IEEE Robotics and Automation Letters (RAL) paper presented at
2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
October 14-18, 2024. Abu Dhabi, UAE

Light-LOAM: A Lightweight LiDAR Odometry and Mapping based on Graph-Matching

Shiquan Yi, Yang Lyu *Member, IEEE*, Lin Hua, Quan Pan *Member, IEEE*, Chunhui Zhao *Member, IEEE*

***Abstract*—Simultaneous Localization and Mapping (SLAM) plays an important role in robot autonomy. Reliability and efficiency are the two most valued features for applying SLAM in robot applications. In this paper, we consider achieving a reliable LiDAR-based SLAM function in computation-limited platforms, such as quadrotor UAVs based on graph-based point cloud association. First, contrary to most works selecting salient features for point cloud registration, we propose a non-conspicuous feature selection strategy for reliability and robustness purposes. Then a two-stage correspondence selection method is used to register the point cloud, which includes a KD-tree-based coarse matching followed by a graph-based matching method that uses geometric consistency to vote out incorrect correspondences. Additionally, we propose an odometry approach where the weight optimizations are guided by vote results from the aforementioned geometric consistency graph. In this way, the optimization of LiDAR odometry rapidly converges and evaluates a fairly accurate transformation resulting in the back-end module efficiently finishing the mapping task. Finally, we evaluate our proposed framework on different datasets and real-world environments. Experiments show that our SLAM system achieves a comparative level or higher level of accuracy with more balanced computation efficiency compared with the mainstream LiDAR-based SLAM solutions.**

***Index Terms*—LiDAR SLAM, data association, odometry.**

I. Introduction

Simultaneously Localization and Mapping (SLAM) is now considered an indispensable module for many mobile robots to achieve autonomous navigation in challenging GNSS-denied environments. To adapt to different missions and platforms, various SLAM frameworks based on different sensors, processing methods, or functional structures have been proposed. Vision-based and LiDAR-based SLAMs are the two most widely researched streamlines according to the sensor types. Vision-based methods provide the most economical and practical localization solutions in many stable environments. On the other hand, LiDAR-based methods are usually preferred in structure-rich environments and are considered to be more robust against light condition changes than vision sensors. There are several mile-stone LiDAR-based works, such as LiDAR Odometry and Mapping (LOAM)[21], lightweight and ground-optimized LiDAR odometry and mapping (LeGO-LOAM)[12] and Cartographer [5].

Disregarding the taxonomy, most SLAM works focus on improving along two axes, which are 1) localization performance, such as accuracy and reliability, and 2) runtime efficiency, such as computation burden and storage requirement. Usually, the two aspects cannot be optimized both on a resource-limited robot platform. Rather, a trade-off between performance and efficiency is typically required to set up a proper SLAM system function in practical robot applications.

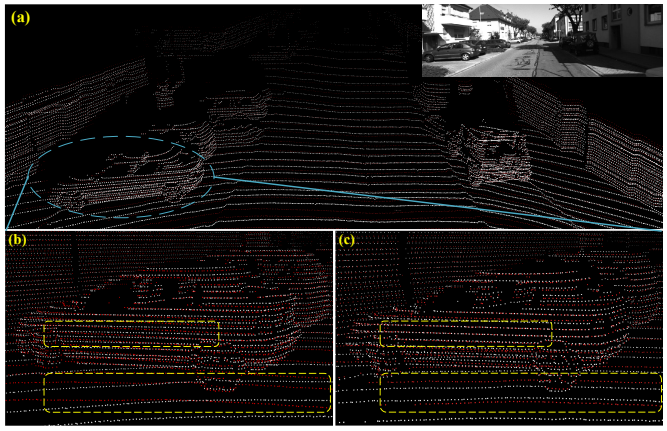


Fig. 1. Feature points alignment between two consecutive scans using different data association methods. (a) A scan of a scenario's point cloud. (b) K-nearest neighbor method. (c) Graph-based two-stage matching method.

In this paper, we develop a lightweight LiDAR-based SLAM with comparative performance to the state-of-the-art methods, but with balanced computation requirements so as to adapt to resource-limited platforms. Contribution are mainly on two folds. First, we develop an innovative SLAM front-end which includes a non-conspicious feature selection strategy and a graph-based feature matching function to achieve better point cloud registration. Fig. 1 is a visual representation illustrating qualitative comparisons in the point cloud alignment task at the front-end. Secondly, to benefit from the reliable registration of the front end, we develop a light back-end that can be executed more efficiently on a computation-limited platform. Experiments validations are carried out with both public datasets and self-collected data. Our implementation of Light-LOAM will be freely available at: https://github.com/BrenYi/Light-LOAM.

II. Related Works

Roughly, the LiDAR-base SLAM function can be divided into two subfunctions, which are the front-end odometry, and the back-end map optimization. In the front-end part, the main purpose is to obtain the incremental transformations between consecutive frames through scan-to-scan or scan-to-map alignment. In the back-end part, an estimator/optimizer that integrates more state constraints is utilized to improve the front-end estimation accuracy and smoothness. In most SLAM

Shiquan Yi, Yang Lyu, Lin Hua, Quan Pan, Chunhui Zhao are with the School of Automation, Northwestern Polytechnical University, Xi'an Shaanxi, 710072 China. This work was supported by the National Natural Science Foundation of China under Grant 62203358, Grant 62233014, and Grant 62073264. Corresponding author: Yang Lyu, email: lyu.yang@nwpu.edu.cn.

Copyright ©2024 IEEE

frameworks, the back-end task consumes more computation resources than the front-end and is usually updated more slowly. With the utilization of the multi-thread technique, the slower thread determines the functional frequency of the whole SLAM function. In a robot control loop, a slower map update rate may affect the robot feedback reaction in unknown environments. Therefore, we considered achieving a more balanced SLAM functional partition with improved performance for practical robot applications. More specifically, we put more computation effort in the front end to improve the point cloud alignment performance, and thanks to a more accurate front end, we can shrink the computation overheads of the back end part.

In the front-end part, there are mainly two streams to achieve the point cloud registration between scans or between scan and map. The first stream is to obtain relative transformation directly from point-to-point correspondences. Iterative closest point (ICP) and its variants, such as the GICP [11], are the most widely used methods to align two-point clouds. ICP-based methods are widely used in early-stage 2D LiDAR-based SLAM frameworks when the size of the point cloud is much more sparse than that of 3D LiDARs. Besides, the Normal distribution transformation (NDT) is considered another direct point cloud registration method. Although it uses all points, the NDT method represents the point cloud with normal distributions and then calculates the relative transformations in a probabilistic fashion. However, the direct methods use all points to optimize the relative transformation, therefore its computation requirement can be a major concern for processing real-time 3D LiDAR scan sequences. The second stream is to extract salient features from point clouds, and then to carry out registration only with the feature points. In the milestone work, LOAM, points are selected based on their curvature and then assigned as planar points and edge points, and then registration is carried out based on the nearest neighborhood methods. Similar geometric features are also used in LeGO-LOAM[12] and F-LOAM[15]. Besides the geometric calculated features, there is also a trend to implement learning-based methods to obtain features represented as deep neural networks due to their power to represent nonlinearity in the descriptors. However, the learning-based methods are often criticized as data-dependent and may not be ready to be used in unknown environments. Although with different representations of the point cloud scans, most methods above use KD-tree-based technology for efficient correspondence indexing. While KD-tree[10] is widely used for establishing initial correspondences in SLAM systems[21], [15], [12], [13], [16], it can introduce erroneous associations, particularly in noisy or occluded environments.

