Quantized LIO for acclerated logistics and inspection applications

Boyang Lou

Abstract— **LiDAR-Inertial Odometry (LIO) is widely used for autonomous navigation, but its deployment on Size, Weight, and Power (SWaP)-constrained platforms remains challenging due to the computational cost of processing dense point clouds. Conventional LIO frameworks rely on a single onboard processor, leading to computational bottlenecks and high memory demands, making real-time execution difficult on embedded systems. To address this, we propose QLIO, a multi-processor distributed quantized LIO framework that reduces computational load and bandwidth consumption while maintaining localization accuracy. QLIO introduces a quantized state estimation pipeline, where a co-processor pre-processes LiDAR measurements, compressing point-to-plane residuals before transmitting only essential features to the host processor. Additionally, an rQ-vector-based adaptive resampling strategy intelligently selects and compresses key observations, further reducing computational redundancy. Real-world evaluations demonstrate that QLIO achieves a 14.1× reduction in per-observation residual data while preserving localization accuracy. Furthermore, we release an open-source implementation to facilitate further research and real-world deployment. These results establish QLIO as an efficient and scalable solution for real-time autonomous systems operating under computational and bandwidth constraints.**

Supplementary Material

The video, code, and supplementary materials will be available at: https://github.com/luobodan/QLIO

I. Introduction

The rise of the low-altitude economy [1] and humanoid robotics [2]–[4] is accelerating demand for efficient, accurate [5], [6], and depth-aware 3D navigation [7]. As these systems expand into general outdoor dynamic environments [8], real-time [9] localization and mapping must be adapted to meet increasing constraints on computational efficiency [10]–[13], power consumption, and deployment feasibility.

LiDAR-Inertial Odometry (LIO) and SLAM have been widely explored for robotic perception and navigation, with numerous solutions [10]–[12], [14]–[23] improving mapping accuracy [24]–[28] and localization robustness [29]–[31]. However, existing methods remain **heavily biased toward** algorithmic performance metrics [32]–[35] such as Absolute Pose Error (APE) and Relative Pose Error (RPE), often neglecting feasibility for lightweight, real-time applications [36], [37]. While some approaches improve runtime efficiency to operate on Size, Weight, and Power (SWaP)-constrained platforms [10]–[12], [38], they often sacrifice robustness.

While integrating visual perception [25], [38], [42]–[44] improves state estimation, it increases computational complexity. The cloud-based LiDAR localization [45]–[47] offloads processing but suffers from latency and network constraints [48], [49]. Efficient compression schemes [50]–[54] help address bandwidth limitations, yet balancing compression efficiency and data fidelity remains challenging. Meanwhile, advances in SoC architectures [55] and distributed processing frameworks [56] are shifting the research focus to edge computing, optimizing real-time compression, data transmission, and localized processing under strict SWaP constraints.

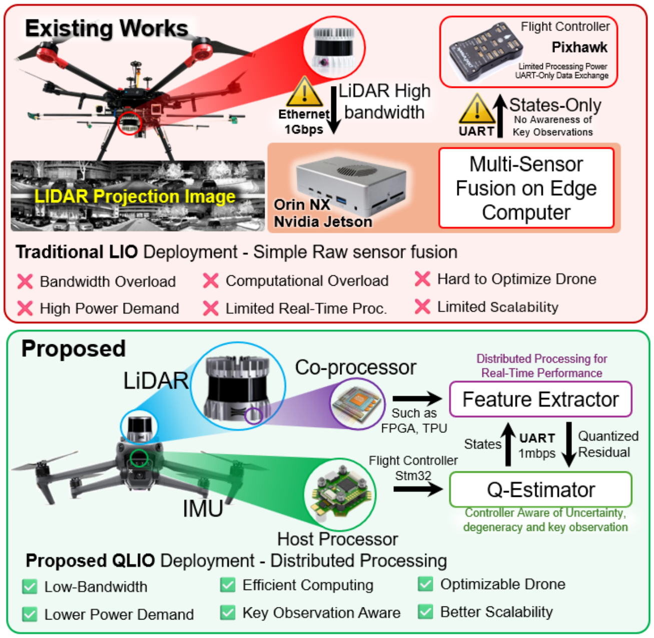

Fig. 1. Comparison of Traditional LIO deployment [39]–[41] vs. Proposed QLIO: Enhancing Efficiency with Distributed Processing.

SoC and distributed processing [55]–[57] have proven effective for 2D visual pipelines. However, deploying such a solution to 3D LIO in real-world systems remains challenging due to the trade-off between SWaP and robustness. Robust algorithms require substantial processing power [58], yet practical platforms, such as UAVs and humanoids, must remain lightweight and efficient. This trade-off limits scalability, demanding a rethink of LIO architectures for adaptive processing and efficiency.

To bridge this gap, we propose QLIO, a multi-processor distributed quantized LIO framework tailored for edge devices. Unlike prior work such as Quantized Visual-Inertial Odometry (QVIO) [56], which applies quantization-based estimation to visual features [59], [60], QLIO introduces LiDAR-specific optimizations that enhance efficiency while maintaining localization accuracy. Our method employs structured point-to-plane residual compression and rQ-vector-based adaptive resampling to reduce computational load and bandwidth consumption without compromising

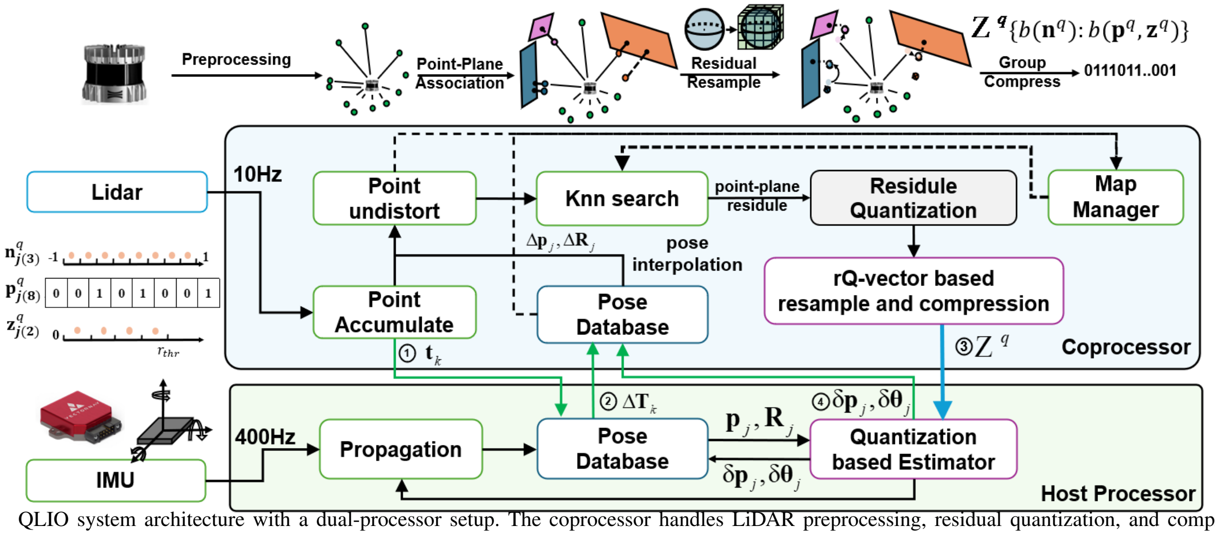

Fig. 2. QLIO system architecture with a dual-processor setup. The coprocessor handles LiDAR preprocessing, residual quantization, and compression, while the host processor performs IMU propagation and quantized state estimation, optimizing efficiency for real-time localization.

geometric integrity. Our contribution can be summarized as follows.

