

RESEARCH ON HIGH-FREQUENCY ODOMETRY IN LASER SLAM USING POINT-SUBMAP MATCHING

Junyang Bian¹, He Huang¹, Junxing Yang^{1*}, Siqi Wang¹, Xin Zhang¹

¹ School of Geomatics and Urban Spatial Informatics, Beijing University of Civil Engineering and Architecture, Beijing, China - bji1066702005@163.com, huanghe@bucea.edu.cn, yangjunxing@bucea.edu.cn, w2248183443@163.com, zx2806404866@163.com

KEY WORDS: Point-by-point Update, LiDAR, Multi-sensor Fusion, Simultaneous Localization and Mapping.

ABSTRACT:

The stability of front-end odometry output is the cornerstone of real-time mapping and high-precision localization. However, in autonomous driving, factors such as sensor types, driving speed, and signal occlusion can impact system performance, particularly causing trajectory drift, reduced accuracy, and decreased reliability during high-speed travel. This study employs a point-by-point update approach for odometry, achieving high-frequency odometry output and mapping updates nearly matching the point sampling rate, effectively eliminating point cloud distortions and providing solid support for tracking high-speed motion and navigation localization. This method emphasizes maintaining registration accuracy and robustness under high-frequency output conditions. Moreover, by utilizing backward propagation to optimize the estimation of posture and position, and ignoring inter-frame processing, this research implements an iterative extended Kalman filter, achieving tight coupling with IMU data. Additionally, the use of the iVox data structure accelerates point cloud searching, enhancing system processing efficiency. Testing in various scenarios has demonstrated that our method exhibits superior performance and robustness across different scene data, showcasing its potential application in the field of autonomous driving.

1. INTRODUCTION

SLAM (Simultaneous Localization and Mapping) technology stands as a pivotal innovation in robotics and computer vision, enabling robots and autonomous systems to adeptly navigate uncharted environments while concurrently constructing a map of their surroundings. This technology finds extensive application in autonomous vehicles, drone navigation, and indoor robot maneuvering. Vision-based SLAM systems excel in precise positioning, yet they face challenges in generating dense maps. These challenges arise partly due to environmental factors like lighting and partly due to the constraints of computing resources. On the other hand, LiDAR offers accurate, dense, and direct 3D point cloud data, while the inertial measurement unit (IMU) provides real-time positional information. Together, they form a synergistic, tightly coupled LIO (Lidar-Inertial odometry) system.

Recent advancements in lidar odometry have focused on accumulating lidar points into frames to extract feature points (areas and lines), then matching and merging a point cloud frame with a previously matched submap to create a new submap, or alternatively, conducting direct frame-to-frame matching. This approach somewhat mitigates computational strain; however, it struggles with rapid map drift, particularly in high-speed vehicular movement, due to the reliance on substantial overlap between matching frames. One major drawback of scan-to-map registration is that the odometry is estimated at the rate of the scan (or frame), limiting the odometry output frequency at the frame rate. The limited output frequency will cause a delay in the odometry equal to the scan duration. In this paper,

In this work, we address these issues by point-by-point state update. More specifically, our contributions are as follows: 1) We propose a point-wise LiDAR-inertial odometry (LIO) framework, that integrates LiDAR points at their precise moment of capture, bypassing the need for frame accumulation. This approach effectively eliminates motion distortion within the frame, facilitating high frequency odometry outputs and mapping updates that closely match the point sampling rate; 2) In order to compensate for potential errors in point-submap matching, the system integrates IMU and LiDAR data into the backend. This integration forms a tightly coupled LIO system, utilizing Iteratively Extended Kalman Filter (IEKF) to construct state and motion equations; 3) To enhance the point cloud search rate and maintain the real-time performance of the LIO system, the frontend employs an incremental voxel (iVox) data structure. This structure tracks all map points within a specific map capacity, incorporates an LRU cache into the submap, and passively deletes long-unused voxels when capacity is exceeded.

2. RELATED

2.1 LiDAR Odometry

Recent advancements in 3D LiDAR odometry and mapping have predominantly harnessed the LiDAR Odometry and Mapping framework (Zhang and Singh, 2014). In this framework, raw LiDAR points are amalgamated into a frame (or scan) to delineate feature points, such as edges and planes. These identified feature points are subsequently aligned with the preceding scan to produce odometry data at the scan frequency (typically 10Hz), and a compilation of several recent scans forms a condensed submap. This submap is then incrementally integrated into the global map at a reduced rate (around 1Hz),

* Corresponding author

facilitating the refinement of the LiDAR's pose in relation to the map. This bifurcated structure of scan-to-scan and scan-to-map integration underpins the LOAM architecture and has catalyzed a plethora of subsequent scholarly inquiries. Notably, Lego-LOAM enhances scan-to-scan matching accuracy through the incorporation of ground constraints, while LINS (Qin et al.,2020) amalgamates IMU data with scan registration. Concurrently, other research endeavors have concentrated on amplifying computational efficiency or augmenting accuracy within this domain.