Recent years have seen a notable increase in research efforts directed toward leveraging the principles of geometric consistency and graph theory to tackle the point cloud data association problem. Bailey et al.'s pioneering work [1] introduced the application of geometric consistency in addressing the 2D LiDAR-based map-building problem. This approach involves the construction of a graph, wherein the selection of the correct correspondence set is facilitated through the identification of the maximum common subgraph. In a related vein, Lajoie et al. [7] utilized pairwise consistent measurements to mitigate spurious loop closures within a distributed SLAM system. This methodology similarly involves the maximization of a consistent subgraph, which finds application in the domain of multi-robot map merging, as documented in the literature [9]. Yang et al.[17] introduced the TEASER graph-theoretic framework, which incorporates the truncated least squares optimization method and maximum clique inlier selection technology to effectively eliminate numerous spurious correspondences. Its efficacy has been substantiated in point registration and scan-matching tasks. In the studies [18] and [19], the researchers utilize geometrically consistent graphs in conjunction with a variety of voting strategies to rank correspondences and select dependable inliers. Another graph-theoretic framework development, named Clipper [8], constructs a weighted graph and formulates the inlier association as an optimization problem, ultimately solving for the densest subgraph of consistent correspondences. While these graph-based solutions for association problems have demonstrated effectiveness, particularly in adhering to the geometric consistency constraint, it remains a challenge to optimize their efficiency. Notably, their performance falls short of desired levels when confronted with large-scale point cloud alignment within stringent time constraints, for example, in a LiDAR-based SLAM framework.

In this paper we propose a LiDAR-based SLAM framework with more balanced front-end and back-end based on graph-based matching methods. We describe our method pipeline in Section III. We validate our proposed method extensively in Section IV. We conclude the paper in Section V.

III. LIGHT-LOAM

We present the pipeline of our Light-LOAM SLAM system in Fig. 2, which is composed of three core stages: *pre-processing*, *two-stage feature matching*, and *pose estimation*. In the pre-processing stage, we begin by filtering out disjoint points from each point cloud scan. To select stable corner and plane features with subtle local geometry attributes, we employ a non-conspicuous selection method and filter out the most significant corner and plane features. This is one main difference against other methods[21], [12], [15]. A two-stage feature-matching process is carried out afterward. In the first stage, a KD-Tree-based method[21] is employed to establish initial correspondences for the selected features. Then, we introduce a graph-based consistency voting mechanism to assess these correspondence relationships, effectively filtering out unreliable associations. Moving to the front-end odometry module, the consistent scores of reliable point pairs are leveraged to optimize transformations, resulting in initial, relatively precise pose estimations. Finally, with the support of these initial reliable estimations, the mapping module optimizes more accurate poses in a more efficient fashion.

A. Feature Extraction and Selection

a) Disjoint Point Removal: Given the substantial volume of data produced by 3D LiDAR sensors, feature extraction and feature-based alignment is a widely adopted approach for efficient transformation evaluation. However, it is imperative to

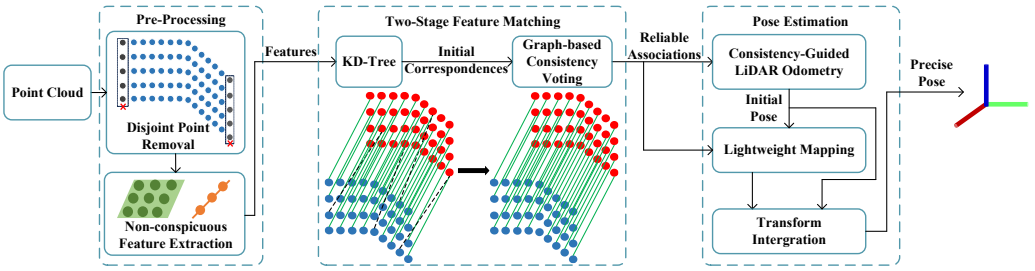
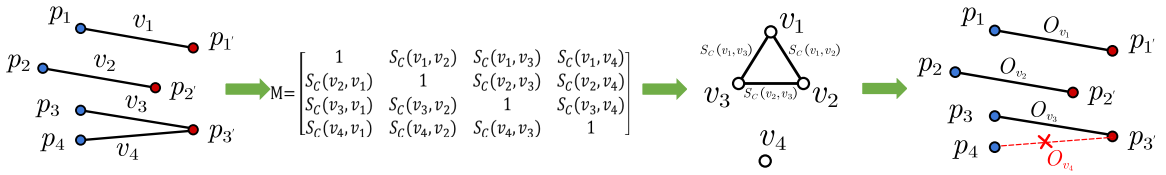


Fig. 2. Overview of Light-LOAM System



(a) Initial Correspondences (b) Affinity Matrix (c) Compatibility Graph (d) Reliable Associations Voting

Fig. 3. Process of Graph-based Two-stage Feature Matching

eliminate disjoint objects before extracting feature candidates. Disjoint points may often represent outliers or segments of occluded objects, and their inclusion can significantly degrade subsequent feature association and pose estimation quality. Therefore, in alignment with prior work [2], we employ the following criteria to exclude these discontinued points:

$$\left|\left\|\mathbf{p}_{i+1}^{k}-\mathbf{p}_{i}^{k}\right\|_{2}-\left\|\mathbf{p}_{i-1}^{k}-\mathbf{p}_{i}^{k}\right\|_{2}\right|>\sigma_{\text{disjoint}}. \tag{1}$$

where $\mathbf{p}_i^k$ represents point i located in the kth laser beam channel, and $\sigma_{disjoint}$ serves as the judgment threshold. A point is classified as disjoint if the absolute difference between its Euclidean distances to neighbors on both sides exceeds $\sigma_{disjoint}$. Otherwise, it is considered a consecutive candidate.

b) Non-conspicuous Feature Extraction and Selection: After eliminating disjoint point, we extract feature points from each laser beam channel. The local geometric attribute of a point is characterized using the smoothness metric (2).

$$r(\mathbf{p}_i^k)=\frac{1}{|\mathcal{S}|\left\|\mathbf{p}_i^k\right\|}\left\|\sum_{j\in\mathcal{S},j\notin i}(\mathbf{p}_i^k-\mathbf{p}_j^k)\right\|. \tag{2}$$

where $\mathcal{S}$ denotes the assessed point set, encompassing the candidate and its adjacent objects on both the left and right sides. The notation $|\mathcal{S}|$ denotes the cardinality of this set. A candidate is designated as a corner feature if its smoothness $r(\mathbf{p}_i^k)$ exceeds the threshold r_t; otherwise, it is identified as a plane feature. Following the approach in [21], our classification process involves assessing 5 points on either side of a candidate, utilizing a default threshold of $r_t = 0.1$ for feature extraction in practical implementation.