- Distributed Multi-Processor LIO Architecture: We propose QLIO, a distributed LiDAR-Inertial Odometry framework, leveraging a co-processor for LiDAR preprocessing and a host processor for state estimation. This reduces computational load and bandwidth usage, specifically tailored for LiDAR data.
- Quantized MAP Estimation for LIO: We introduce a multi-bit quantized Maximum A Posteriori (QMAP) estimator for LIO, optimizing 3D LiDAR residuals, reducing data transmission while preserving localization accuracy.
- rQ-Vector-Based Adaptive Resampling and Compression: We present an rQ-vector-based resampling and compression technique, focusing on LiDAR's 3D geometric structure to improve data efficiency and minimize computational redundancy.
- Open-Source LIO Framework with Real-World Validation: Our open-source QLIO framework is validated on real-world datasets, demonstrating significant bandwidth reduction (14.1×) while maintaining high localization accuracy, making it ideal for SWaP-constrained systems.

II. Related Works

To reduce the computational load of the state estimation [61], LiDAR-SLAM systems often utilize the structural characteristics to quantize the point cloud measurements through feature extraction [62]–[65] or voxel downsampling [10], [11]. In subsequent work, downsampling based on an octree structure was proposed [66], as well as adaptive downsampling methods [17] to improve performance in degradation scenarios [67]. A loop-closure BTC descriptor [68] was proposed to obtain a lightweight representation using binary height encoding.

Quantized state estimation [69], though underexplored, has a foundation in observation quantization. The field originated from the SOI-KF framework [45], which introduced 1-bit residual sign discrimination to balance information retention and efficiency. Later advancements extended quantization theory to distributed architectures through multi-bit MAP estimation and IQEKF variants [46], [47], [70]. QVIO [56] further developed this by integrating dual quantization strategies: zQVIO for multi-bit observation quantization and rQVIO for 1-bit residual quantization, achieving millimeter precision with minimal bandwidth. While effective for VIO, QVIO's 2D pixel-based quantization does not generalize to LIO, which depends on 3D point-cloud geometry. Addressing this gap, QLIO employs structured point-to-plane residual quantization and adaptive rQ-vector grouping for efficient LiDAR data compression while preserving geometric consistency. This innovation bridges quantization theory to LIO, ensuring accurate and bandwidth-efficient processing.

III. Quantized LiDAR-Inertial-Odometry

Our framework inherits architectural principles from FastLIO2 [10] and QVIO [56], assuming fixed and pre-calibrated LiDAR-IMU extrinsic parameters excluded from the optimization manifold. The system maintains the core state vector defined in FastLIO2:

$$\mathbf{x} \triangleq \left[{}^G\mathbf{R}_I^T, {}^G\mathbf{p}_I^T, {}^G\mathbf{v}_I^T, \mathbf{b}_\omega^T, \mathbf{b}_a^T, {}^G\mathbf{g}_I^T\right]^T \in \mathcal{M} \tag{1}$$

$$\mathbf{u} \triangleq \left[\omega_m^T \quad \mathbf{a}_m^T\right]^T, \mathbf{w} \triangleq \left[\mathbf{n}_\omega^T \quad \mathbf{n}_\mathrm{a}^T \quad \mathbf{n}_{\mathrm{b}\omega}^T \quad \mathbf{n}_{\mathrm{ba}}^T\right]^T \tag{2}$$

$$\mathbf{x}_{i+1} = \mathbf{x}_i \boxplus \left(\Delta t \mathbf{f}\left(\mathbf{x}_i, \mathbf{u}_i, \mathbf{w}_i\right)\right) \tag{3}$$

$\mathbf{R}$, $\mathbf{p}$, and $\mathbf{v}$ represent rotation, translation, and velocity, respectively. $\mathbf{b}_a$ and $\mathbf{b}_\omega$ are acceleration biases and the angular velocity of the IMU. $\mathbf{g}$ denote the gravity and $\mathbf{w}$ and $\mathbf{u}$ denote IMU measurement noise and control input. **Global frame** ${}^G(\cdot)$ represents the World-fixed reference frame initialized at system startup. **Body frame** ${}^I(\cdot)$ rigidly attached to IMU with origin at its measurement center, establishing the inertial reference for kinematic computations and **LiDAR frame** ${}^L(\cdot)$ defined by LiDAR's optical center, where raw point clouds are initially captured in this right-handed Cartesian system.

A. System Overview

Our system architecture comprises two synchronized components as illustrated in Figure 2.

$$\begin{aligned} \mathbf{x}_{i+1} &= \mathbf{x}_i \boxplus \left(\Delta t \mathbf{f}\left(\widehat{\mathbf{x}}_i, \mathbf{u}_i, \mathbf{0}\right)\right), \widehat{\mathbf{x}}_0 = \overline{\mathbf{x}}_{k-1}, \\ \widehat{\mathbf{P}}_{i+1} &= \mathbf{F}_{\mathbf{x}_i} \widehat{\mathbf{P}}_i \mathbf{F}_{\mathbf{x}_i}^T + \mathbf{F}_{\mathbf{w}_i} \mathbf{Q}_i \mathbf{F}_{\mathbf{w}_i}^T ; \widehat{\mathbf{P}}_0 = \overline{\mathbf{P}}_{k-1} \end{aligned} \tag{4}$$

The host processor performs IMU integration and pose updates (Equation 4), while a dedicated co-processor efficiently handles point cloud processing, including nearest neighbor search and residual computation. Unlike Visual-Inertial Odometry (VIO) systems that rely on image-based feature tracking and quantized pixel measurements [56], our method is tailored for LiDAR-Inertial Odometry (LIO), leveraging structured point-to-plane residual quantization to compress geometric information efficiently. By integrating a quantization-aware four-stage pipeline, our architecture minimizes bandwidth and computational costs while maintaining precise inter-processor synchronization.

1. Upon LiDAR scan arrival at timestamp $\mathbf{t}_k$, coprocessor sent the pose request with the timestamp $\mathbf{t}_k$ to the host-processor.

2. The host-processor responds with the relative pose transform $\Delta \mathbf{T}_k = (\Delta \mathbf{p}_k, \Delta \mathbf{R}_k)$ obtained by the short term IMU integration from $\mathbf{t}_{k-1}$ to $\mathbf{t}_k$. Then the coprocessor downsample and undistorted the points using the relative pose interpolation and transform points to the global coordinate system:

$$\begin{aligned} {}^L\mathbf{p}_j &= {}^L\mathbf{T}_I \Delta \mathbf{T}_j {}^I\mathbf{T}_L \mathbf{p}_j \\ {}^G\mathbf{p}_j &= \Delta \mathbf{T}_k \mathbf{T}_{k-1} {}^I\mathbf{T}_L {}^L\mathbf{p}_j \end{aligned} \tag{5}$$

where $\Delta \mathbf{T}_j = \left(\Delta \mathbf{p}_j, \Delta \mathbf{p}_{j_{ori}}\right)$ respected to the LiDAR relative pose from the $\mathbf{p}_{j_{ori}}$ arrived time $\mathbf{t}_j$ to the end of the scan $\mathbf{t}_k$. Then using the ${}^G\mathbf{p}_j$ perform knn search based on map manager, finding the nearest five points. If the they can fit an plane, we will find the associations between points and plane, as well as the residual measurements $\mathbf{z}_i$ in [10]:

$$\mathbf{z}_i = \mathbf{u}_i^T \left({}^G\mathbf{p}_i - {}^G\mathbf{m}_i\right) = \|\mathbf{n}_i\|, \quad \mathbf{n}_i = d_i \cdot \frac{\mathbf{u}_i^T}{\|\mathbf{u}_i^T\|} \tag{6}$$

$$0 = \mathbf{h}_i\left(\mathbf{x}_k, {}^L\mathbf{p}_i + {}^L\mathbf{n}_i\right) \tag{7}$$

$$\mathbf{z}_i + \mathbf{h}_i(\mathbf{x}_k - \hat{\mathbf{x}}_k) = -\mathbf{v}_i \sim \mathcal{N}\left(0, \mathbf{R}_i\right) \tag{8}$$

where $\mathbf{v}_i$ and $\mathbf{R}_i$ denote the measurment noise and covariance matrix. Residual vector $\mathbf{n}$ is obtained by projecting the displacement vector $(\mathbf{p}_i - \mathbf{m}_i)$ onto the normal vector $\mathbf{u}_i^T$ of the associated plane.