While the delineation between scan-to-scan and scan-to-map processes substantially mitigates the computational load for odometry, it precipitates a swift accrual of drift during the scan-to-scan registration phase. Furthermore, this registration necessitates significant overlap between successive scans, a condition that may not be consistently achievable with small FoV solid-state LiDARs. In response to these challenges, direct scan-to-map (or scan-to-local map) methodologies have gained traction, employing strategies such as point maps, ICP, NDT, Surfel maps, or voxel maps. To specifically address the constrained FoV challenge presented by solid-state LiDARs, some methods have introduced parallel scan-to-map techniques. Moreover, the incorporation of IMU measurements into the scan-to-map registration process, within an optimized iterated Kalman filter framework, has emerged as a promising solution. A pivotal element of the scan-to-map methodology is the sustenance of an adaptable map structure that supports the incremental integration of new scan points while ensuring efficient data access. The advent of an incremental k-d tree, or ikd-Tree, as a map structure, has played a crucial role in equipping systems like FAST-LIO2 (Xu et al.,2022) with the capability to execute real-time odometry and mapping for spinning LiDARs at 10Hz and solid-state LiDARs at 100Hz, even on resource-constrained ARM-based computing platforms.

A notable limitation of the scan-to-map registration process is its dependency on the scan (or frame) rate, which inherently restricts the odometry update frequency to this rate. Consequently, there is a latency in the odometry updates equivalent to the scan duration, creating an unnecessary constraint on the odometry bandwidth, as dictated by the Nyquist Shannon sampling theorem. This challenge has been partially mitigated by innovations such as Lola-SLAM (Karimi et al.,2021) and LLOL (Qu et al.,2022), which segment a scan into multiple subscans, registering each to the map instantaneously upon reception, thus achieving odometry frequencies of 160Hz and 80Hz, respectively. This remarkable enhancement in high-frequency state updates drastically reduces latency to the microsecond level and substantially increases the odometry bandwidth, representing a significant advancement in the field.

Within the expansive domain of spatial data management, voxel hashing stands out as a compelling alternative to traditional hierarchical tree-based architectures for visual SLAM and volumetric 3D reconstruction tasks. Implementations that utilize hashed voxels for map management and nearest neighbor searches unveil the capacity to significantly improve incremental mapping processes in LiDAR-Inertial Odometry (LIO) systems. This method showcases a scalable and efficient approach, which is particularly advantageous for real-time applications, highlighting its potential to revolutionize the landscape of spatial data handling and analysis.

2.2 Formulation of Inertial Measurements

In the realm of synergizing Inertial Measurement Unit (IMU) data with LiDAR point registration to augment robotic navigation and mapping capabilities, two primary paradigms emerge: the loosely coupled and tightly coupled approaches. Loosely coupled frameworks utilize IMU measurements to formulate an initial pose estimate, which serves as a precursor for the subsequent scan registration endeavors (Zhao et al.,2019). In contrast, tightly coupled methodologies integrate IMU data and LiDAR points within a cohesive optimization structure, delineating two main implementations: the Extended Kalman Filter (EKF)-based and the optimization-based techniques. EKF-based (Hemann et al.,2016) methods meticulously assimilate IMU measurements within the EKF's propagation phase to refine pose estimations, which are then coalesced with LiDAR measurements in the EKF's update phase. Alternatively, optimization-based approaches employ IMU data preintegration to ascertain relative pose constraints, integrating these preintegrated constraints with point registration discrepancies to bolster system robustness and precision.

Tightly coupled methods are generally recognized for their superior robustness and accuracy over loosely coupled alternatives. In conventional tightly coupled systems, IMU data are processed as inputs within a kinematic model, aiding in the propagation for EKF implementations or pre-integration across a frame's duration. However, this conventional approach may face saturation issues when the robotic motion exceeds the measurement limits of the IMU. In contrast, some research initiatives leverage IMU data to measure angular velocity and linear acceleration, as forecasted by a continuous-time trajectory model. This model then concurrently refines trajectory parameters and LiDAR scan registration factors, effectively counteracting IMU saturation during vigorous movements (Gentil et al.,2019). Our approach resonates with this concept, employing IMU data as output measurements from the model, but innovates by depicting robot motion as a stochastic process intertwined with kinematic dynamics. This unique amalgamation allows for the simultaneous update of the state with LiDAR points within an EKF framework, proficiently managing saturated IMU data during extreme dynamics, like severe vibrations and rapid movements. To the best of our knowledge, this functionality to operate with saturated IMU data within LiDAR-inertial systems is unparalleled in existing studies, signifying a pivotal breakthrough in the domain.