In conventional LiDAR-based SLAM systems, such as LOAM[21], FLOAM[15], LEGO-LOAM[12], the perceived space is commonly divided into several subregions. Feature candidates with the highest or lowest smoothness attributes are selected from each subregion for subsequent feature matching. However, our Light-LOAM SLAM system introduces an innovative non-conspicuous feature selection strategy. As mentioned earlier, feature selection is typically guided by a discriminative principle. But, are these discriminative features truly robust and capable of serving as high-quality optimization samples? It's worth noting that some outliers or occluded points can exhibit highly discriminative geometric attributes. Therefore, we hold the view that candidates with weaker smoothness attributes than the top conspicuous ones may be more valuable and robust for data association.

Our priority is to select weaker corner and plane features as optimized candidates within each subregion. Before initiating feature selection process, points are first sorted in descending order based on their smoothness values within each subregion:

$$\mathcal{F}^k=\left\{\mathbf{p}_i^k,\cdots,\mathbf{p}_{i+j}^k|r(\mathbf{p}_i^k)>\cdots>r(\mathbf{p}_{i+j}^k)\right\}. \tag{3}$$

We choose the m sharpest points after the first k points in the ordered set $\mathcal{F}^k$ and designate them as the edge feature set $\mathcal{F}_e^k$. Similarly, we select the n flattest candidates before the last l points and include them in the planar set $\mathcal{F}_s^k$.

$$\mathcal{F}_e^k=\left\{\mathbf{p}_{i+k+1}^k,\cdots,\mathbf{p}_{i+k+m}^k\,|\,\mathbf{p}_x^k\in\mathcal{F}^k,r(\mathbf{p}_x^k)>r_t\right\}. \tag{4}$$

$$\mathcal{F}_s^k=\left\{\mathbf{p}_{i+j-l-n}^k,\cdots,\mathbf{p}_{i+j-l-1}^k\,|\,\mathbf{p}_x^k\in\mathcal{F}^k,r(\mathbf{p}_x^k)<r_t\right\}. \tag{5}$$

In our implementation, we horizontally divide each laser beam channel into 6 subregions. Excessive feature rejection in $\mathcal{F}^k$ limits optimization samples, affecting pose estimation precision in SLAM. Through multiple experiments, setting $k = 1$ and $l = 2$ proves to be appropriate. Similar to [21], the number of selected edge and planar features, denoted as m and n, respectively, are set to 2 and 4.

This non-conspicuous feature selection method, combined with the disjoint point removal preprocessing step, efficiently filters out more outliers, ensuring a more reliable set of candidates for subsequent pose estimation.

B. Graph-based Two-stage Feature Matching

Identifying corresponding features from both the last scan of the point cloud and the existing constructed map is a fundamental prerequisite for subsequent scan-to-scan and scan-to-map alignments. KD-tree[10] is a widely used method for establishing correspondence relationships due to its efficiency and effectiveness, as evidenced by its adoption in various works[21], [12], [15]. Despite its prevalence, KD-tree

is susceptible to errors caused by environmental occlusions, outliers, and noise within the point cloud, leading to inaccurate pose estimations. For instance, as shown in Fig. 4, situations can arise where more than one candidate feature from the current scan matches the same point from the last scan or map as its closest counterpart, leading to false multi-to-one correspondence cases. To mitigate such issues and reduce spurious correspondences, we introduce a novel graph-based two-stage correspondence selection method.

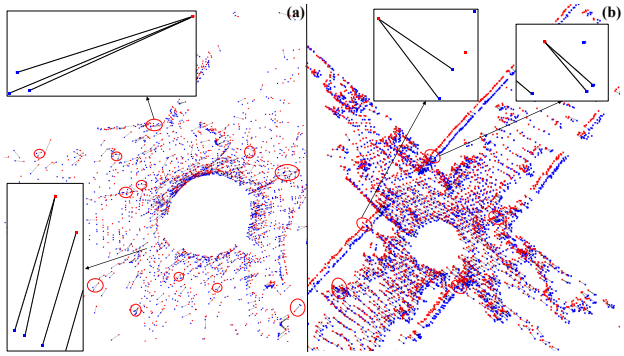


Fig. 4. The demonstration of initial feature correspondences generated by KD-tree. In each correspondence, the red point is the source feature from the current scan, and the blue one is its corresponding target object from the last scan or map. Ellipses indicate incorrect data associations. (a)Illustration for initial corresponding relationships of the front-end. (b)Illustration for initial corresponding relationships of the back-end.

a) Initial Correspondences Determination by KD-tree: We start by using KD-tree to find correspondences for our feature candidates, assuming the closest point from the last scan or map is the true correspondence for each feature. This establishes our initial set of point pairs using the formula (6).

$$\mathcal{W} \triangleq \left\{ \left(\bar{\mathbf{p}}_i^k, \bar{\mathbf{p}}_{i'}^t\right) | \, \bar{\mathbf{p}}_i^k \in \mathcal{F}_e^k \cup \mathcal{F}_s^k, \bar{\mathbf{p}}_{i'}^t \in \mathcal{F}_{e'} \cup \mathcal{F}_{s'} \right\}. \tag{6}$$

where $\bar{\mathbf{p}}_i^k$ represents source features from the edge set $\mathcal{F}_e^k$ or the planar group $\mathcal{F}_s^k$ of the current scan, and the target feature point $\bar{\mathbf{p}}_{i'}^t$ corresponds to their closest corresponding point from the target feature sets $\mathcal{F}_{e'}$ and $\mathcal{F}_{s'}$. At the front end, these target sets comprise the last scan of feature sets $\mathcal{F}_e^{k-1}$ and $\mathcal{F}_s^{k-1}$, while in the mapping stage, they represent the feature maps $\mathcal{F}_e^W$ and $\mathcal{F}_s^W$.

b) Reliable Associations via Consistent Graph: In the second stage, a graph-based correspondence validation algorithm is introduced. Beginning with the initial putative associations $\mathcal{W}$ from the KD-tree, a compatibility graph is constructed based on the principle of geometric consistency.