3.Once point-plane association is established, the overall state estimation problem will solely depend on the associated points and residual vectors. We can use s bits quantize them with $\mathbf{f}_{p_{(s)}}^q$, $\mathbf{f}_{z_{(s)}}^q$, $\mathbf{f}_{n_{(s)}}^q$ and perform resampling(Section III-B):

$$\begin{aligned} \mathbf{z}_i^q &= \mathbf{b}\left(\mathbf{z}_i\right) = \mathbf{f}_{z(s)}^q\left(\mathbf{z}_i\right) \\ {}^L\mathbf{p}_i^q &= \mathbf{f}_{p(s)}^q\left({}^L\mathbf{p}_i\right), {}^G\mathbf{n}_i^q = \mathbf{f}_{n(s)}^q\left({}^G\mathbf{n}_i\right) \end{aligned} \tag{9}$$

and then zip the all quantization result $\mathbf{z}^q, \mathbf{p}^q, \mathbf{n}^q$ to a group, $\mathcal{Z}^q$, and compressed to the bitstream, transmitting them to the host processor.

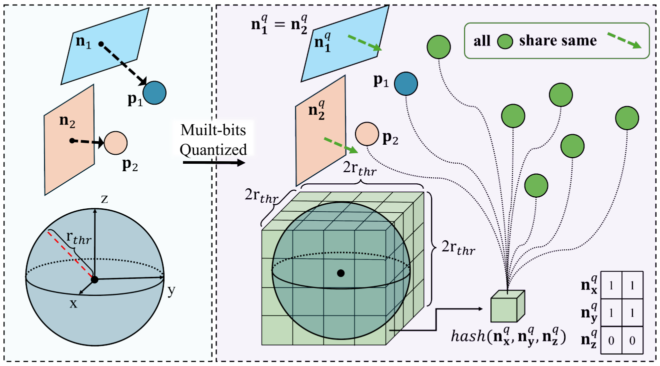

Fig. 3. The residual vector space is quantized into a voxel space based on INT-2 quantization. 3D points with the same quantized residuals are grouped into the same hash index. Additionally, the point clouds can be compressed together according to the hash index.

4.The host processor performs a quantization-based estimator (Section III-C) and updates the pose to the database.

B. rQ-vector based Resampling and Group Compression

Although point cloud observations have undergone preliminary filtering through voxel downsampling and point-plane association, the remaining points and residual vectors (28 bytes per measurement based on Float32) remain computationally intensive for direct transmission, consuming substantial communication bandwidth. As shown in figure 3, we propose a quantization-based resampling approach that leverages quantized residual vector to construct hash indices and group the points, then adaptive downsample the points those shared with same rQ-vector. If we assume residual $\mathbf{z} = \|\mathbf{n}_i\| < r_{thr}$, the quantization discretizes residual vectors by allocating s bits per axis, creating 2^s intervals in each dimension that voxelize the normal vector space. This produces 2^{3s} unique space cells $\mathcal{C}_k$ where $k \in [1, 2^{3s}]$, each corresponding to a specific rQ-vector range. For any observed point $\mathbf{p}_i$ with original residual vector $\mathbf{n}_i$, the quantization process $\mathbf{f}_{p(s)}^q$ will generate a $3s$-bit hash key. All points sharing identical $\mathbf{n}^q$ are aggregated into the same hash bucket $\mathcal{H}_k$, where the bucket index k can computed by hash function:

$$\mathbf{p}_i^q \in \mathcal{H}_k, k = hash(\mathbf{n}_i^q) \tag{10}$$

This spatial hashing ensures that the points with similar residuals are clustered into identical buckets. We then perform an adaptive 3D downsampling process to each buckets, where downsample size ds_k can calculate as follows:

$$ds_k = ds_0 + \alpha \sum_{i=1}^{n_k} \frac{1}{n_k} \|\mathbf{p}_i\| \tag{11}$$

where ds_0 is the cloud preprocessing downsampling parameter and α is the distance penalty coefficient derived from the average Euclidean distance of the points within the same bucket.

Additionally, the filtered points and their corresponding residuals can subsequently be grouped based on rQ-vectors, employing a group encoding scheme with shared header structures to reduce transmission bandwidth. And the groups

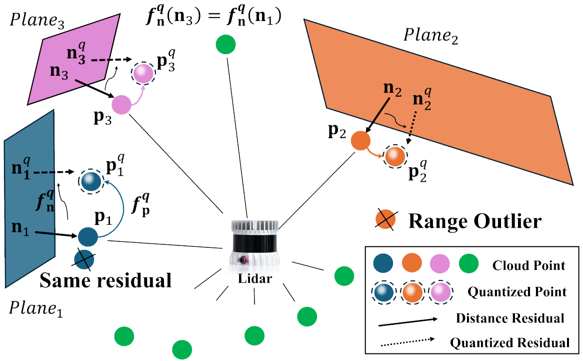

Fig. 4. Quantization of the points and their association point-plane residuals.

$\mathcal{Z}^q$ can then be constructed as:

$$\mathcal{Z}^q = \left\{b({}^1\mathbf{n}^q_k) : b({}^1\mathbf{z}^q_k, {}^1\mathbf{p}^q_k), b({}^2\mathbf{z}^q_k, {}^2\mathbf{p}^q_k), b({}^3\mathbf{z}^q_k, {}^3\mathbf{p}^q_k), \ldots\right\}, \left\{b({}^2\mathbf{n}^q_{k+1}) : b({}^1\mathbf{z}^q_{k+1}, {}^1\mathbf{p}^q_{k+1}), b({}^2\mathbf{z}^q_{k+1}, {}^2\mathbf{p}^q_{k+1}), \ldots\right\} \ldots \tag{12}$$

where $b(\cdot)$ denotes the multi-bit binary quantization.

C. Quantization-based Estimation

The quantization-based estimation problem can formulate as a MAP problem [46] [56]:

$$\begin{aligned} &\arg\max p\left(\mathbf{x}|\mathbf{b}\left(\mathbf{z}\right)\right) \\ &= p\left(\mathbf{x}\right)\prod_{i=1}^{m} p\left(\mathbf{b}\left(\mathbf{z}_i\right)|\mathbf{x}_i\right) \\ &= p\left(\mathbf{x}\right)\prod_{i=1}^{m} p\left(\mathbf{b}(\mathbf{z}_i) \in \left[\mathbf{z}^0_i, \mathbf{z}^1_i\right]|\mathbf{x}_i\right) \end{aligned} \tag{13}$$

The original measurements of the quantized observation residuals $\mathbf{z}^q$ should fall within the quantization interval between $\mathbf{z}^0$ and $\mathbf{z}^1$. Using Equation 8 and Q fuction to :

$$\begin{aligned} p\left(\mathbf{b}(\mathbf{z})|\mathbf{x}\right) &= \mathbf{Pr}\left\{\mathbf{z}^0 < -h\left(\mathbf{x}-\hat{\mathbf{x}}\right) - \mathbf{v} < \mathbf{z}^1 \middle| \mathbf{x}\right\} \\ &= \mathbf{Pr}\left\{\frac{\mathbf{z}^0}{\sigma} < \frac{-h\left(\mathbf{x}-\hat{\mathbf{x}}\right)-\mathbf{v}}{\sigma} < \frac{\mathbf{z}^1}{\sigma} \middle| \mathbf{x}\right\} \\ &= Q\left(\mathcal{X}^1\right) - Q\left(\mathcal{X}^0\right) \end{aligned} \tag{14}$$

$$\mathcal{X}^0 = \frac{-\mathbf{z}^0 - h\left(\mathbf{x}-\hat{\mathbf{x}}\right)}{\sigma},\ \mathcal{X}^1 = \frac{-\mathbf{z}^1 - h\left(\mathbf{x}-\hat{\mathbf{x}}\right)}{\sigma} \tag{15}$$

where $Q(\mathbf{x})$ is the Gaussian tail probability and σ is the scalar of noise standard deviation that normalizes the measurement noise. The MAP problem(Equation 13) is then equivalent to minimizing the following cost function:

$$\begin{aligned} \mathcal{L}\left(\mathbf{x}\right) &= \left\|\mathbf{x}_k \boxminus \hat{\mathbf{x}}_k\right\|^2_{\mathbf{P}_k} + \sum_{i=1}^{m} 2ln\left(Q\left(\mathcal{X}^1_i\right) - Q\left(\mathcal{X}^0_i\right)\right) \\ &= \left\|\mathbf{x}_k \boxminus \hat{\mathbf{x}}_k\right\|^2_{\mathbf{P}_k} + \sum_{i=1}^{m} \left\|\mathbf{H}^{'}_i\tilde{\mathbf{x}} + z^{'}_i\right\|^2_{\mathbf{R}^{'}_k} \end{aligned} \tag{16}$$