3. METHOD

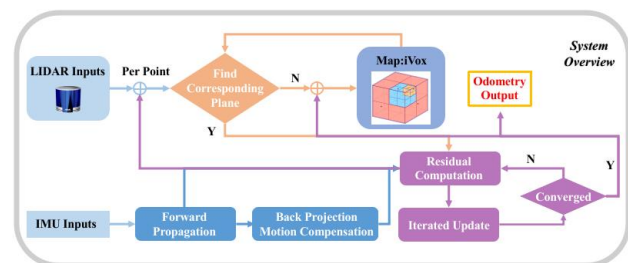


Figure 1. System overview of our approach.

Our system architecture, as depicted in Figure 1, operates on sequentially sampled LiDAR points, updating the state at their respective timestamps, thereby yielding an odometry output at an exceptionally high rate, practically achieving 3-6kHz. Specifically, for each received LiDAR point, we conduct a

search for a corresponding plane in the map. For state estimation purposes, points in the new scan are registered to the map points maintained in a large local map if they match with the planes deduced in the map, utilizing a tightly-coupled iterated Kalman filter framework. The global map points within the large local map are organized by an incremental voxel structure, iVox (Bai et al., 2022). An LRU (Least Recently Used) cache mechanism is implemented in the local map to automatically purge long-unused voxels from iVox, thereby ensuring system efficiency. Thus, iVox tracks all map points within a designated area, which has a specific length and is employed to compute residuals in the state estimation module. The optimized pose eventually registers the points from the new scan into the global coordinate system, integrating them into the map by inserting into iVox at an odometry rate. This integration exemplifies our system's sophisticated mechanism for dynamic map management and state estimation, marking a significant stride in LiDAR-inertial navigation technology.

3.1 State Estimation

LiDAR systems typically operate through point-by-point sampling. Our methodology processes each point directly, rather than between frames, thus obviating the need to estimate the LiDAR pose for each point in the point cloud relative to a specific moment in time. However, to enhance positioning accuracy and system stability, we have adopted the backward propagation method introduced in FAST-LIO, which estimates the LiDAR pose at the end of each scan for every point based on IMU measurements. This approach allows for precise and stable localization by directly correlating each LiDAR point with its respective pose at the scan's conclusion, leveraging the high-resolution temporal data provided by the IMU.

The system nominal state x , error state δx , IMU input u and noise ω are defined as follows:

$$M \triangleq \mathbb{R}^6 \times SO(3) \times \mathbb{R}^9 \quad (1)$$

$$x \triangleq \begin{bmatrix} {}^G p_b^T & {}^G v_b^T & {}^G q_b^T & b_a^T & b_g^T & {}^G g^T \end{bmatrix} \in M \quad (2)$$

$$\delta x \triangleq \begin{bmatrix} {}^G \delta p_b^T & {}^G \delta v_b^T & {}^G \delta \theta_b^T & \delta b_a^T & \delta b_g^T & {}^G \delta g^T \end{bmatrix}^T \in \mathbb{R}^{18} \quad (3)$$

$$u \triangleq \begin{bmatrix} a_m^T & \omega_m^T \end{bmatrix}^T \quad (4)$$

$$\omega \triangleq \begin{bmatrix} n_a^T & n_\omega^T & n_{ba}^T & n_{b\omega}^T \end{bmatrix}^T \quad (5)$$

where ${}^G p_b$, ${}^G v_b$, ${}^G q_b$ are the IMU position, velocity, attitude in the global frame, b_a , b_g are the IMU biases and ${}^G g$ is the gravity in the global frame. a_m , ω_m are IMU measurements. n_a , n_ω are the measurement noise of a_m , ω_m and n_{ba} , $n_{b\omega}$ are the random walk noise of IMU biases.

Using the IMU measurements as input, the nominal state values are predicted using a discrete-time propagation model:

$$x_{k+1} = f(x_k, u_k) \quad (6)$$

with a linearized discrete-time error state transfer equation(7), the covariance matrix of the error state is forward propagated as equation (8).

$$\delta x_{k+1} = (I + F_k \Delta t) \delta x_k + (C_k \Delta t) \omega \quad (7)$$

$$P_{k+1} = (I + F_k \Delta t) P_k (I + F_k \Delta t)^T + (C_k \Delta t) Q_k (C_k \Delta t)^T \quad (8)$$

