of the proposed method. This experiment is convincing due to the fact that MAV has to travel through a bridge over an open lake, typically a challenging scene for the LiDAR-based localization and mapping system, and the result shows that the proposed method can close the loop when MAV returns to the start point without loop closure. The high-quality point cloud demonstrates that the proposed LiDAR-inertial odometry is able to provide high-precision reconstruction in a large-scale environment. To analyze the accuracy and real-time performance of the proposed method, we compare the proposed method with FAST-LIO2 [18], LIO-SAM [14], LOAM [11], CLINS [15], and LINS [13] on a low-power ARM onboard computer (Nvidia Xavier). From the

IEEE TRANSACTIONS ON INDUSTRIAL ELECTRONICS

TABLE I Absolute Trajectory Error (ATE, Meters) over NTU VIRAL datasets. The best result is highlighted in BLUE, the second best result is highlighted in RED, and the third best result is highlighted in BOLD.

Method		Proposed 51	2		Proposed 102	24		Proposed 204	48	FAST-LIO2	LIO-SAM	LOAM
	$K_m = 0$	$K_m = 0.5$	$K_m = 1.0$	$K_m = 0 K_m = 0.5$		$K_m = 1.0$	$K_m = 0$	$K_m = 0.5$	$K_m = 1.0$			
eee 01	_1	0.1429	0.1379	-	0.0996	0.0949	-	0.0944	0.0955	0.2096	0.1022	0.2309
eee 02	-	0.1192	0.1610	-	0.0865	0.0830	-	0.0840	0.0796	0.1649	0.1113	0.1963
eee 03	-	0.1415	0.1613	-	0.1132	0.1089	-	0.1067	0.1108	0.1945	0.1469	0.1730
nya 01	-	0.1277	0.1245	-	0.0962	0.1030	-	0.0937	0.0977	0.1217	0.0943	0.0973
nya 02	-	0.1425	0.1406	-	0.1138	0.1233	-	0.1071	0.1212	0.2424	0.1200	0.1165
nya 03	-	0.1239	0.1230	-	0.1098	0.1062	-	0.1025	0.1064	0.1349	0.1049	0.1029
sbs 01	-	0.1242	0.1237	-	0.1044	0.0998	-	0.0964	0.0965	0.1600	0.1072	0.1793
sbs 02	-	0.1270	0.1264	-	0.1075	0.1006	-	0.0977	0.0968	0.1831	0.0980	0.1175
sbs 03	-	0.1157	0.1167	-	0.0968	0.0962	-	0.0903	0.0953	0.1519	0.1034	0.3016

¹ - denotes that the system failure.

TABLE II

AVERAGE PROCESSING TIME (MILLISECONDS) FOR INDIVIDUAL COMPONENTS WHEN PROCESSING ONE SCAN. THE BEST RESULT IS HIGHLIGHTED IN **BLUE**, THE SECOND BEST RESULT IS HIGHLIGHTED IN **RED**, AND THE THIRD BEST RESULT IS HIGHLIGHTED IN **BOLD**.