By performing Taylor expansion on Equation 16 and the assumption of quantized measurement-based posterior PDFs will still be close to bell-shaped Gaussian PDFs [46], [47] the above equation can be solved using Kalman filtering [10], [56]:

$$\begin{aligned} \mathbf{K} &= \left(\mathbf{H}^{'T}\mathbf{R}^{'}\mathbf{H}^{'} + \mathbf{P}^{-1}\right)\mathbf{H}^{'\mathbf{T}}\mathbf{R}^{'-1}, \\ \overline{\mathbf{x}}_k &= \hat{\mathbf{x}}_k \boxplus \left(-\mathbf{K}\mathbf{z}^{'}_k - \left(\mathbf{I} - \mathbf{K}\mathbf{H}^{'}\right)\left(\mathbf{J}^{'-1}\right)\tilde{\mathbf{x}}_k\right) \end{aligned} \tag{17}$$

Unlike the iterative approach [46], our method is executed in a single pass, eliminating the need for multiple iterations. For further technical details, we refer the reader to the relevant works [47], [56].

IV. Experimental Results

A. Quantization Precision Analysis

We adopt a simple and efficient fixed-point quantization for the LiDAR point $\mathbf{p}$, the residual vector $\mathbf{n}$ and the meansurment $\mathbf{z}$, which maintains a static codebook configuration. We use $(-r_{max}, r_{max})$ to represent the maximum quantization range of the LiDAR point (e.g. r_{max}=200m) and filter out the residual norm greater than the threshold r_{thr} (e.g. 0.04 m), meaning the residual space is truncated into a spherical space (see figure 3). Let l_p, l_n and l_z as the quantization number of bits, fixed-point multi-bit $\mathbf{p}^q$, $\mathbf{n}^q$, $\mathbf{z}^q$ and the required bits number with out rQ N can represent as:

$$\begin{aligned} &\mathbf{p}^q = \mathbf{f}^q_p\left(\mathbf{p}\right) = (p^q_1, p^q_2, p^q_3),\ \mathbf{n}^q = \mathbf{f}^q_n\left(\mathbf{n}\right) = (n^q_1, n^q_2, n^q_3) \\ &p^q_j = \left\lfloor \frac{p_j + r_{max}}{2^{(1-l_p)}r_{max}} \right\rfloor \cdot 2^{(1-l_p)}r_{max} - r_{max} + 2^{(-l_p)}r_{max} \\ &n^q_j = \left\lfloor \frac{n_j + r_{thr}}{2^{(1-l_n)}r_{thr}} \right\rfloor \cdot 2^{(1-l_n)}r_{thr} - r_{thr} + 2^{(-l_n)}r_{thr} \\ &\mathbf{z}^q = \left\lfloor \frac{\mathbf{z}}{2^{(-l_z)}r_{thr}} \right\rfloor \cdot 2^{(-l_z)}r_{thr} + 2^{(-l_z-1)}r_{thr} \\ &N = 3l_p + 3l_n + l_z, j = 1, 2, 3. \end{aligned} \tag{18}$$

where $\left\lfloor \frac{p_j + r_{max}}{2^{(1-l_p)}r_{max}} \right\rfloor$, $\left\lfloor \frac{n_j + r_{thr}}{2^{(1-l_n)}r_{thr}} \right\rfloor$, $\left\lfloor \frac{\mathbf{z}}{2^{(-l_z)}r_{thr}} \right\rfloor$ can then compressed to the binary bit stream based on rQ-vector grouping compressed strategy Equation 12.

B. Real World Experiment

We evaluate the proposed QLIO on the MCD-NTU dataset [41]. The MCD-NTU dataset was collected using an ATV within the NTU campus with a solid-state hybrid LiDAR (Livox-Mid70, Mid), a mechanical LiDAR (Ouster128, OS) and an external IMU sensor(VN100), including 6 sequences. Our system is built based on FastLIO [10]. We first conduct a comparative evaluation between FastLIO and our residual-quantized (rQ) vector-based resampling strategy (rQrs) employing fixed hash hierarchy levels ($l_n = 3$, equivalent to 512 hash buckets). The "X" means failed below and the average absolute trajectory error(ATE) results are shown in the Table I. It can be observed that the accuracy of FastLIO has been significantly improved after the application of rQrs, which is attributed to the resampling of the point observation and the residual, as shown in the Figure 5.

The fixed-step downsampling method arranges observations more regularly(see Figure 5.a "without rQ-resample"),

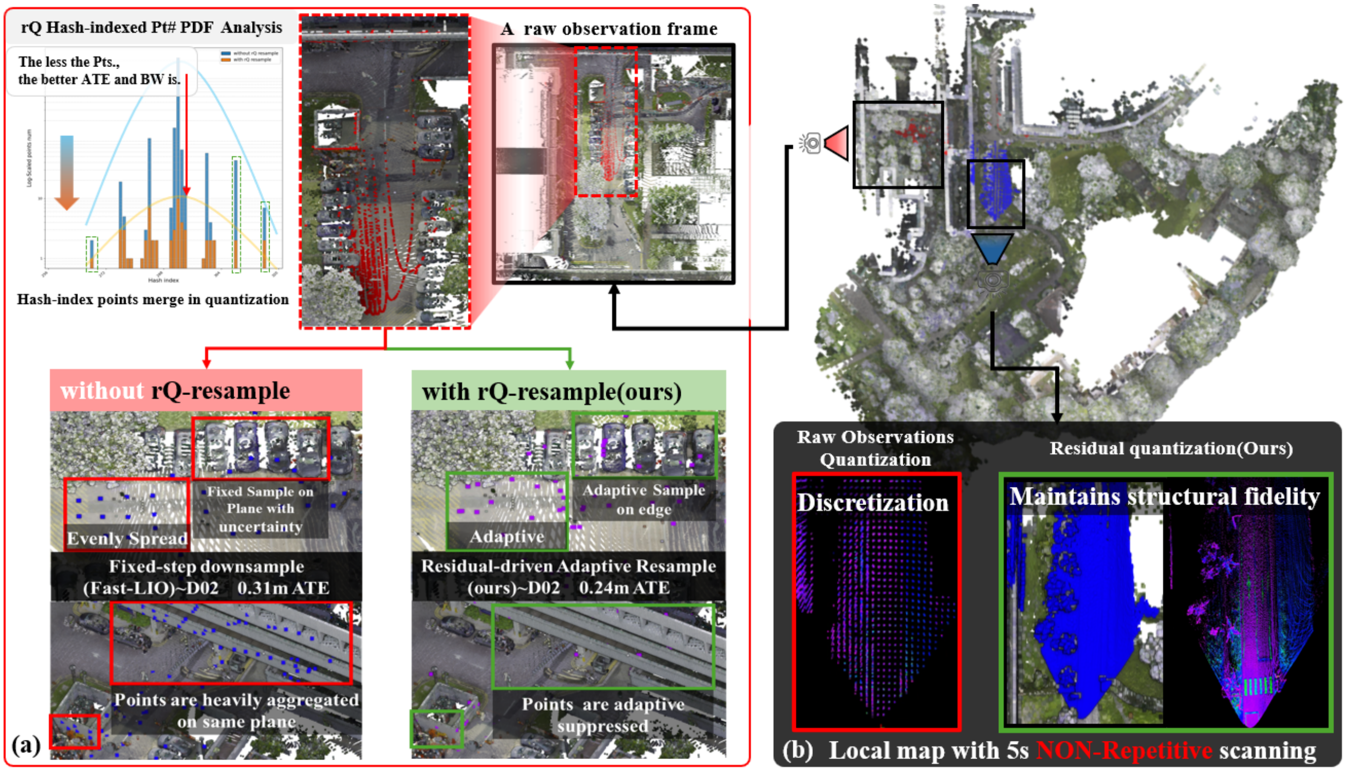
rQ Hash-indexed Pt# PDF Analysis
The less the Pts.,
the better ATE and BW is.
Hash-index points merge in quantization
A raw observation frame
without rQ-resample
with rQ-resample(ours)
Fixed Sample on Plane with uncertainty
Evenly Spread
Adaptive Sample on edge
Adaptive
Fixed-step downsample (Fast-LIO)~D02 0.31m ATE
Residual-driven Adaptive Resample (ours)~D02 0.24m ATE
Points are heavily aggregated on same plane
Points are adaptive suppressed
(a)
Raw Observations Quantization
Residual quantization(Ours)
Discretization
Maintains structural fidelity
(b) Local map with 5s NON-Repetitive scanning