For the detailed forms of $f(x_k, u_k)$, F_k , C_k and Q_k , readers can refer to FAST-LIO (Xu et al., 2021). In addition, the result of the forward propagation are used for motion compensation and registration. We follow the direct method of FAST-LIO2 by downsampling the sub-frame lidar points and searching the plane for each point, then constructing point-to-plane residuals after fitting the local plane patches:

$$h_j(x_k^K) = {}^G n_j^T \left({}^G q_{b_k}^K \left({}^b q_l^l p_j + {}^b p_l \right) + {}^G p_{b_k}^K - {}^G c_j \right) \quad (9)$$

Where $h_j(x_k^K)$ is the j -th row of $h(x_k^K)$, ${}^b q_l$ and ${}^b p_l$ are the pre-calibrated extrinsic between lidar and IMU, ${}^G c_j$ are normal vector and centroid of the j -th plane patch respectively, ${}^l p_j$ is the corresponding j -th lidar point, means the k -th iteration. Using the point-to-plane residuals as measurements, data association and error state update operations are performed iteratively FAST-LIO2:

$$K = (H^T R^{-1} H + P_k^{-1})^{-1} H^T R^{-1} \quad (10)$$

$$x_k^{K+1} = x_k^K + K(0 - h(x_k^K)) - (I - KH)(x_k^K - x_k^0) \quad (11)$$

During the above iterations, we have simultaneously performed correction of the nominal state and reset of the error state. When the iterations converge, the state covariance is updated with the following equation:

$$P_k = (I - KH) P_k \quad (12)$$

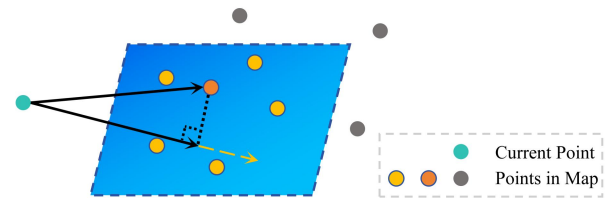


Figure 2. Illustration of direct registration of LiDAR point to map.

3.2 iVox and Mapping

3.2.1 Data Structure of iVox : In iVox, point clouds are initially stored within sparse voxels, which are indexed and hashed into an unordered map via a hash function. Given the sparse nature of LiDAR point clouds, we eschew volumetric representations like TSDF, opting instead for a sparse hashing map that only stores voxels containing at least one point. The hash indices are computable using any spatial hashing algorithm, facilitating efficient and effective storage and retrieval of the point cloud data within the iVox structure, optimizing both memory usage and computational resources.

$$P = [p_x, p_y, p_z]^T, V = \frac{1}{S} [p_x, p_y, p_z]^T \quad (13)$$

$$id_v = hash(v) = (v_x n_x) \text{ xor } (v_y n_y) \text{ xor } (v_z n_z) \text{ mod } N$$

where p_x, p_y, p_z are the coordinates of $P \in \mathbb{R}^3$. The S is the voxel size, n_x, n_y, n_z are large prime numbers and N is the size of hash map, correspondingly.

Within each voxel in iVox, points are stored as a vector, forming the fundamental internal structure. K-Nearest Neighbor (k-NN) searches are conducted within each voxel with a complexity of $O(n)$, where n is the number of points in the voxel. To prevent excessive point insertion into a single voxel during the incremental mapping phase of map construction, a threshold is established. The k-NN search is constrained within a predefined range and unfolds in three stages. Given an iVox structure V and a query point P , the following steps are executed: 1) Identify the voxel index and adjacent voxels, denoted as S . 2) Traverse each voxel in S , conducting a search for up to K nearest neighbors within each voxel. 3) Aggregate the search results and select the optimal K nearest neighbors. The k-NN search within iVox is relatively straightforward and efficient compared to tree-based algorithms.

3.2.2 Incremental Mapping: The incremental mapping in iVox encompasses both the addition and deletion of points. Addition is straightforward; to enhance voxel search efficiency, an octree data structure is integrated within each voxel, facilitating the process through the insertion of new points and the creation of new voxels as required. To prevent excessive point accumulation in a single voxel, a filtering approach akin to VoxelGrid, similar to that employed in FastLIO2, is utilized to skip unnecessary point insertions. Given that the nearest neighbor voxel indices are precomputed, a current point is not inserted if any neighbor is closer to the center of the voxel grid. The leaf size of the filter is a crucial parameter in balancing precision and speed; a larger leaf size prevents too many insertions into a single voxel but can reduce the accuracy of the k-NN search. Drawing from the experiences with Faster-LIO, a leaf size of 0.5 meters typically performs well across most datasets.