Method	Proposed 512			P	Proposed 1024			Proposed 2048			FAST-LIO2			LIO-SAM			LOAM							
	Total	FE	SE	Map	Total	FE	SE	Map	Total	FE	SE	Map	Total	FE	SE	Map	Total	FE	SE	Map	Total	FE	SE	Map
eee 01	15.60	2.17	11.70	1.72	25.30	2.18	21.66	1.46	40.15	2.33	36.07	1.76	22.23	0	18.16	4.07	125.47	2.29	42.99	80.19	229.28	3.04	15.26	210.98
eee 02	15.89	3.22	10.96	1.72	23.26	2.31	19.42	1.53	38.08	2.54	33.76	1.78	27.94	0	23.26	4.68	102.69	2.96	40.79	58.95	193.61	3.30	14.93	175.37
eee 03	16.36	3.18	11.31	1.87	25.59	3.32	20.70	1.57	39.74	3.42	34.50	1.82	40.38	0	32.23	8.14	112.00	2.76	50.01	59.23	173.55	3.23	15.19	155.13
nya 01	12.26	3.06	8.08	1.12	18.05	3.34	13.48	1.24	30.56	3.30	25.60	1.66	26.86	0	23.63	3.23	75.61	3.22	28.81	43.58	126.81	3.23	15.15	108.43
nya 02	12.19	3.12	7.92	1.15	18.15	3.59	13.24	1.52	30.06	3.33	25.04	1.69	25.34	0	22.78	2.56	78.47	3.07	37.01	38.38	132.27	3.17	15.24	113.85
nya 03	12.74	3.19	8.01	1.15	17.28	3.21	12.79	1.27	30.56	3.37	25.49	1.69	18.85	0	16.97	1.88	80.16	3.13	30.95	46.08	136.76	3.34	15.94	117.48
sbs 01	13.27	2.41	9.36	1.50	21.50	2.84	17.15	1.51	33.46	2.39	29.29	1.78	28.64	0	24.41	4.23	102.98	2.56	41.10	59.23	192.29	2.56	12.94	176.78
sbs 02	13.22	2.48	9.28	1.47	20.50	2.88	16.09	1.53	33.17	2.60	28.76	1.81	20.89	0	17.89	3.01	99.62	2.67	34.81	62.19	193.54	2.61	12.89	178.04
sbs 03	13.07	2.49	9.12	1.47	21.06	2.98	16.58	1.50	33.56	2.49	29.28	1.80	32.15	0	27.74	4.40	96.41	2.62	33.30	60.48	208.85	2.43	13.02	193.40



Fig. 4. The mapping result of the proposed method.

result shown in Table III, thanks to the parallel acceleration of the low-power onboard GPU, the proposed method achieves the best real-time performance than other methods. Compared with optimization-based methods, the proposed method is 8 times faster than LIO-SAM, 12 times faster than LOAM, and 15 times faster than CLINS. Even compared with the state-ofthe-art filter-based method FAST-LIO2, the proposed method saves more than 24% of the total processing time and is 7 times faster than LINS. For accuracy, end-to-end errors are also reported in Table III. Both LIO-SAM, LOAM, CLINS, and LINS shows large drift because their total computation time greatly exceeded 100ms, which leads them to drop a lot of LiDAR scan to achieve real-time performance. When compared with FAST-LIO2, the state-of-the-art filter-based LiDAR odometry, the proposed method still shows better endto-end accuracy and real-time performance, thanks to the fully

paralleled optimization method introduced in Section IV-B. Fig. 5 reports the computation time for the individual components of state estimation (the method described in Algorithm 1) when performing the large-scale reconstruction. The FMM process dominates the real-time performance of the state estimation process, which is in charge of finding correspondence voxel for each feature. It should be noted that the FMM process can be quite computationally expensive since $512 \times k$ features should be conducted, where 512 is the number of Monte-Carlo sampling, and k is the number of features in one sampling. In this paper, the FMM process is split into $512 \times k$ threads and accelerated through Nvidia CUDA. Thanks to this strategy, we can cut down the mean FMM time to 34.33ms compared to several seconds when using a single CPU thread. While cost function construction process is accelerated through a reduction sum operation in CUDA Thrust [28] (details can be found in Section IV-C), has an average processing time of 5.72ms, and the optimal solution is approximated through an importance sampling technique, which has an average processing time of 0.42ms. Thank to the Gradient-free optimization method introduced in Section IV-B which can approximate optimal solution with a single parallel-iteration solver, the proposed tightly-coupled LiDAR odometry has highly real-time performance under lowpower GPU acceleration. The average running time of the proposed method for process one scan is 49.46ms, much less than 100ms, which demonstrates that the proposed LiDARinertial odometry can achieve real-time without dropping any

IEEE TRANSACTIONS ON INDUSTRIAL ELECTRONICS

 TABLE III

 END-TO-END ERROR (METERS) AND AVERAGE PROCESSING TIME

 (MILLISECONDS) FOR INDIVIDUAL COMPONENTS WHEN PROCESS ONE

 SCAN. THE BEST RESULT IS HIGHLIGHTED IN BOLD.

Method	Proposed	FAST-LIO2	LIO-SAM	LOAM	CLINS	LINS
Feature Extraction	6.85	0	11.78	24.43	13.55	15.79
State Estimation	40.47	58.59	244.92	51.41	741.51	63.22
Mapping	2.14	6.54	141.44	528.85	21.20	269.17
Total	49.46	65.13	398.12	604.69	776.26	348.18
End-to-End Error	0.12	0.17	5.33	19.02	14.12	30.52



Fig. 5. The computation time for state estiamtion.

scan (LiDAR rotation rate is 10 Hz) and is suitable to perform on low-power onboard computing power.