Before delving into the concept of geometric consistency, let's assume the existence of two correct associations, $\left(\bar{\mathbf{p}}_i^k, \bar{\mathbf{p}}_{i'}^t\right)$ and $\left(\bar{\mathbf{p}}_j^k, \bar{\mathbf{p}}_{j'}^t\right)$, which share an identical set of transformation parameters denoted as $(\mathbf{R}, \mathbf{T})$. These two pairs of associations can be formulated as:

$$\bar{\mathbf{p}}_{i'}^t = \mathbf{R}\bar{\mathbf{p}}_i^k + \mathbf{T}. \tag{7}$$

$$\bar{\mathbf{p}}_{j'}^t = \mathbf{R}\bar{\mathbf{p}}_j^k + \mathbf{T}. \tag{8}$$

Theoretically, the Euclidean distance between two target points remains constant across different frames, as expressed by (9), embodying what we refer to as geometric consistency.

$$\left\|\bar{\mathbf{p}}_{i'}^t - \bar{\mathbf{p}}_{j'}^t\right\|_2 = \left\|\bar{\mathbf{p}}_i^k - \bar{\mathbf{p}}_j^k\right\|_2. \tag{9}$$

We can leverage this constraint to evaluate the compatibility of correspondences within a graph space, rather than the Euclidean space. To illustrate this conveniently, let's assume there are four hypothetical association cases generated from the KD-tree, as depicted in Fig. 3 (a). We can construct a compatibility graph, where each vertex, denoted as $\mathbf{v_i} \triangleq (\bar{\mathbf{p}}_i^k, \bar{\mathbf{p}}_{i'}^t)$, represents the ith association relationship. The edges in the graph indicate that the two associations, v_i and v_j, are compatible or geometrically consistent. Within the graph displayed in Fig. 3 (c), there exist four associations: v_1, v_2, v_3, and v_4. Notably, v_1, v_2, and v_3 are mutually geometrically consistent.

Following this, as depicted in Fig. 3, the compatibility graph is constructed using an affinity matrix M. Each entry $\mathbf{M}(i,j)$ in this matrix represents the geometric consistency score of the correspondence pair (v_i, v_j) and is calculated quantitatively as:

$$S_c(v_i, v_j) \triangleq \exp(-\frac{d(v_i, v_j)^2}{\sigma^2}). \tag{10}$$

Here, the term $d(v_i, v_j)$ is defined as:

$$d(v_i, v_j) \triangleq \left\|\bar{\mathbf{p}}_{i'}^t - \bar{\mathbf{p}}_{j'}^t\right\|_2 - \left\|\bar{\mathbf{p}}_i^k - \bar{\mathbf{p}}_j^k\right\|_2. \tag{11}$$

where σ serves as a distance adjustment parameter. Notably, S_c ranges from 0 to 1, achieving 1 for perfect geometric consistency. The diagonal entry $\mathbf{M}(i,i)$ consistently equals 1. A lower score indicates a higher degree of inconsistency.

Based on the compatibility graph, we employ a voting rule to assess the quality of each correspondence. The voting mechanism, formulated as in Equation (12), facilitates the computation of the voting score o_i for each association, providing a means to assess its consistency.

$$o_i = o(v_i) \triangleq \sum_{j=0, j\neq i}^{|\mathcal{W}|} \left\lfloor \frac{S_c(v_i, v_j)}{\eta} \right\rfloor. \tag{12}$$

where η functions as the voting threshold that determines the compatibility of two associations in the voting process. Additionally, $|\mathcal{W}|$ denotes the cardinality of the correspondence set $\mathcal{W}$. In this scheme, an association subject v_i receives one vote if its consistency score $S_c(v_i, v_j)$ with association v_j meets or exceeds the threshold η. The final level of consistency in the voting process is determined by all consistent voters.

Following the completion of our voting pipeline, we can represent the sequence of voting results in descending order as follows:

$$\mathcal{O} = \{o_i \mid o_1 > o_2 > \cdots > o_i, i \in [1, |\mathcal{W}|]\}. \tag{13}$$

In the event that a correspondence candidate v_i receives a voting score o_i lower than $x\%$ of the total number of association candidates, it is considered an unreliable association and is subsequently filtered out. After removing these outlier associations, we obtain the final set of reliable associations $\mathcal{O}'$ along with their corresponding scores $\mathcal{W}'$:

$$\mathcal{O}' = \{o_i \mid o_i \in \mathcal{O}, o_i > x\,|\mathcal{O}|\}. \tag{14}$$

$$\mathcal{W}' = \{v_i \mid v_i \in \mathcal{W}, o(v_i) \in \mathcal{O}'\}. \tag{15}$$

In analyzing the computational complexity of our graph-based matching algorithm with N correspondences, the construction of the compatibility graph and the correspondence ranking using quicksort have time complexities of $O(N^2)$ and $O(N \log N)$, respectively. This results in a total time complexity of $O(N^2+N \log N)=O(N^2)$ for our graph-based voting algorithm. To maintain real-time performance, we partition the perceptual space into n subregions. Correspondence relationships within each subregion form subgraphs, handling tasks in both odometry and mapping stages. In the odometry stage, each subregion processes around 200 correspondences, averaging a total processing time of approximately 3 ms. During the mapping stage, each subregion deals with roughly 350 correspondences, with an average total processing time of about 7 ms. This correspondence division across subregions ensures both real-time performance and accurate results.

C. Consistency-Guided LiDAR Odometry

In LiDAR SLAM systems, odometry is pivotal for refining initial poses through scan-to-scan point cloud matching. The odometry module usually provides high-frequency but somewhat imprecise pose estimations, acting as an initial input for the mapping module. More accurate initial transformations estimated by the odometry module accelerate the convergence of the final robot pose estimation, resulting in a reduction in the computational cost of the mapping back-end.In light of this, we propose a novel LiDAR odometry mechanism in which pose optimization is guided by the voting results from the compatibility graph.

In our odometry module, we aim to optimize the transformation $\mathbf{T}_k^{k-1} \in SE(3)$ representing the motion from the kth frame to the $(k-1)$th frame, and update the global pose $\mathbf{T}_k^W \in SE(3)$ of the point cloud in the k-th frame. Prior to optimization, we correct motion distortions in the point cloud by assuming uniform motion. Similar to LOAM[21], we define two types of residual terms. The first term is the point-to-line distance residual, given by:

$$f_e^L(v_i) = \frac{\left|(\tilde{\mathbf{p}}_{(i,k)}^L - \bar{\mathbf{p}}_{(i',k-1)}^L) \times (\tilde{\mathbf{p}}_{(i,k)}^L - \bar{\mathbf{p}}_{(j',k-1)}^L)\right|}{\left|\bar{\mathbf{p}}_{(i',k-1)}^L - \bar{\mathbf{p}}_{(j',k-1)}^L\right|}. \quad (16)$$

In (16), $\tilde{\mathbf{p}}_{(i,k)}^L = \mathbf{T}_k^{k-1}\bar{\mathbf{p}}_{(i,k)}^L$, where $\bar{\mathbf{p}}_{(i,k)}^L \in \mathcal{F}_e^k$ represents a corner feature. $\bar{\mathbf{p}}_{(i',k-1)}^L \in \mathcal{F}_e^{k-1}$ represents the closest object from the same laser beam channel to $\tilde{\mathbf{p}}_{(i,k)}^L$. The point-pair $(\bar{\mathbf{p}}_{(i,k)}^L, \bar{\mathbf{p}}_{(i',k-1)}^L) \in \mathcal{W}'$ signifies a reliable correspondence identified through our graph-based two-stage feature matching. Additionally, $\bar{\mathbf{p}}_{(j',k-1)}^L \in \mathcal{F}_e^{k-1}$ is another nearest neighbor of $\tilde{\mathbf{p}}_{(i,k)}^L$ from different laser channel. $\bar{\mathbf{p}}_{(i',k-1)}^L$ and $\bar{\mathbf{p}}_{(j',k-1)}^L$ together form a line geometric residual term.