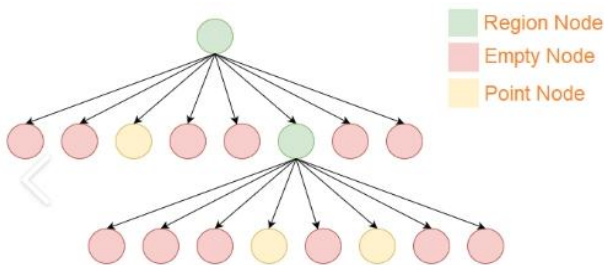


Figure 3. Illustration of octree registration.

Incremental deletion in iVox differs from the k-d tree structure, where the latter traverses the entire map for operations, a process too resource-intensive when applied to an entire voxel hash map. Therefore, instead of actively removing points outside the current field of view, we employ a Least Recently Used (LRU) cache strategy within a time window to maintain the local map. In addition to the voxel hash map, we also keep a queue of recently accessed voxels and establish a maximum size for it. Should the voxel count exceed this limit, we eliminate the oldest voxels in memory. Given that both insertion and deletion operations in the hash map are of $O(1)$ complexity, this approach is sufficiently efficient and accurate for our LIO system, balancing precision and operational speed effectively.

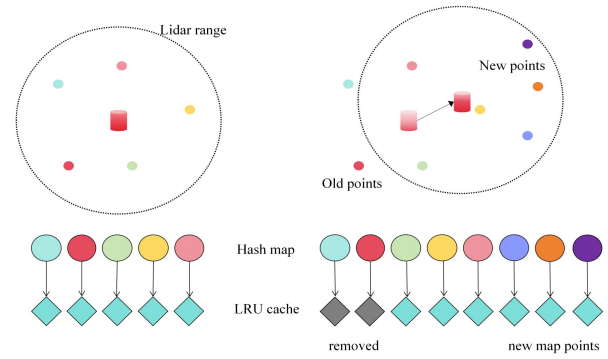


Figure 4. Update the iVox with LRU cache. Relocate the newly created and recently utilized voxels to the back of the cache, and purge those that have remained unused for a predetermined duration.

4. EXPERIMENT

4.1 Platforms

The experimental setup in this study employs the AGILE mobile platform, which is equipped with a nine-axis IMU for precise inertial navigation, four infrared obstacle avoidance sensors, and wheel motors as the propulsion components, achieving a maximum speed of 5 m/s and a load capacity of 50 kg. The control system utilizes a high-performance industrial computer with an i7 CPU and 16G of memory, supporting the ROS operating system, ensuring robust computational power and stability. Additionally, the experimental platform is outfitted with HESAI's Pandar40 LiDAR, capable of 360-degree omnidirectional scanning, high-density point cloud output, and long-range distance measurement, facilitating efficient, accurate, and comprehensive environmental perception.

On the software side, the experimental platform is operational on Ubuntu 20.04 and integrates with the ROS, offering robust software support and an extensive algorithm library for the experiments

4.2 Mapping effect analysis

The mapping results of a park, a teaching building, and a corridor are respectively showcased in Figures 5, 6, and 7. In each figure, we initially present a global view of the final mapping outcome (sub-figure (a)), followed by a focus on specific local areas containing large planes, such as walls (sub-figure (b)). We investigate the consistency of points on the wall (sub-figure (c)), which facilitates a comparison of mapping accuracy between FAST-LIO2 (as seen in (b1) and (c1)) and our method (as seen in (b2) and (c2)). Finally, to illustrate in-frame motion distortion, points from a scan cycle (0.1 seconds) in the selected local areas are displayed (sub-figure (d)), along with a further zoom-in (sub-figure (e)), for both FAST-LIO2 (as (d1) and (e1)) and our method (as (d2) and (e2)), where green points represent the LiDAR points registered in the current scan, and white points denote the mapping results accumulated up to the current scan.

Sub-figure (c) of Figures 5, 6, and 7 reveals that the overall map of FAST-LIO2 (c1) is noticeably thicker than our map (c2). This phenomenon is attributed to the in-frame motion compensation within each individual scan, as depicted in sub-

figures (e) of Figures 5, 6, and 7. It is observed that all red points surrounding the selected wall should belong to the same plane; however, for FAST-LIO2 (e1), these points are dispersed off the wall due to in-frame motion distortion. This distortion is significantly mitigated in our method, demonstrating enhanced performance in maintaining structural consistency in the mapping process.