Fig. 5. The test result on MCD-NTU dataset. (a) Conventional methods exhibit uniform planar point distributions, while our rQ-resample strategy preserves residual coherence, achieving lower ATE with fewer points. (b) Standard INT8 compression to the raw observation (left) induces discrete local map errors and causes point-plane association failure, whereas residual-aware quantization maintains structural fidelity.

TABLE I

ATE IN METERS FOR DIFFERENT METHODS AND SEQUENCES.

Method/Seq.	**D01**	**D02**	**D10**	**N04**	**N08**	**N13**
OS	1.48	0.27	2.08	1.59	2.05	1.29
OS-rQrs	1.28	**0.23**	**1.28**	1.25	1.62	**0.89**
Mid	1.09	0.31	1.63	0.85	1.54	1.88
Mid-rQrs	**0.91**	0.24	1.47	**0.49**	**1.33**	2.33
OS-dacro(8)	X	X	X	X	X	X
Mid-INT(8)	X	15.75	X	X	X	X

TABLE II

FOR DIFFERENT METHODS AND SEQUENCES.

Method/Seq.	d-c cost/ms	p.r	bits/mea.	avg.num
OS	0	1	96	15020.65
OS-rQrs	0.59	1	96	566.07
Mid	0	1	96	2799.16
Mid-rQrs	0.23	1	96	**248.98**
OS-dacro(8)	1868.25	**3.98**	24.12	16555.97
Mid-INT(8)	**0.14**	4	**24**	2665.32

yet induces directional residual concentration bias through its fixed sampling interval, resulting in observation direction imbalance (as per distribution analysis). However, our method resolves this imbalance, achieving measurement uniformity with fewer sampling points while maintaining optimization stability.

In addition, we implemented two conventional point cloud compression methods within our LiDAR-Inertial Odometry (LIO) pipeline to simulate centralized processing with fully compressed observations. To our knowledge, conventional compression methods are rarely tested in such tightly coupled LIO frameworks (most existing evaluations focus on pure LiDAR odometry like LOAM). Specifically, we applied INT8 Quantization and 8-bit Dacro two distinct compression techniques. Both approaches maintained identical post-processing pipelines with the original LIO algorithm where p.r (point cloud compression ratio), d-c (decompress-compress) cost time, and avg.num (average number of points per scan included in state estimation) were measured under varying compression configurations. Surprisingly, both methods exhibited significant performance degradation: INT8 fails because of the precision loss from point value truncation destroyed planar continuity, as shown in Figure 5.b, where original smooth surfaces became discrete grids. This fundamentally disrupted point-to-plane data association. Dacro limits because of the excessive compression/decompression latency (see Table II) caused cumulative IMU integration drift during processing gaps, ultimately compromising state estimation. It should be noted that the observed INT8 quantization-induced plane matching failures may stem from implementation specifics rather than inherent limitations of INT8 quantization. Alternative quantization strategies (e.g., log-scale quantization, variable-bit quantization) could theoretically resolve this, though our exhaustive experiments with these approaches revealed suboptimal performance-all variants required extensive hyperparameter tuning with additional parameters (detailed ablation studies will be documented in the supplementary material) thereby introducing prohibitive complexity in benchmark control. Consequently, we adopted the deterministic min-max INT8 quantization scheme to ensure reproducibility, sacrificing marginal accuracy for experimental stability. Notably, these approaches represent common paradigms for edge-device state estimation and we will then introduce our targeted solution QLIO framework.

TABLE III

ATE METERS/RADS IN DIFFERENT l_n INFLUENCES

M/O	l_n	**D01**	**D02**	**D10**	**N04**	**N08**	**N13**
OS	1	X	1.21/0.02	X	X	X	X
	2	1.57/**0.02**	**0.25/0.01**	1.39/0.03	1.37/**0.03**	1.54/**0.02**	**0.86/0.04**
	3	1.41/**0.02**	0.29/**0.01**	**1.36**/0.03	1.36/**0.03**	1.82/**0.02**	0.91/**0.04**
Mid	1	X	X	X	X	X	X
	2	1.96/0.03	0.56/0.02	1.57/**0.02**	1.32/**0.03**	3.89/**0.02**	X
	3	**1.23**/0.03	0.29/0.02	1.41/**0.02**	**0.62/0.03**	**1.40/0.02**	X

TABLE IV

ATE (M/RAD) ACROSS METHODS AND SEQUENCES WITH VARYING l_z.

M/O	l_z	**D01**	**D02**	**D10**	**N04**	**N08**	**N13**
OS	1	1.37/**0.02**	**0.24/0.01**	**1.32**/0.03	1.40/0.03	1.76/**0.02**	0.88/**0.04**
	2	1.57/**0.02**	**0.24/0.01**	1.36/0.03	1.32/**0.03**	1.58/**0.02**	**0.86/0.04**
	3	1.30/**0.02**	0.25/**0.01**	1.44/0.03	1.41/**0.03**	1.90/**0.02**	0.88/**0.04**
	4	**1.23/0.02**	**0.24/0.01**	1.51/0.03	1.43/**0.03**	1.79/**0.02**	**0.86/0.04**
Mid	1	1.34/**0.02**	0.43/0.02	1.93/**0.02**	1.75/0.04	3.09/**0.02**	X
	2	1.99/**0.02**	0.29/0.02	1.97/**0.02**	1.23/**0.03**	3.28/**0.02**	X
	3	1.36/**0.02**	0.32/0.02	1.38/**0.02**	**1.21**/0.04	**1.56/0.02**	X
	4	1.26/**0.02**	0.31/0.02	1.57/**0.02**	1.26/0.04	2.46/**0.02**	X

We first performed comparative ablation studies on residual component quantization bits l_n versus residual-vector quantization bits l_z (which may also directly determine the number of the hash bucket) systematically varying each parameter from 1-bit to 3-bit configurations under rQrs. This parametric sweep enabled empirical analysis of accuracy variations under progressive quantization refinement. Our experiments demonstrate that residual vector quantization with $l_n = 1$ consistently induces system failure across both OS and MID platforms, indicating catastrophic sensitivity to insufficient quantization precision. The experimental results of increasing the quantization level reveal distinct platform-dependent characteristics: l_n demonstrates negligible accuracy variation on OS, whereas MID exhibits consistent precision improvement that correlates with its unique multi-echo radar architecture. As the residual-vector quantization bits l_z increases, both MID and OS demonstrate minimal variation across all test sequences, with the notable exception of the N13 sequence (as shown in Table V).

On the N13 sequence involving rapid 180° rotations, we observed MID70 divergence. This is attributed to mid70's single-frame observation mechanism, which differs from Ouster in that it fails to capture a complete spherical point cloud. The incomplete spatial sampling leads to concentrated distributions of residual vectors after low-bit quantization, significantly disrupting original data associations. As we progressively increased quantization bits, system accuracy recovered accordingly. Precision returned to baseline levels when $l_n = 3$. This demonstrates that QLIO requires higher bit allocation under degraded conditions to maintain system stability. However, even when increasing to 4-bit quantization (compared to the original 32-bit float representation), this tradeoff remains worthwhile for devices with stringent communication bandwidth requirements.