2) Application of MAV autonomous flight: To attest the practicality of the proposed localization and mapping method, two autonomous flights are performed, which are called aggressive flight test and fully autonomous flight test. During the aggressive flight test, the proposed method is used for feedback control of the MAV which tracks a high-speed trajectory. Poses estimated by the proposed method are shown in Fig. 6(a), in addition, the linear velocity and angular velocity are shown in Fig. 6(b). The maximum velocity reaches 9m/s, the maximum angular velocity achieves 180 deg/s (the square root of the triaxial angular rate achieves 245 deq/s) during the aggressive flight test. The results show that the proposed LiDAR-inertial odometry is suitable for MAV real-time control and trajectory tracking. For the fully autonomous flight test, a real-world autonomous flight is performed in a cluttered forest scenario. This test is illustrated due to only the onboard computing power is adopted to run the proposed localization and mapping algorithm with the additional workload, such as the motion planning algorithm [30], sensor drivers, in real-time. During



Fig. 6. Results of the aggresive flight test.



Fig. 7. Fully autonomous flight.

this flight test, MAV track the reference trajectories for achieving obstacles avoidance, and the localization and mapping results are illustrated in Fig. 7. The results demonstrated that the proposed LiDAR-inertial odometry has the ability to enable the fully autonomous flight without a prior map and external sensings.

VI. CONCLUSION

In this article, an efficient and high-accuracy localization and mapping algorithm is developed for MAV. The proposed method formulate the localization and mapping problem as a stochastic optimization problem, which fuses IMU measurements and LiDAR measurements in a tightly-coupled manner. A novel LiDAR matching cost is constructed by formulating the LiDAR matching problem as an MLE problem and simplifying it as a piecewise function through GMM and acceptance-rejection sampling, which consider the likelihood of correct matchings and the number of incorrect matchings simultaneously. Thanks to the gradient-free optimization, the nonlinear and noncontinuous stochastic optimization problem is solved by a fully paralleled Monte-Carlo sampling, which can provide real-time localization and mapping with a lowpower onboard computer. The proposed LiDAR-inertial odometry is validated with an extensive evaluation in both public datasets and real-world experiments. The results show that the proposed method produces real-time and accurate localization and mapping results, and has a great potential to enable many autonomous aerial robots applications.

REFERENCES

- L. Fusini, T. I. Fossen, and T. A. Johansen, "Nonlinear observers for gnss- and camera-aided inertial navigation of a fixed-wing uav," *IEEE Transactions on Control Systems Technology*, vol. 26, DOI 10.1109/TCST.2017.2735363, no. 5, pp. 1884–1891, 2018.
- [2] F. Nan, S. Sun, P. Foehn, and D. Scaramuzza, "Nonlinear mpc for quadrotor fault-tolerant control," *IEEE Robotics and Automation Letters*, DOI 10.1109/LRA.2022.3154033, pp. 1–1, 2022.
- [3] T.-M. Nguyen, M. Cao, S. Yuan, Y. Lyu, T. H. Nguyen, and L. Xie, "Viral-fusion: A visual-inertial-ranging-lidar sensor fusion approach," *IEEE Transactions on Robotics*, DOI 10.1109/TRO.2021.3094157, pp. 1–20, 2021.
- [4] K. Koide, J. Miura, and E. Menegatti, "A portable three-dimensional lidar-based system for long-term and wide-area people behavior measurement," *International Journal of Advanced Robotic Systems*, vol. 16, DOI 10.1177/1729881419841532, no. 2, 2019.
- [5] S. Umeyama, "Least-squares estimation of transformation parameters between two point patterns," *IEEE Computer Architecture Letters*, vol. 13, no. 04, pp. 376–380, 1991.
- [6] A. Segal, D. Haehnel, and S. Thrun, "Generalized-icp," in *Proceedings of Robotics: Science and Systems*, DOI 10.15607/RSS.2009.V.021, Seattle, USA, Jun. 2009.

© 2022 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information. Authorized licensed use limited to: Shanghai Jiaotong University. Downloaded on February 06,2023 at 09:42:22 UTC from IEEE Xplore. Restrictions apply.