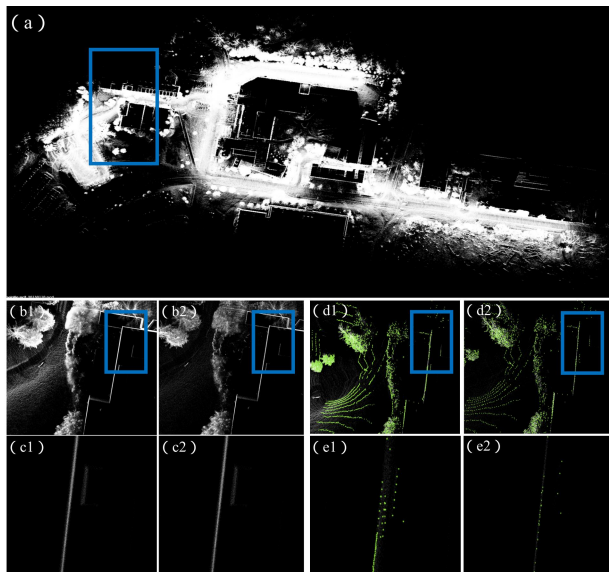


Figure 5. Illustration of motion distortion for the Park.

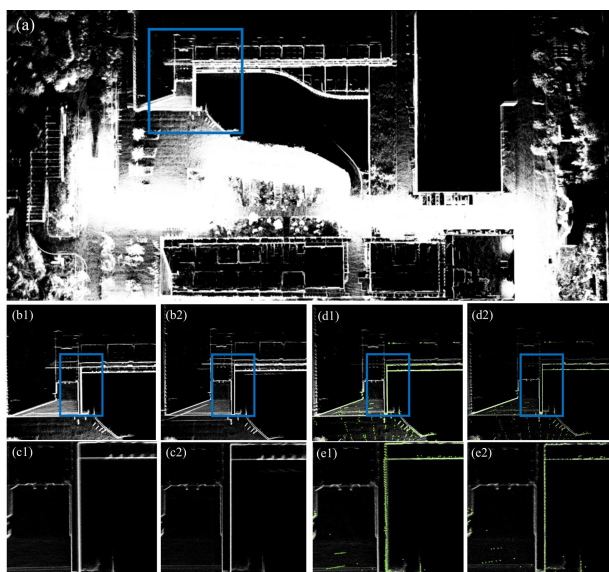


Figure 6. Illustration of motion distortion for the teaching building sequence.

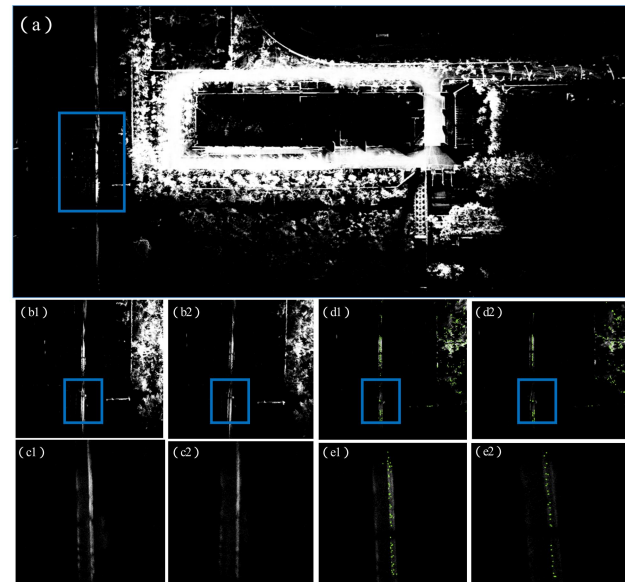


Figure 7. Illustration of motion distortion for the corridor.

FAST-LIO2 utilizes backward propagation based on IMU measurements to project all scanned points to the pose at the end of the scan. In contrast, our method optimizes the estimation of pose and position, discarding the inter-frame processing, a process that can be easily affected by IMU measurement noise, bias estimation errors, and the limited IMU sampling rate. Specifically, sensor vibrations cause rapid changes in acceleration and angular velocity within an IMU sample interval, leading to significant IMU propagation errors under the assumption that angular velocity and acceleration are constant during the sampling interval. In contrast, by integrating LiDAR points at their actual sampling times without accumulating any points, our approach fundamentally eliminates motion distortion, as demonstrated in sub-figures (c1) and (c2).

4.3 Mapping accuracy analysis

In the SLAM algorithm domain, the current standard for accuracy evaluation involves the use of the odometry and SLAM evaluation tool (evo) to determine the trajectory's precision. The primary evaluation metrics include Absolute Pose Error (APE) and Relative Pose Error (RPE). APE analysis compares the pose data computed by the system against the assumed true trajectory based on continuous time data, deriving trajectory discrepancies at each corresponding time point to quantify the trajectory error. Conversely, RPE assesses the root mean square error of the Lie algebra for each pose along the trajectory over continuous time, factoring in both rotational and translational errors of the motion pose. The error metrics for APE and RPE cover various dimensions, encompassing maximum, minimum, average, median error values, standard deviation, root mean square error, and other statistical indices, providing a comprehensive evaluation of the trajectory accuracy.