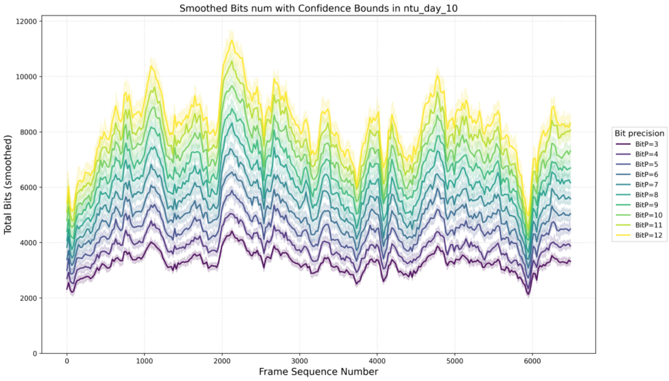

Fig. 6. The total bits per frame in D10 seq.

TABLE V

ATE UNDER VARYING l_z AND l_n USING MID70 IN N13.

N13-MID	$l_n = 1$	$l_n = 2$	$l_n = 3$
$l_z = 1$	X	X	X
$l_z = 2$	X	X	1.31/**0.04**
$l_z = 3$	X	15.54/0.14	**1.25/0.04**
$l_z = 4$	X	4.14/0.06	1.45/**0.04**

Based on our prior experiments, we fixed the 3-bit residual vector quantization and 2-bit residual quantization schemes($l_n = 3$, $l_z = 2$) then conducted systematic tests on LiDAR point quantization bits ranging from 3 to 12 bits using Ouster. The corresponding ATE meter/rads results, per-observation bit counts, and byte consumption are detailed inTable VI .The results demonstrate that implementing the rQrs strategy significantly enhances precision while drastically reducing observation counts. This efficiency gain stems from our group compression protocol - points sharing same rQ-vector are packaged together during compression, thereby reducing the cost required for rQ-vector processing. Surprisingly, significant fluctuations in the quantization bit number did not lead to substantial accuracy loss. Notably, progressive reduction of quantization bits from 12 to 3 bits reveals a precision-bandwidth trade-off: 3-bit quantization achieves minimal measurement overhead (5.84bits per mea.)but also the minimal precision while more bit quantization achieves more precision gain. Crucially, even with 12-bit quantization under rQrs(14.84bits per mea.), total per-frame bandwidth consumption on the MCD-D10 dataset remains below 1.5kB(as shown in Figure 6). This translates to <1.2Mbps total bandwidth usage, enabling practical transmission via edge-device-friendly protocols like UART/SPI (typically supporting 1-10Mbps rates).

V. Conclusion and Future Work

QLIO redefines LIO deployment on edge devices by integrating multi-processor quantization, adaptive resampling, and extreme bandwidth reduction. It enables real-time LiDAR state estimation on low-power platforms such

TABLE VI

ATE IN METERS/RADS AND THE BITS/BYTES OCCUPATION PER MEASUREMENTS FOR l_p AND SEQUENCES.

seq	rQrs	$l_p = 3$	$l_p = 4$	$l_p = 5$	$l_p = 6$	$l_p = 7$	$l_p = 8$	$l_p = 9$	$l_p = 10$	$l_p = 11$	$l_p = 12$
D01	**w/o**	2.95/0.04	2.10/0.04	1.76/0.04	1.62/0.04	1.59/0.04	1.63/0.04	1.61/0.04	1.57/0.04	1.64/0.04	1.68/0.04
	w/	1.98/0.03	1.74/0.03	1.51/0.03	1.52/0.03	1.40/0.03	1.48/0.03	1.46/0.03	1.55/0.03	1.44/0.03	1.51/0.03
D02	**w/o**	0.33/0.01	0.31/0.01	0.32/0.01	0.31/0.01	0.31/0.01	0.31/0.01	0.31/0.01	0.32/0.01	0.31/0.01	0.31/0.01
	w/	0.24/0.01	0.24/0.01	0.25/0.01	0.26/0.01	0.25/0.01	0.25/0.01	0.25/0.01	0.25/0.01	0.26/0.01	0.26/0.01
D10	**w/o**	2.74/0.04	2.52/0.04	2.74/0.04	2.76/0.04	2.63/0.04	2.71/0.04	2.69/0.04	2.66/0.04	2.61/0.04	2.58/0.04
	w/	1.18/0.04	1.06/0.04	1.31/0.04	1.27/0.04	1.36/0.04	1.34/0.04	1.30/0.04	1.30/0.04	1.42/0.04	1.30/0.04
N04	**w/o**	1.60/0.03	1.61/0.03	1.64/0.03	1.64/0.03	1.67/0.03	1.66/0.03	1.70/0.03	1.67/0.03	1.67/0.03	1.65/0.03
	w/	1.21/0.03	1.27/0.03	1.31/0.03	1.41/0.03	1.31/0.03	1.37/0.03	1.41/0.03	1.39/0.03	1.39/0.03	1.40/0.03
N08	**w/o**	2.62/0.02	2.76/0.02	3.17/0.02	2.91/0.02	3.08/0.02	2.98/0.02	2.98/0.02	3.07/0.02	3.06/0.02	2.89/0.02
	w/	2.07/0.02	1.53/0.02	1.67/0.02	1.85/0.02	1.68/0.02	1.48/0.02	1.54/0.02	1.84/0.02	1.56/0.02	1.75/0.02
N13	**w/o**	1.41/0.05	1.34/0.05	1.30/0.05	1.30/0.05	1.29/0.05	1.29/0.05	1.29/0.05	1.30/0.05	1.29/0.05	1.29/0.05
	w/	0.89/0.05	0.85/0.05	0.86/0.05	0.87/0.05	0.88/0.05	0.88/0.05	0.89/0.05	0.87/0.05	0.87/0.05	0.87/0.05
avg.b/B	**w/o**	20/2.50	23/2.87	26/3.25	29/3.62	32/4.00	35/4.37	38/4.75	41/5.12	44/5.50	47/5.87
	w/	5.84/0.73	6.84/0.85	7.84/0.98	8.84/1.10	9.84/1.23	10.84/1.35	11.84/1.48	12.84/1.60	13.84/1.73	14.84/1.85

as drones and humanoid robots. Unlike 2D visual feature quantization methods like QVIO, QLIO introduces structured 3D point-to-plane quantization and a LiDAR-specific MAP-based estimator. The experimental results show a 15.1× per-observation bandwidth reduction while preserving localization accuracy, demonstrating the need for sensor-specific quantization strategies in LiDAR odometry.

However, QLIO currently relies on heuristic quantization parameters, which may **limit** adaptability to dynamically changing environments. Additionally, its **multi-processor architecture**, while improving efficiency, introduces hardware dependencies that require careful integration with different computing platforms. Our future research will prioritize the development of an **intelligent parameter optimization framework**, such as implementing a reinforcement learning-based architecture with multi-head attention mechanisms for adaptive point cloud feature selection. This approach will systematically replace heuristic parameter design in QLIO with bandwidth-aware optimization, particularly improving spectral efficiency in SLAM applications through adaptive data compression strategies based on communication condition monitoring.

REFERENCES

[1] S. Yuan, Y. Yang, T. H. Nguyen, T.-M. Nguyen, J. Yang, F. Liu, J. Li, H. Wang, and L. Xie, "Mmaud: A comprehensive multi-modal anti-uav dataset for modern miniature drone threats," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*, 2024.

[2] T. He, Z. Luo, X. He, W. Xiao, C. Zhang, W. Zhang, K. M. Kitani, C. Liu, and G. Shi, "Omnih2o: Universal and dexterous human-to-humanoid whole-body teleoperation and learning," *Conference on Robot Learning (CoRL)*, 2024.