For the planar residual, we have:

$$f_s^L(v_i) = \mathbf{n_s} \cdot (\tilde{\mathbf{p}}_{(i,k)}^L - \bar{\mathbf{p}}_{(i',k-1)}^L). \quad (17)$$

where

$$\mathbf{n_s} = \frac{(\bar{\mathbf{p}}_{(i',k-1)}^L - \bar{\mathbf{p}}_{(j',k-1)}^L) \times (\bar{\mathbf{p}}_{(i',k-1)}^L - \bar{\mathbf{p}}_{(l',k-1)}^L)}{\left|(\bar{\mathbf{p}}_{(i',k-1)}^L - \bar{\mathbf{p}}_{(j',k-1)}^L) \times (\bar{\mathbf{p}}_{(i',k-1)}^L - \bar{\mathbf{p}}_{(l',k-1)}^L)\right|}. \quad (18)$$

In (17), $\tilde{\mathbf{p}}_{(i,k)}^L = \mathbf{T}_k^{k-1}\bar{\mathbf{p}}_{i,k}^L$ where $\bar{\mathbf{p}}_{i,k}^L \in \mathcal{F}_s^k$ is a planar feature. $\bar{\mathbf{p}}_{(i',k-1)}^L$ and $\bar{\mathbf{p}}_{(j',k-1)}^L$ are the first and second closest objects to the projected feature $\tilde{\mathbf{p}}_{(i,k)}^L$ from the same laser beam channel. Additionally, $\bar{\mathbf{p}}_{(l',k-1)}^L$ is the other nearest point from a different channel. These three points from the $(k-1)$th scan collectively construct a plane to establish a point-to-plane residual. Certainly, $(\bar{\mathbf{p}}_{i,k}^L, \bar{\mathbf{p}}_{(i',k-1)}^L) \in \mathcal{W}'$ is also a valid association obtained through the two-stage feature matching process.

In section II.B, we introduced the compatibility graph to filter out unreliable associations, mitigating issues that could degrade optimization. Each correspondence is associated with a consistent voting score indicating its level of reliability. Consequently, we understand the potential positive contribution of each association. Leveraging the voting results from the two-stage feature matching process, we assign higher weights to these more reliable associations. The process of designing custom weights can be formulated as follows:

$$W_i = \begin{cases} \alpha \cdot \frac{o_i - o_{\min}}{o_{\max} - o_{\min}}, i \in [0, \lambda\,|\mathcal{O}'|] \\ 1, i \in (\lambda\,|\mathcal{O}'|, |\mathcal{O}'|] \end{cases} . \quad (19)$$

The associations in the top $\lambda\%$ of the ordered set $\mathcal{O}'$ receive custom optimization weights. o_{min} and o_{max} represent the minimum and maximum scores in the set $\mathcal{O}'$, and α is a scale factor. Additionally, the weights for the remaining associations remain unaltered.

Finally, the pose is estimated by minimizing the total of weighted residual terms:

$$\min_{\mathbf{T}_k^{k-1}} \sum W_i f_e^L(v_i) + \sum W_j f_s^L(v_j). \quad (20)$$

With the aid of cost function (20), we can determine the pose $\mathbf{T}_k^{k-1}$ using the Levenberg-Marquardt method[4] as follows:

$$\mathbf{T}_k^{k-1} \leftarrow \mathbf{T}_k^{k-1} - (\mathbf{J}^T\mathbf{J} + \lambda \text{diag}(\mathbf{J}^T\mathbf{J}))^{-1}\mathbf{J}^T f. \quad (21)$$

In the optimization process, where λ is the Lagrange multiplier, J is the Jacobian matrix, and f is the residual vector term, equation (21) iterates to minimize the cost function (20) and obtain the transformation $\mathbf{T}_k^{k-1}$. The residual terms linked to these highly reliable correspondences, with increased weights, play a predominant role during optimization. As a result, the gradient descent direction and parameter updates are predominantly guided by these high-quality associations, leading to a more accurate and closer convergence to the ground truth in pose estimation.

D. Lightweight LiDAR Mapping

The mapping module, typically the back-end, handles precise global pose estimation but at a lower frequency. However, we now present a streamlined mapping module that balances accuracy and efficiency.

During the two-stage feature matching in the mapping module, a KD-tree identifies the neighborhood set C, consisting of the five nearest objects to each feature $\bar{\mathbf{p}}_i^k$ from the mapped cloud. In the second graph-based stage, we assign the centroid $\bar{\mathbf{p}}_{i',m}^W$ of C as the temporary corresponding object of

feature $\bar{\mathbf{p}}_i^k$, establishing feature correspondence relationships $(\bar{\mathbf{p}}_i^k, \bar{\mathbf{p}}_{i',m}^W)$. These relationships are utilized to construct the compatibility graph and perform voting. While the compatibility graph is used to remove unreliable correspondences during mapping, the voting results are not utilized to weigh the importance of each association.

In the optimization process of the mapping module, two types of residual terms are constructed: a point-to-line residual term and a point-to-plane residual term. The residual terms $f_e^M(v_i)$ and $f_s^M(v_j)$ are formulated using the same equations as (16) and (17), respectively, which aligns with the LOAM-based solution[21]. The cost function is defined as:

$$\min_{\mathbf{T}_k^M} \sum f_e^M(v_i) + \sum f_s^M(v_j). \tag{22}$$

In this optimization process, we estimate the global pose $\mathbf{T}_k^M$ using the Levenberg-Marquardt method[4] without any preferential treatment among the residual terms.

Thanks to the removal of unreliable associations and the guidance from the two-stage feature matching, odometry poses converge quickly. This provides a more accurate initial transformation estimation for the mapping module, resulting in a faster and more precise global pose calculation. The corresponding results are presented in section IV. B.

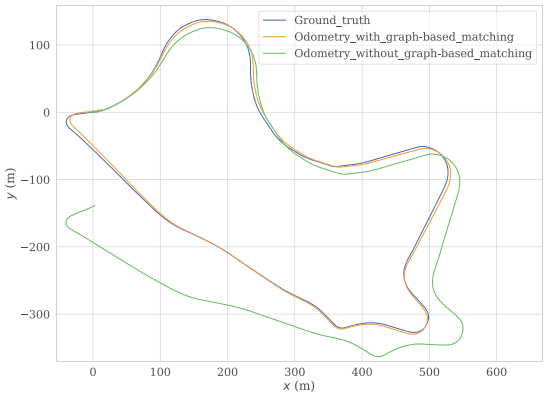


Fig. 5. Trajectories estimated by the different front-end odometry modules for the KITTI 09 sequence.