Regarding the robustness of the laser SLAM system, this study conducted multiple experiments on the gate, street, and hall sequences of the M2DGR dataset to assess the algorithm's robustness across different scenarios. The performance was compared with the well-regarded FAST-LIO2 algorithm, and various error metrics were analyzed. The findings are as follows:

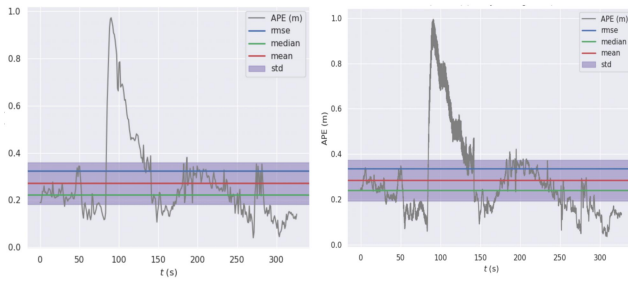


Figure 8. Comparison of APE on the M2DGR dataset's gate_02 sequence, with FAST-LIO2 on the left and our method on the right.

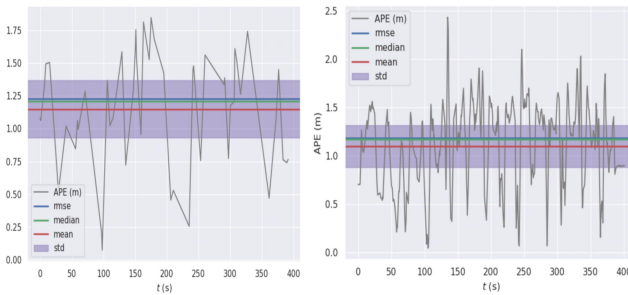


Figure 9. Comparison of APE on the M2DGR dataset's hall_05 sequence, with FAST-LIO2 on the left and our method on the right.

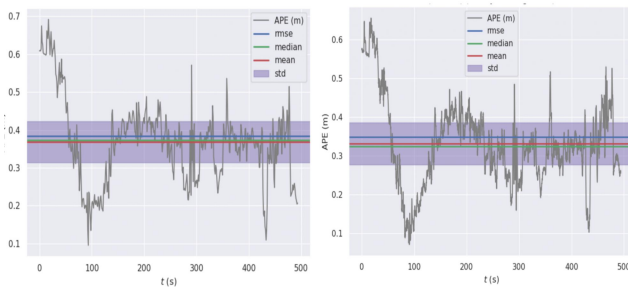


Figure 10. Comparison of APE on the M2DGR dataset's street_06 sequence, with FAST-LIO2 on the left and our method on the right.

We employed seven sequences to evaluate the mapping accuracy of our system. Figures 8, 9, and 10 depict the Absolute Trajectory Error (ATE) comparisons between FAST-LIO2 and our method across different scenarios within these sequences, with our method represented on the right side of each figure. In Figures 8 and 10, the trajectory accuracy margin between the two methods is minimal, illustrating that our point-by-point update strategy facilitates high-frequency state updates. Figure 9, set in a scenario featuring dense rotations and moving objects, suggests that our algorithm may better exploit the high resolution and update rate of LiDAR data, providing a more coherent and detailed environmental representation. This capability allows for a more precise capture of dynamic changes in the environment, thereby demonstrating superior mapping accuracy.

M2DGR sequence	FAST-LIO2	Faster-LIO	ours
gate_02	0.324	0.336	0.352
gate_03	0.216	0.245	0.192
street_03	0.125	0.121	0.111
street_04	0.459	0.588	0.410
street_06	0.326	0.349	0.365
street_08	0.179	0.165	0.170
street_09	1.632	1.538	1.510
hall_05	1.228	1.181	1.112

Table1. Comparison of RMSE (meters) for ours, FAST-LIO2 and Faster-LIO on public sequences.

Table 1 illustrates that our system achieved comparable or superior accuracy relative to other algorithms. Notably, in the hall_05 sequence, where the robot executed numerous rotations, our system enhanced the precision significantly using the point-by-point update strategy and the ESKS framework. However, in the street_06 and gate_02 sequences, the accuracy of our system and Faster-LIO was observed to be lower than that of FAST-LIO2, which can be attributed to the inherent precision advantage of the k-d tree-based LIO method over voxel-based approaches. In these experiments, FAST-LIO2, Faster-LIO, and our system all employed a 0.5-meter voxel filter to process the raw point cloud data. These experimental outcomes underscore the high robustness and precision of our method under dynamic and challenging environmental conditions, demonstrating its potential applicability in the realms of autonomous driving and robotic navigation.