IEEE TRANSACTIONS ON INDUSTRIAL ELECTRONICS

- [7] P. Biber and W. Straßer, "The normal distributions transform: A new approach to laser scan matching," in *Proceedings 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2003)(Cat. No. 03CH37453)*, vol. 3, pp. 2743–2748. IEEE, 2003.
- [8] M. Yokozuka, K. Koide, S. Oishi, and A. Banno, "Litamin: Lidarbased tracking and mapping by stabilized icp for geometry approximation with normal distributions," in 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), DOI 10.1109/IROS45743.2020.9341341, pp. 5143–5150, 2020.
- [9] M. Yokozuka, K. Koide, S. Oishi, and A. Banno, "Litamin2: Ultra light lidar-based slam using geometric approximation applied with kl-divergence," in 2021 IEEE International Conference on Robotics and Automation (ICRA), DOI 10.1109/ICRA48506.2021.9560947, pp. 11619–11625, 2021.
- [10] K. Chen, B. T. Lopez, A.-a. Agha-mohammadi, and A. Mehta, "Direct lidar odometry: Fast localization with dense point clouds," *IEEE Robotics and Automation Letters*, vol. 7, DOI 10.1109/LRA.2022.3142739, no. 2, pp. 2000–2007, 2022.
- [11] J. Zhang and S. Singh, "Loam: Lidar odometry and mapping in real-time," in *Proceedings of Robotics: Science and Systems*, DOI 10.15607/RSS.2014.X.007, Berkeley, USA, Jul. 2014.
- [12] H. Ye, Y. Chen, and M. Liu, "Tightly coupled 3d lidar inertial odometry and mapping," in 2019 International Conference on Robotics and Automation (ICRA), pp. 3144–3150. IEEE, 2019.
- [13] C. Qin, H. Ye, C. E. Pranata, J. Han, S. Zhang, and M. Liu, "Lins: A lidar-inertial state estimator for robust and efficient navigation," in 2020 *IEEE International Conference on Robotics and Automation (ICRA)*, DOI 10.1109/ICRA40945.2020.9197567, pp. 8899–8906, 2020.
- [14] T. Shan, B. Englot, D. Meyers, W. Wang, C. Ratti, and D. Rus, "Lio-sam: Tightly-coupled lidar inertial odometry via smoothing and mapping," in 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 5135–5142. IEEE, 2020.
- [15] J. Lv, K. Hu, J. Xu, Y. Liu, X. Ma, and X. Zuo, "Clins: Continuoustime trajectory estimation for lidar-inertial system," in 2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), DOI 10.1109/IROS51168.2021.9636676, pp. 6657–6663, 2021.
- [16] W. Xu and F. Zhang, "Fast-lio: A fast, robust lidar-inertial odometry package by tightly-coupled iterated kalman filter," *IEEE Robotics and Automation Letters*, vol. 6, DOI 10.1109/LRA.2021.3064227, no. 2, pp. 3317–3324, 2021.
- [17] Y. Cai, W. Xu, and F. Zhang, "ikd-tree: An incremental kd tree for robotic applications," arXiv preprint arXiv:2102.10808, 2021.
- [18] W. Xu, Y. Cai, D. He, J. Lin, and F. Zhang, "Fast-lio2: Fast direct lidar-inertial odometry," *IEEE Transactions on Robotics*, DOI 10.1109/TRO.2022.3141876, pp. 1–21, 2022.
- [19] Z. Liu and F. Zhang, "Balm: Bundle adjustment for lidar mapping," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 3184–3191, 2021.
- [20] B. D. Flury, "Acceptance-rejection sampling made easy," SIAM Review, vol. 32, no. 3, pp. 474–476, 1990.
- [21] H. Shen, Q. Zong, B. Tian, and H. Lu, "Voxel-based localization and mapping for multi-robot system in gps-denied environments," *IEEE Transactions on Industrial Electronics*, DOI 10.1109/TIE.2022.3153822, 2022.
- [22] C. Forster, L. Carlone, F. Dellaert, and D. Scaramuzza, "Imu preintegration on manifold for efficient visual-inertial maximum-a-posteriori estimation," in *Proceedings of Robotics: Science and Systems*, DOI 10.15607/RSS.2015.XI.006, Rome, Italy, Jul. 2015.
- [23] T. Stoyanov, M. Magnusson, H. Andreasson, and A. J. Lilienthal, "Fast and accurate scan registration through minimization of the distance between compact 3d ndt representations," *The International Journal of Robotics Research*, vol. 31, no. 12, pp. 1377–1393, 2012.
- [24] G. Williams, P. Drews, B. Goldfain, J. M. Rehg, and E. A. Theodorou, "Information-theoretic model predictive control: Theory and applications to autonomous driving," *IEEE Transactions on Robotics*, vol. 34, DOI 10.1109/TRO.2018.2865891, no. 6, pp. 1603–1622, 2018.
- [25] B. M. ter Haar Romeny, "The gaussian kernel," Front-End Vision and Multi-Scale Image Analysis: Multi-Scale Computer Vision Theory and Applications, written in Mathematics, pp. 37–51, 2003.
- [26] A. Doucet, N. De Freitas, N. J. Gordon et al., Sequential Monte Carlo methods in practice, vol. 1, no. 2. Springer, 2001.
- [27] S. Jones, "Introduction to dynamic parallelism," in GPU Technology Conference Presentation S, vol. 338, p. 2012, 2012.
- [28] N. Bell and J. Hoberock, "Thrust: A productivity-oriented library for cuda," in *GPU computing gems Jade edition*, pp. 359–371. Elsevier, 2012.