[3] J. Li, S. Yuan, M. Cao, T.-M. Nguyen, K. Cao, and L. Xie, "Hcto: Optimality-aware lidar inertial odometry with hybrid continuous time optimization for compact wearable mapping system," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 211, pp. 228–243, 2024.

[4] J. Li, Q. Leng, J. Liu, X. Xu, T. Jin, M. Cao, T.-M. Nguyen, S. Yuan, K. Cao, and L. Xie, "Helmetposer: A helmet-mounted imu dataset for data-driven estimation of human head motion in diverse conditions," in *Proc. of ICRA, IEEE*, 2025.

[5] X. Zheng and J. Zhu, "Traj-lo: In defense of lidar-only odometry using an effective continuous-time trajectory," *IEEE RA-L*, 2024.

[6] S. Yuan, B. Lou, T.-M. Nguyen, P. Yin, M. Cao, X. Xu, J. Li, J. Xu, S. Chen, and L. Xie, "Large-scale uwb anchor calibration and one-shot localization using gaussian process," in *Proc. of ICRA*. IEEE, 2025.

[7] S. Wang, S. Li, Y. Zhang, S. Yu, S. Yuan, R. She, Q. Guo, J. Zheng, K. H. Ong, L. Chandra, S. Srijeyan, A. Sivadas, T. Aggarwal, H. Liu, H. Zhang, C. Chen, J. Jiang, L. Xie, and W. P. Tay, "Uavscenes: A multi-modal dataset for uavs," in *ICCV*, 2025.

[8] H. Cai, S. Yuan, X. Li, J. Guo, and J. Liu, "BEV-LIO(LC): Bev image assisted lidar-inertial odometry with loop closure," in *Proc. of IROS*. IEEE, 2025.

[9] T. Deng, G. Shen, X. Chen, S. Yuan, H. Shen, G. Peng, Z. Wu, J. Wang, L. Xie, D. Wang *et al.*, "Mcn-slam: Multi-agent collaborative neural slam with hybrid implicit neural scene representation," *arXiv preprint arXiv:2506.18678*, 2025.

[10] W. Xu, Y. Cai, D. He, J. Lin, and F. Zhang, "Fast-lio2: Fast direct lidar-inertial odometry," *IEEE Transactions on Robotics*, 2022.

[11] C. Bai, T. Xiao, Y. Chen, H. Wang, F. Zhang, and X. Gao, "Faster-lio: Lightweight tightly coupled lidar-inertial odometry using parallel sparse incremental voxels," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 4861–4868, 2022.

[12] T.-M. Nguyen, X. Xu, T. Jin, Y. Yang, J. Li, S. Yuan, and L. Xie, "Eigen is all you need: Efficient lidar-inertial continuous-time odometry with internal association," *IEEE Robotics and Automation Letters*, vol. 9, no. 6, pp. 5330–5337, 2024.

[13] Z. Qi, S. Yuan, F. Liu, H. Cao, T. Deng, J. Yang, and L. Xie, "Air-embodied: An efficient active 3dgs-based interaction and reconstruction framework with embodied large language model," *arXiv preprint arXiv:2409.16019*, 2024.

[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 *Proc. of IROS*. IEEE, 2020.

[15] T.-M. Nguyen, S. Yuan, M. Cao, L. Yang, T. H. Nguyen, and L. Xie, "Miliom: Tightly coupled multi-input lidar-inertia odometry and mapping," *IEEE Robotics and Automation Letters*, 2021.

[16] T. Shan, B. Englot, C. Ratti, and D. Rus, "Lvi-sam: Tightly-coupled lidar-visual-inertial odometry via smoothing and mapping," in *Proc. of ICRA*. IEEE, 2021.

[17] H. Lim, D. Kim, B. Kim, and H. Myung, "Adalio: Robust adaptive lidar-inertial odometry in degenerate indoor environments," in *International Conference on Ubiquitous Robots (UR)*. IEEE, 2023.

[18] T.-M. Nguyen, D. Duberg, P. Jensfelt, S. Yuan, and L. Xie, "Slict: Multi-input multi-scale surfel-based lidar-inertial continuous-time odometry and mapping," *IEEE Robotics and Automation Letters*, vol. 8, no. 4, pp. 2102–2109, 2023.

[19] I. Vizzo, T. Guadagnino, B. Mersch, L. Wiesmann, J. Behley, and C. Stachniss, "Kiss-icp: In defense of point-to-point icp–simple, accurate, and robust registration if done the right way," *IEEE Robotics and Automation Letters*, vol. 8, no. 2, pp. 1029–1036, 2023.

[20] K. Chen, R. Nemiroff, and B. T. Lopez, "Direct lidar-inertial odometry: Lightweight lio with continuous-time motion correction," in *Proc. of ICRA*. IEEE, 2023.

[21] Z. Chen, Y. Xu, S. Yuan, and L. Xie, "ig-lio: An incremental gicp-based tightly-coupled lidar-inertial odometry," *IEEE Robotics and Automation Letters*, vol. 9, no. 2, pp. 1883–1890, 2024.

[22] X. Ji, S. Yuan, P. Yin, and L. Xie, "Lio-gvm: an accurate, tightly-coupled lidar-inertial odometry with gaussian voxel map," *IEEE Robotics and Automation Letters*, vol. 9, no. 3, pp. 2200–2207, 2024.

[23] Y. Wu, T. Guadagnino, L. Wiesmann, L. Klingbeil, C. Stachniss, and H. Kuhlmann, "Lio-ekf: High frequency lidar-inertial odometry using extended kalman filters," in *Proc. of ICRA*. IEEE, 2024.

[24] M. Eisoldt, J. Gaal, T. Wiemann, M. Flottmann, M. Rothmann, M. Tassemeier, and M. Porrmann, "A fully integrated system for hardware-accelerated tsdf slam with lidar sensors (hatsdf slam)," *Robotics and Autonomous Systems*, vol. 156, p. 104205, 2022.

[25] J. Lin and F. Zhang, "R 3 live: A robust, real-time, rgb-colored, lidar-inertial-visual tightly-coupled state estimation and mapping package," in *Proc. of ICRA*. IEEE, 2022.

[26] T. Jin, X. Xu, Y. Yang, S. Yuan, T.-M. Nguyen, J. Li, and L. Xie, "Robust loop closure by textual cues in challenging environments," *IEEE Robotics and Automation Letters*, 2025.

[27] J. Li, T.-M. Nguyen, S. Yuan, and L. Xie, "Pss-ba: Lidar bundle adjustment with progressive spatial smoothing," in *Proc. of IROS*, 2024.

[28] J. Li, T.-M. Nguyen, M. Cao, S. Yuan, T.-Y. Hung, and L. Xie, "Graph optimality-aware stochastic lidar bundle adjustment with progressive spatial smoothing," in *arXiv preprint arXiv:2410.14565*, 2024.

[29] Y. Ma, J. Xu, S. Yuan, T. Zhi, W. Yu, J. Zhou, and L. Xie, "Mm-lins: a multi-map lidar-inertial system for over-degenerate environments," *IEEE Transactions on Intelligent Vehicles*, vol. -, pp. 1–11, 2024.

[30] P. Yin, H. Cao, T.-M. Nguyen, S. Yuan, S. Zhang, K. Liu, and L. Xie, "Outram: One-shot global localization via triangulated scene graph and global outlier pruning," in *Proc. of ICRA*. IEEE, 2024.

[31] C. Zhao, K. Hu, J. Xu, L. Zhao, B. Han, K. Wu, M. Tian, and S. Yuan, "Adaptive-lio: Enhancing robustness and precision through environmental adaptation in lidar inertial odometry," *IEEE IoT-J*, 2024.

[32] H. Rebecq, T. Horstschäfer, G. Gallego, and D. Scaramuzza, "Evo: A geometric approach to event-based 6-dof parallel tracking and mapping in real time," *IEEE Robotics and Automation Letters*, 2016.

[33] W. Wang, D. Zhu, X. Wang, Y. Hu, Y. Qiu, C. Wang, Y. Hu, A. Kapoor, and S. Scherer, "Tartanair: A dataset to push the limits of visual slam," in *Proc. of IROS*. IEEE, 2020, pp. 4909–4916.