5. CONCLUSION

This paper integrates a robust LiDAR-Inertial Odometry (LIO) framework based on an innovative point-by-point update scheme that refreshes the system state at the actual sampling time of each point, thereby avoiding point accumulation within a frame. This strategy effectively addresses the longstanding issue of in-frame motion distortion and enables high-frequency odometry outputs, facilitating the tracking of rapid movements. Moreover, the integration of map points through the iVox data structure accelerates map searching. Our method has been rigorously tested across various scenes and motion patterns in both public datasets and proprietary data collections, demonstrating excellent mapping results and comparable odometry accuracy against other advanced algorithms, proving its capability in mapping tasks under high-speed motion. Future work aims to develop a reliable mapping method that relies solely on LiDAR, independent of inertial measurement units, to further enhance the robustness and efficiency of the mapping process.

ACKNOWLEDGEMENTS

This research was funded by the National Natural Science Foundation of China (Grant Numbers 42201483), the China Postdoctoral Science Foundation (Grant Numbers 2022M710332) and Funding for Postdoctoral Research Activities in Beijing(2023-zz-140).

REFERENCES

- Zhang, J., Singh, S., 2014. LOAM: Lidar Odometry and Mapping in Real-time. Robotics: Science and Systems.
- Qin, C., Ye, H., Pranata, C. E., Han, J., Zhang, S., Liu, M., 2020. LINS: A Lidar-Inertial State Estimator for Robust and Efficient

Navigation. IEEE International Conference on Robotics and Automation (ICRA), Paris, France, pp. 8899-8906. doi: 10.1109/ICRA40945.2020.9197567.

Xu, W., Cai, Y., He, D., Lin, J., Zhang, F., 2022. FAST-LIO2: Fast Direct LiDAR-Inertial Odometry. IEEE Transactions on Robotics, 38(4), pp. 2053-2073. doi: 10.1109/TRO.2022.3141876.

Karimi, M., Oelsch, M., Stengel, O., Babaian, E., Steinbach, E., 2021. LoLa-SLAM: Low-Latency LiDAR SLAM Using Continuous Scan Slicing. IEEE Robotics and Automation Letters, 6(2), pp. 2248-2255. doi: 10.1109/LRA.2021.3060721.

Qu, C., Shivakumar, S. S., Liu, W., Taylor, C. J., 2022. LLOL: Low-Latency Odometry for Spinning Lidars. International Conference on Robotics and Automation (ICRA), Philadelphia, PA, USA, pp. 4149-4155. doi: 10.1109/ICRA46639.2022.9811605.

Bai, C., Xiao, T., Chen, Y., Wang, H., Zhang, F., Gao, X., 2022. Faster-LIO: Lightweight Tightly Coupled Lidar-Inertial Odometry Using Parallel Sparse Incremental Voxels. IEEE Robotics and Automation Letters, 7(2), pp. 4861-4868. doi: 10.1109/LRA.2022.3152830.

Xu, W., Zhang, F., 2021. FAST-LIO: A Fast, Robust LiDAR-Inertial Odometry Package by Tightly-Coupled Iterated Kalman Filter. IEEE Robotics and Automation Letters, 6(2), pp. 3317-3324. doi: 10.1109/LRA.2021.3064227.

Zhao, S., Fang, Z., Li, H., Scherer, S., 2019. A Robust Laser-Inertial Odometry and Mapping Method for Large-Scale Highway Environments. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Macau, China, pp. 1285-1292. doi: 10.1109/IROS40897.2019.8967880.

Hemann, G., Singh, S., Kaess, M., 2016. Long-range GPS-denied aerial inertial navigation with LIDAR localization. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Daejeon, Korea (South), pp. 1659-1666. doi: 10.1109/IROS.2016.7759267.

Gentil, C. L., Vidal-Calleja, T., Huang, S., 2019. IN2LAMA: INertial Lidar Localisation And MApping. International Conference on Robotics and Automation (ICRA), Montreal, QC, Canada, pp. 6388-6394. doi: 10.1109/ICRA.2019.8794429.

Shan, T., Englot, B., Meyers, D., Wang, W., Ratti, C., Rus, D., 2020. LIO-SAM: Tightly-coupled Lidar Inertial Odometry via Smoothing and Mapping. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Las Vegas, NV, USA, pp. 5135-5142. doi: 10.1109/IROS45743.2020.9341176.

Anderson, S., Barfoot, T. D., 2015. Full STEAM ahead: Exactly sparse gaussian process regression for batch continuous-time trajectory estimation on SE(3). IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Hamburg, Germany, pp. 157-164. doi: 10.1109/IROS.2015.7353368.