IV. Experiments

A. Experiment Setup

To evaluate the performance of our Light-LOAM SLAM system, we conducted a series of experiments using the numerous datasets and real-world environments. We extensively validated our system through rigorous assessments and ablation studies using two diverse datasets: KITTI[3] and M2DGR[20]. The KITTI dataset was utilized for evaluating the system's performance in outdoor environments. Additionally, the M2DGR dataset was instrumental in assessing the system's robustness in challenging indoor scenarios, including navigating random movements in the hall (Hall01-Hall03), entering and exiting an lift (Lift01), and other situations. The experiments are conducted on a laptop with an octa-core 3.3GHz processor and 16GB of memory. Real-world testing is carried out on a UAV equipped with an Ouster OS1-32-U LiDAR sensor and a DJI Manifold-2G companion computer featuring an Arm Cortex-A57 CPU and 8GB of memory.

To quantitatively evaluate the accuracy of our SLAM system and facilitate comparison with other approaches, we employed the Absolute Trajectory Error (ATE) metric [22]. The ATE metric measures the disparity between the estimated poses generated by our system and the ground-truth pose values.

B. Experiment on Public Datasets

In this section, we conducted ablation studies and validated our Light-LOAM system. To assess its performance, we compared Light-LOAM with state-of-the-art LiDAR-based systems: LOAM[21], LeGO-LOAM[12], HDL-Graph-SLAM[6], and KISS-ICP[14]. Notably, for comparison, we used A-LOAM[1], an updated version of the LOAM system, rather than the older LOAM version.

1) Ablation Study: To assess the impact of our advanced algorithm on the Light-LOAM SLAM system, we conducted ablation studies on the non-conspicuous feature selection algorithm and the graph-based two-stage feature matching method.

When evaluating the non-conspicuous feature selection method, Our(a) can also be viewed as the LOAM system with the inconspicuous feature selection. The results evaluated on the KITTI and M2DGR datasets are presented in Table I and able IV, respectively. These results clearly demonstrate that the non-conspicuous feature selection method improves the pose estimation accuracy of the Light-LOAM system. This selection strategy notably improves the system's capability to choose more reliable feature samples.

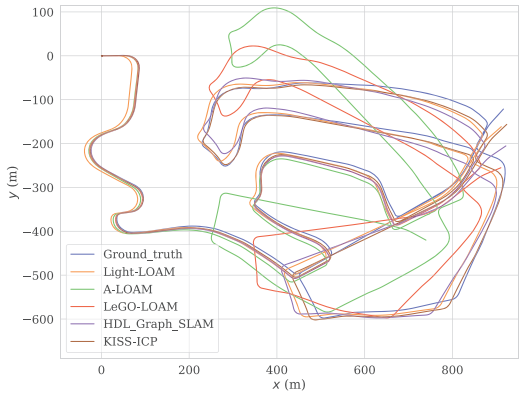


Fig. 6. Ground truth and estimated trajectories for the KITTI 02 sequence.

Our(b) represents the Light-LOAM version with only the two-stage graph-based feature matching module. Its localization results on the two datasets aforementioned are presented in Table I and Table IV respectively. These results indicate a substantial improvement in the accuracy of pose estimation. Additionally, We evaluate our graph-based matching method and non-conspicuous feature selection by comparing them with the LOAM system in the front-end odometry stage. In this experiment, Light-LOAM and LOAM differ solely in the integration of our graph-based feature matching module or the application of non-conspicuous feature selection. The results from Table II and Table V demonstrate improved pose estimation accuracy using the graph-based feature matching method or the non-conspicuous feature selection module in odometry estimation. Furthermore, the trajectory result for

[1]https://github.com/HKUST-Aerial-Robotics/A-LOAM

TABLE I
RMSE ATE ON PUBLIC KITTI ODOMTERY DATASET

Sequence	00	01	02	03	04	05	06	07	08	09	10	Average
A-LOAM	2.624/ 0.023	19.795/ 0.076	117.323/ 0.374	0.851/ 0.025	**0.406/ 0.013**	2.561/ 0.019	0.762/ 0.014	0.590/ 0.009	3.569/ 0.026	**1.451**/ 0.018	1.301/ 0.019	13.748/ 0.056
LeGO-LOAM	6.339/ 0.021	199.682/ 0.197	58.276/ 0.183	0.991/ 0.026	0.549/ 0.017	2.586/ 0.015	0.960/ 0.014	1.110/ 0.018	3.993/ 0.026	2.183/ 0.021	2.397/ 0.023	25.370/ 0.051
HDL-Graph-SLAM	6.213/ 0.045	419.680/ 0.229	19.872/ 0.088	1.098/ 0.045	27.442/ 0.027	3.052/ 0.034	1.321/ 0.019	0.662/ 0.017	4.304/ 0.041	3.008/ 0.037	2.937/ 0.036	44.508/ 0.056
KISS-ICP	6.017/ 0.020	31.232/ **0.025**	**7.769/ 0.026**	3.357/ **0.011**	1.086/ 0.601	**1.936/ 0.011**	0.880/ **0.006**	0.758/ **0.007**	4.997/ **0.017**	3.725/ **0.015**	2.304/ **0.009**	5.824/ 0.068
Ours(a)	2.401/ 0.020	18.980/ 0.074	101.250/ 0.376	**0.804**/ 0.025	0.417/ 0.017	2.273/ 0.018	0.718/ 0.013	0.589/ 0.009	3.479/ 0.026	1.448/ 0.019	1.164/ 0.021	12.138/ 0.056
Ours(b)	2.581/ 0.020	19.593/ 0.074	10.984/ 0.041	0.895/ 0.051	0.456/ 0.017	2.376/ 0.020	0.701/ 0.013	**0.573**/ 0.009	3.496/ 0.028	1.285/ 0.019	1.253/ 0.020	4.018/ 0.028
Ours	**2.315/ 0.018**	**18.854**/ 0.074	10.528/ 0.042	0.849/ 0.024	0.411/ 0.016	2.190/ 0.015	**0.659**/ 0.013	0.587/ 0.009	**3.473**/ 0.029	1.482/ 0.019	**1.104**/ 0.019	**3.859/ 0.025**

Note that Ours(a) refers to our Light-LOAM version only equipped with the non-conspicuous feature selection strategy. Ours(b) refers to the Light-LOAM version only equipped with the two-stage feature matching module. Ours refers to the version with all modules. The ATE metric is typically represented in the 'A/B' format, where 'A' denotes the error in translation (measured in meters), and 'B' represents the error in rotation (measured in radians).