- [29] T.-M. Nguyen, S. Yuan, M. Cao, Y. Lyu, T. H. Nguyen, and L. Xie, "Ntu viral: A visual-inertial-ranging-lidar dataset, from an aerial vehicle viewpoint," *The International Journal of Robotics Research*, DOI 10.1177/02783649211052312, 2022.
- [30] H. Lu, Q. Zong, S. Lai, B. Tian, and L. Xie, "Real-time perceptionlimited motion planning using sampling-based mpc," *IEEE Transactions* on *Industrial Electronics*, DOI 10.1109/TIE.2022.3140533, 2022.





Hongming Shen received his B.S. degree in flight vehicle design and engineering from Central North University, Taiyuan, China, in 2015, and the M.S. degree in aerospace transportation and control from Beijing Institute of Technology, Beijing, China, in 2017. He is currently working toward the Ph.D. degree in control theory and control engineering with Tianjin University, Tianjin, China. His current research interests include state estimation, multi-sensor fusion, localization and mapping, and aerial Robotics.

Qun Zong received the B.S., M.S., and Ph.D. degrees in automatic control from Tianjin University, Tianjin, China, in 1983, 1988, and 2002, respectively.

He is currently a Professor and one of the academic pacesetters in control theory and control engineering with the School of Electrical and Information Engineering, Tianjin University. He is also the Director of the New Aircraft Guidance and Control Center of the Ministry of Education, the Expert Groups Deputy Head of the Major

Project of the Chinese Ministry of Education, Beijing, China, and the Deputy Director of the New Aircraft Joint Research Center, Tianjin. His current research interests include guidance, control and simulation for flight vehicles, coordination control of multiagent systems, fault diagnosis and fault-tolerant control, and optimization control.

Dr. Zong is also a Committee Member of the Chinese Association of the Automation Control Theory Technical Committee, the Guidance, Navigation and Control Technical Committee of Chinese Society of Aeronautics and Astronautics (CSAA), and so on. He also serves as an Editorial Board Member for several scientific journals, including *Journal* of Control Theory and Applications and *Journal of Astronautics*.



Bailing Tian received the B.S., M.S. and Ph.D degrees in automatic control from Tianjin University, Tianjin, China, in 2006, 2008, and 2011, respectively. He was an academic visitor at the School of Electrical and Electronic Engineering, University of Manchester from June 2014 to June 2015. He is currently a professor at the School of Electrical and Information Engineering, Tianjin University. His main research interests include finite time control, motion planning, simultaneous localization and mapping, and incontrol for unmanned systems.

tegrated guidance and control for unmanned systems.





Hanchen Lu received the B.S. degree in automatic control from Central South University, Changsha, China, in 2015. He is currently working toward the Ph.D. degree in motion planning with School of Electrical and Information Engineering, Tianjin University, Tianjin, China. From September 2019 to September 2020, he was a visiting student at the School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore. His main research interests include motion planning and control of

micro-aerial vehicles.