[34] M. Helmberger, K. Morin, B. Berner, N. Kumar, G. Cioffi, and D. Scaramuzza, "The hilti slam challenge dataset," *IEEE Robotics and Automation Letters*, vol. 7, no. 3, pp. 7518–7525, 2022.

[35] A. D. Nair, J. Kindle, P. Levchev, and D. Scaramuzza, "Hilti slam challenge 2023: Benchmarking single+ multi-session slam across sensor constellations in construction," *IEEE RA-L*, 2024.

[36] C. Wang, J. Yuan, and L. Xie, "Non-iterative slam," in *International Conference on Advanced Robotics (ICAR)*. IEEE, 2017.

[37] Z. Yang, K. Xu, S. Yuan, and L. Xie, "A fast and light-weight noniterative visual odometry with rgb-d cameras," *Unmanned Systems*, vol. 0, no. 0, pp. 1–13, 2024.

[38] C. Zheng, W. Xu, Z. Zou, T. Hua, C. Yuan, D. He, B. Zhou, Z. Liu, J. Lin, F. Zhu *et al.*, "Fast-livo2: Fast, direct lidar-inertial-visual odometry," *IEEE Transactions on Robotics*, 2024.

[39] 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*, vol. 38, no. 2, pp. 958–977, 2021.

[40] 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," *IJRR*, vol. 41, no. 3, pp. 270–280, 2022.

[41] T.-M. Nguyen, S. Yuan, T. H. Nguyen, P. Yin, H. Cao, L. Xie, M. Wozniak, P. Jensfelt, M. Thiel, J. Ziegenbein *et al.*, "Mcd: Diverse large-scale multi-campus dataset for robot perception," in *Proc. of CVPR*, 2024, pp. 22 304–22 313.

[42] K. Xu, Y. Hao, S. Yuan, C. Wang, and L. Xie, "Airvo: An illumination-robust point-line visual odometry," in *Proc. of IROS*. IEEE, 2023.

[43] P. Pfreundschuh, H. Oleynikova, C. Cadena, R. Siegwart, and O. Andersson, "Coin-lio: Complementary intensity-augmented lidar inertial odometry," in *Proc. of ICRA*. IEEE, 2024.

[44] K. Xu, Y. Hao, S. Yuan, C. Wang, and L. Xie, "Airslam: An efficient and illumination-robust point-line visual slam system," *IEEE Transactions on Robotics*, 2025.

[45] A. Ribeiro, G. B. Giannakis, and S. I. Roumeliotis, "Soi-kf: Distributed kalman filtering with low-cost communications using the sign of innovations," *IEEE Transactions on signal processing*, 2006.

[46] N. Trawny, S. I. Roumeliotis, and G. B. Giannakis, "Cooperative multi-robot localization under communication constraints," in *Proc. of ICRA*. IEEE, 2009.

[47] E. J. Msechu, S. I. Roumeliotis, A. Ribeiro, and G. B. Giannakis, "Decentralized quantized kalman filtering with scalable communication cost," *IEEE Transactions on Signal Processing*, 2008.

[48] J. W. Hegeman, V. B. Sardeshmukh, R. Sugumaran, and M. P. Armstrong, "Distributed lidar data processing in a high-memory cloud-computing environment," *Annals of GIS*, vol. 20, no. 4, pp. 255–264, 2014.

[49] G. Mohanarajah, V. Usenko, M. Singh, R. D'Andrea, and M. Waibel, "Cloud-based collaborative 3d mapping in real-time with low-cost robots," *IEEE TASE*, 2015.

[50] R. Schnabel and R. Klein, "Octree-based point-cloud compression." *PBG@ SIGGRAPH*, vol. 3, no. 3, 2006.

[51] F. Galligan, M. Hemmer, O. Stava, F. Zhang, and J. Brettle, "Google/draco: a library for compressing and decompressing 3d geometric meshes and point clouds," 2018.

[52] D. Graziosi, O. Nakagami, S. Kuma, A. Zaghetto, T. Suzuki, and A. Tabatabai, "An overview of ongoing point cloud compression standardization activities: Video-based (v-pcc) and geometry-based (g-pcc)," *Transactions on Signal and Information Processing*, 2020.

[53] S. Wang, J. Jiao, P. Cai, and L. Wang, "R-pcc: A baseline for range image-based point cloud compression," in *International Conference on Robotics and Automation (ICRA)*. IEEE, 2022.

[54] J. Wang, D. Ding, Z. Li, X. Feng, C. Cao, and Z. Ma, "Sparse tensor-based multiscale representation for point cloud geometry compression," *IEEE TPAMI*, vol. 45, no. 7, pp. 9055–9071, 2022.

[55] D. Jiang, B. Liu, J. Wang, A. Hu, Y. Zhao, M. Bao, Z. Fan, Z. Shen, K. Wang, and C. Wang, "Live demonstration: A reconfigurable, energy-efficient and high-frame-rate ekf-slam accelerator based soc design for autonomous mobile robot applications," in *IEEE International Symposium on Circuits and Systems (ISCAS)*. IEEE, 2024.

[56] Y. Peng, C. Chen, and G. Huang, "Quantized visual-inertial odometry," in *Proc. of ICRA*. IEEE, 2024.

[57] Z. Huai and G. Huang, "A consistent parallel estimation framework for visual-inertial slam," *IEEE Transactions on Robotics*, 2024.

[58] X. Ji, S. Yuan, J. Li, P. Yin, H. Cao, and L. Xie, "Sgba: Semantic gaussian mixture model-based lidar bundle adjustment," *RAL*, 2024.

[59] P. Geneva, K. Eckenhoff, W. Lee, Y. Yang, and G. Huang, "Openvins: A research platform for visual-inertial estimation," in *IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020.

[60] G. P. Huang, A. I. Mourikis, and S. I. Roumeliotis, "Observability-based rules for designing consistent ekf slam estimators," *IJRR*, 2010.

[61] X. Li, S. Yuan, H. Cai, S. Lu, W. Wang, and J. Liu, "Ll-localizer: A life-long localization system based on dynamic i-octree," *arXiv preprint arXiv:2504.01583*, 2025.

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

[63] T. Shan and B. Englot, "Lego-loam: Lightweight and ground-optimized lidar odometry and mapping on variable terrain," in *Proc. of IROS*. IEEE, 2018.

[64] M. A. Esfahani, H. Wang, K. Wu, and S. Yuan, "Unsupervised scene categorization, path segmentation and landmark extraction while traveling path," in *ICARCV*. IEEE, 2020.

[65] J. Xu, W. Yu, S. Huang, S. Yuan, L. Zhao, R. Li, and L. Xie, "M-divo: Multiple tof rgb-d cameras enhanced depth-inertial-visual odometry," *IEEE IoT-J*, 2024.

[66] Z. Liu, H. Li, C. Yuan, X. Liu, J. Lin, R. Li, C. Zheng, B. Zhou, W. Liu, and F. Zhang, "Voxel-slam: A complete, accurate, and versatile lidar-inertial slam system," *arXiv preprint arXiv:2410.08935*, 2024.

[67] J. Xu, G. Huang, W. Yu, X. Zhang, L. Zhao, R. Li, S. Yuan, and L. Xie, "Selective kalman filter: When and how to fuse multi-sensor information to overcome degeneracy in slam," *arXiv preprint arXiv:2412.17235*, 2024.

[68] C. Yuan, J. Lin, Z. Liu, H. Wei, X. Hong, and F. Zhang, "Btc: A binary and triangle combined descriptor for 3-d place recognition," *IEEE Transactions on Robotics*, vol. 40, pp. 1580–1599, 2024.

[69] M. Fu and C. E. de Souza, "State estimation for linear discrete-time systems using quantized measurements," *Automatica*, vol. 45, no. 12, pp. 2937–2945, 2009.

[70] E. D. Nerurkar and S. I. Roumeliotis, "A communication-bandwidth-aware hybrid estimation framework for multi-robot cooperative localization," in *Proc. of IROS*. IEEE, 2013, pp. 1418–1425.