TABLE II
THE COMPARISONS OF RMSE ATE ESTIMATED BY FRONT-END ODOMETRY MODULE ON KITTI DATASET

Sequence	00	01	02	03	04	05	06	07	08	09	10	Average
Odometry(a)	41.661/ 0.179	59.887/ 0.122	97.972/ 0.317	3.773/ 0.054	**1.263/ 0.031**	24.339/ 0.175	1.941/ 0.022	4.918/ 0.052	46.183/ 0.218	40.706/ 0.114	6.667/ 0.074	29.937/ 0.123
Odometry(b)	**11.250/ 0.063**	**29.381/ 0.103**	**21.238/ 0.116**	**1.926**/ 0.064	1.553/ 0.061	**5.056**/ 0.071	2.635/ 0.045	**2.720/ 0.039**	**9.134/ 0.098**	**3.596/ 0.058**	**3.407/ 0.051**	**8.354/ 0.081**
Odometry(c)	37.130/ 0.172	39.140/ 0.114	96.810/ 0.315	3.061/ **0.051**	3.646/ 0.077	18.887/ **0.149**	**1.819/ 0.021**	5.049/ 0.063	37.686/ 0.204	34.375/ 0.129	5.218/ 0.069	25.711/ 0.124

Note that Odometry(a) refers to A-LOAM's front-end, while Odometry(b) and Odometry(c) denote Light-LOAM's front-end with only the two-stage feature matching module and non-conspicuous feature selection, respectively..

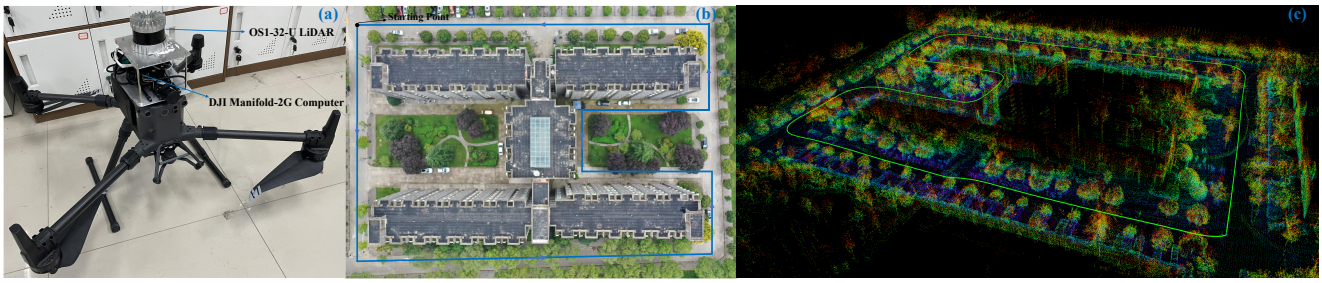


Fig. 7. Light-LOAM system in a real-world environment. (a) DJI M300 UAV for experiment. (b) The campus building. (c) Mapping result of Light-LOAM.

TABLE III
AVERAGE RUNTIME OF EACH MODULE FOR ONE SCAN(MS)

Method	Pre-Pocessing	Odometry	Mapping
A-LOAM	**17.9**	33.87	99.5
LeGO-LOAM	32.3	17.0	82.8
HDL-Graph-SLAM	21.5	**16.8**	168.2
Ours	23.7	43.5	**65.8**
KISS-ICP	-	17.6	-

TABLE IV
RMSE ATE ON PUBLIC M2DGR DATASET(METER)

Sequence	Hall01	Hall02	Hall03	Room01	Door01	Lift01	Average
A-LOAM	0.249	0.302	0.537	1.367	0.287	0.157	0.483
LeGO-LOAM	6.382	33.597	5.389	**0.638**	2.884	0.397	8.215
HDL-Graph-SLAM	0.266	**0.286**	**0.331**	1.296	0.334	0.191	0.451
KISS-ICP	**0.242**	0.503	0.455	5.423	0.327	0.215	1.194
Ours(a)	0.249	0.299	0.475	1.314	0.257	0.154	0.458
Ours(b)	0.250	0.299	0.418	1.281	0.282	0.153	0.447
Ours	0.249	0.300	0.405	1.284	**0.280**	**0.151**	**0.445**

Note that indoor sequences in M2DGR don't provide ground truth for the rotation part of the pose.

sequence 09 depicted in Fig. 5 highlights the effectiveness of the graph-based feature matching mechanism in enhancing the accuracy of estimated poses by the front-end. Our graph-based feature matching approach efficiently filters incorrect correspondences, speeding up pose optimization convergence. In essence, this method shows promise in achieving accurate pose estimation with a limited number of point-pairs, guided by our graph-based feature matching strategy.

2) Validation of Light-LOAM: In this subsection, we evaluate the performance of our Light-LOAM SLAM system in terms of accuracy and efficiency. We start by comparing it with state-of-the-art solutions, as shown in Table I and Table IV. The results clearly indicate that our Light-LOAM system consistently outperforms others in terms of average performance. Even in challenging scenarios, such as sequences 01 and 02 of the KITTI dataset, Light-LOAM maintains its strong performance. For a visual representation of these results, see Fig. 6, demonstrating the strong estimation performance of our Light-LOAM. These findings highlight Light-LOAM's robustness and effectiveness, particularly in challenging environments where other SLAM algorithms face difficulty in accurate pose estimations.

In addition to assessing accuracy, we have also evaluated the efficiency of our Light-LOAM system. Table III provides an overview of the average runtime results of all datasets. Additionally, it's worth noting that KISS-ICP is a point-to-point odometry framework devoid of additional preprocessing or mapping modules. In comparison to other methods' back-ends, ours operates with significantly lower computational demands. The mapping module in our SLAM system is lightweight. This improvement in efficiency is attributed to the more precise initial poses provided by consistency-guided odometry. These findings highlight our system's ability to efficiently maintain a global map. Light-LOAM can take a well-balanced compromise between computational cost and localization accuracy.

TABLE V
THE COMPARISON OF RMSE ATE ESTIMATED BY FRONT-END ODOMETRY MODULE ON M2DGR DATASET(METER)

Sequence	Hall01	Hall02	Hall03	Room01	Door01	Lift01	Average
Odometry(a)	3.133	1.222	1.210	**2.977**	7.387	0.448	2.746
Odometry(b)	**0.704**	**0.544**	**0.977**	3.734	**0.282**	**0.129**	**1.062**
Odometry(c)	2.810	0.981	1.217	3.060	6.987	0.438	2.582

C. Experiment on Realworld Environment

In this section, we assess the localization and mapping performance of Light-LOAM in a real-world dormitory building using a DJI M300 UAV, depicted in Fig. 7(a). The UAV circled the building at 1.2 m/s and returned to the starting point to check for closed-loop trajectory formation. For comparison purposes, we also conducted the same localization tasks using state-of-the-art solutions on the same device. We utilized

TABLE VI
RMSE ATE OF REALWORLD SCENARIO

Method	A-LOAM	LeGO-LOAM	HDL-Graph-SLAM	KISS-ICP	Ours
ATE(m/rad)	1.246/ 0.023	1.323/ 0.028	1.259/ 0.024	1.201/ 0.020	**0.991/ 0.017**

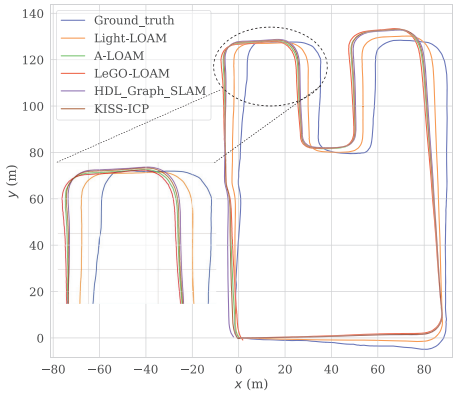


Fig. 8. Trajectories estimated by different systems in the real-world environment.

the results from the RTK-GPS(Real-Time Kinematic Global Positioning System) module on the DJI M300 as our ground truth for comparison. The estimated trajectories are visualized in Fig. 8, and the results are summarized in Table VI. Notably, the ATE of Light-LOAM is 0.991/0.017, which is lower than its competitors. The outcome of the localization and mapping performance is presented in Fig. 7, demonstrating that our mapping results exhibit high accuracy and are capable of completing loop closures without the need for additional loop closure techniques in real-world environments. These results underscore the effectiveness and precision of our Light-LOAM system in real-world scenarios.

V. CONCLUSION AND FUTURE WORK

In this letter, we introduce a lightweight LiDAR SLAM system, Light-LOAM, that employs graph-based matching techniques for efficient and accurate pose estimation. Departing from traditional LOAM-based approaches, we propose a non-conspicuous feature extraction strategy for obtaining stable features. Our graph-based two-stage feature matching method assesses the consistency of associations, filtering unreliable correspondences. The consistency-guided odometry module provides reliable initial pose estimations, and the lightweight mapping module completes localization and mapping tasks. Experiments on public datasets and in real-world scenarios show Light-LOAM outperforming the state-of-the-art solutions in accuracy and efficiency. The consistency graph effectively filters out outlier associations, enabling high-quality pose optimization with a limited number of feature samples. Future work involves integrating IMU data to mitigate point cloud distortion and designing a more robust graph-based feature matching method for enhanced reliability in data associations, leading to more accurate pose estimations.

REFERENCES

[1] Tim Bailey, Eduardo Mario Nebot, JK Rosenblatt, and Hugh F Durrant-Whyte. Data association for mobile robot navigation: A graph theoretic approach. In *2020 IEEE International Conference on Robotics and Automation.*, volume 3, pages 2512–2517. IEEE, 2000.

[2] Pengxin Chen, Wenzhong Shi, Sheng Bao, Muyang Wang, Wenzheng Fan, and Haodong Xiang. Low-drift odometry, mapping and ground segmentation using a backpack lidar system. *IEEE Robotics and Automation Letters*, page 7285–7292, Oct 2021.

[3] A Geiger, P Lenz, C Stiller, and R Urtasun. Vision meets robotics: The KITTI dataset. *The International Journal of Robotics Research*, page 1231–1237, Sep 2013.

[4] Richard Hartley and Andrew Zisserman. *Multiple view geometry in computer vision*. Cambridge university press, 2003.

[5] Wolfgang Hess, Damon Kohler, Holger Rapp, and Daniel Andor. Real-time loop closure in 2d lidar slam. In *2016 IEEE international conference on robotics and automation (ICRA)*, pages 1271–1278. IEEE, 2016.

[6] Kenji Koide, Jun Miura, and Emanuele Menegatti. A portable three-dimensional lidar-based system for long-term and wide-area people behavior measurement:. *International Journal of Advanced Robotic Systems*, page 172988141984153, Mar 2019.

[7] Pierre-Yves Lajoie, Benjamin Ramtoula, Yun Chang, Luca Carlone, and Giovanni Beltrame. DOOR-SLAM: Distributed, online, and outlier resilient slam for robotic teams. *IEEE Robotics and Automation Letters*, page 1656–1663, Apr 2020.

[8] Parker C Lusk, Kaveh Fathian, and Jonathan P How. Clipper: A graph-theoretic framework for robust data association. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 13828–13834. IEEE, 2021.

[9] Joshua G. Mangelson, Derrick Dominic, Ryan M. Eustice, and Ram Vasudevan. Pairwise consistent measurement set maximization for robust multi-robot map merging. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, May 2018.

[10] Marius Muja and David G Lowe. Fast approximate nearest neighbors with automatic algorithm configuration. *VISAPP (1)*, 2(331-340):2, 2009.

[11] Aleksandr Segal, Dirk Haehnel, and Sebastian Thrun. Generalized-ICP. In *Robotics: science and systems*, volume 2, page 435. Seattle, WA, 2009.

[12] Tixiao Shan and Brendan Englot. LEGO-LOAM: Lightweight and ground-optimized lidar odometry and mapping on variable terrain. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4758–4765. IEEE, 2018.

[13] Tixiao Shan, Brendan Englot, Drew Meyers, Wei Wang, Carlo Ratti, and Daniela Rus. LIO-SAM: Tightly-coupled lidar inertial odometry via smoothing and mapping. In *2020 IEEE/RSJ international conference on intelligent robots and systems (IROS)*, pages 5135–5142. IEEE, 2020.

[14] Ignacio Vizzo, Tiziano Guadagnino, Benedikt Mersch, Louis Wiesmann, Jens Behley, and Cyrill 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*, 8(2):1029–1036, 2023.

[15] Han Wang, Chen Wang, Chun-Lin Chen, and Lihua Xie. F-LOAM: Fast lidar odometry and mapping. In *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4390–4396. IEEE, 2021.

[16] Wei Xu and Fu Zhang. Fast-LIO: A fast, robust lidar-inertial odometry package by tightly-coupled iterated kalman filter. *IEEE Robotics and Automation Letters*, 6(2):3317–3324, 2021.

[17] Heng Yang, Jingnan Shi, and Luca Carlone. TEASER: Fast and certifiable point cloud registration. *IEEE Transactions on Robotics*, page 314–333, Apr 2021.

[18] Jiaqi Yang, Jiahao Chen, Siwen Quan, Wei Wang, and Yanning Zhang. Correspondence selection with loose–tight geometric voting for 3-D point cloud registration. *IEEE Transactions on Geoscience and Remote Sensing*, 60:1–14, 2022.

[19] Jiaqi Yang, Xiyu Zhang, Shichao Fan, Chunlin Ren, and Yanning Zhang. Mutual voting for ranking 3D correspondences. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2023.

[20] Jie Yin, Ang Li, Tao Li, Wenxian Yu, and Danping Zou. M2dgr: A multi-sensor and multi-scenario slam dataset for ground robots. *IEEE Robotics and Automation Letters*, 7(2):2266–2273, 2021.

[21] Ji Zhang and Sanjiv Singh. LOAM: Lidar odometry and mapping in real-time. In *Robotics: Science and Systems*, volume 2, pages 1–9. Berkeley, CA, 2014.

[22] Zichao Zhang and Davide Scaramuzza. A tutorial on quantitative trajectory evaluation for visual (-inertial) odometry. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 7244–7251. IEEE, 2018.