2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
October 1-5, 2023. Detroit, USA

2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) | 978-1-6654-9190-7/23/$31.00 ©2023 IEEE | DOI: 10.1109/IROS55552.2023.10341462

Multi-Session, Localization-oriented and Lightweight LiDAR Mapping Using Semantic Lines and Planes

Zehuan Yu, Zhijian Qiao, Liuyang Qiu, Huan Yin and Shaojie Shen

***Abstract*— In this paper, we present a centralized framework for multi-session LiDAR mapping in urban environments, by utilizing lightweight line and plane map representations instead of widely used point clouds. The proposed framework achieves consistent mapping in a coarse-to-fine manner. Global place recognition is achieved by associating lines and planes on the Grassmannian manifold, followed by an outlier rejection-aided pose graph optimization for map merging. Then a novel bundle adjustment is also designed to improve the local consistency of lines and planes. In the experimental section, both public and self-collected datasets are used to demonstrate efficiency and effectiveness. Extensive results validate that our LiDAR mapping framework could merge multi-session maps globally, optimize maps incrementally, and is applicable for lightweight robot localization.**

I. Introduction

Light Detection and Ranging (LiDAR) sensors are becoming standard equipment for mobile robots in the last two decades. Today's LiDAR mapping approaches, such as simultaneous localization and Mapping (SLAM), can provide dense 3D point cloud maps [1], [2]. Such maps have enabled various robotic navigation applications, particularly in GPS-unfriendly environments. With the deployment of multiple mobile robots for long-term autonomy, multiple LiDAR maps are constructed from multi-session data and it poses a challenge to researchers: a geometrically consistent global map is desired for robotic teams to operate effectively.

To address this challenge, researchers have designed advanced LiDAR mapping systems in a distributed [3], [4] or centralized [5], [6] manner. These systems could build a global point cloud map by integrating a chain of methods and techniques, such as place recognition, local matching, and graph optimization. These existing works lead multi-robot multi-session LiDAR mapping to a level of success, e.g., underground exploration in complex environments [7]. However, almost all multi-session or multi-robot mapping systems use dense feature points or point clouds. These maps bring underlying problems for robot localization and mapping, such as processing such maps on a cheap server and communication on resource-constrained vehicles, especially in a large-scale environment. One potential solution is to compress or downsample a point cloud map [8], [9]. But it is hard to guarantee map consistency when a new session is incrementally merged into the reduced point cloud map.

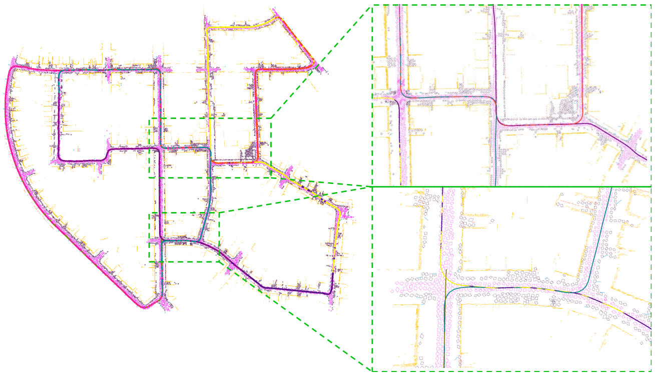

Fig. 1. Mapping results generated from our proposed framework. Multi-session trajectories are with different colors in the zoom-in view. Only semantic-aided lines and planes are utilized to achieve this global and local consistent map.

In this study, we propose a centralized LiDAR mapping framework for multi-session LiDAR mapping by using lightweight lines and planes rather than conventional point clouds. We carefully design a centralized LiDAR mapping framework based on lines and planes. Specifically, we build a map merging method as place recognition on the Grassmannian manifold [10], which unifies the data associations of lines and planes. We then conduct map optimization in a coarse-to-fine manner: first pose graph optimization followed by global bundle adjustment. Note that both global map merging and refinement use lightweight lines and planes without dense representations. Finally, the generated consistent map is lightweight while maintaining the accuracy of map-based localization.

Overall, the primary contributions of this study can be summarized as follows:

- We present a lightweight, consistent, and multi-session LiDAR mapping framework in urban environments with lines and planes.
- We design a global map merging method on the Grassmannian manifold of lines and planes, ensuring the global consistency of pose graph optimization.
- We propose a novel bundle adjustment with parameterized lines and planes, improving the consistency of LiDAR mapping.
- The proposed framework is validated with both public

This work was supported in part by the HKUST Postgraduate Studentship, in part by the HKUST-DJI Joint Innovation Laboratory, and in part by the Hong Kong Center for Construction Robotics (InnoHK center supported by Hong Kong ITC).

Zehuan Yu, Zhijian Qiao, Huan Yin and Shaojie Shen are with the Department of Electronic and Computer Engineering, The Hong Kong University of Science and Technology, Hong Kong, China. E-mail: zyuay@connect.ust.hk, zqiaoac@connect.ust.hk, eehyin@ust.hk, eeshaojie@ust.hk. Liuyang Qiu is with the Harbin Institute of Technology (Shenzhen). E-mail: qiuliuyang11@gmail.com

Corresponding author: Huan Yin

dataset and self-driving simulator, as well as multi-session data in large-scale urban environments.

II. Related Work

A. Advances in LiDAR Mapping

One well-known LiDAR-based robot mapping technique is based on scan matching methods, like iterative closest point (ICP) [11]. Scan matching generally requires dense point clouds to guarantee matching accuracy. In 2014, Zhang and Singh [1] proposed a feature points-based pioneering work, LOAM, which extracts local LiDAR features for odometry estimation and mapping. LOAM has inspired various LiDAR SLAM systems [12]. LOAM and its improved versions heavily rely on front-end feature extraction modules, making it less robust when applied to challenging scenarios. One promising direction is improving feature point design for a more robust and accurate LiDAR mapping system. Pan *et al.* [13] proposed to use principal components analysis to extract feature points, and build multi-metric least square optimization for pose estimation. Liu and Zhang [14] developed theoretical derivatives of LiDAR bundle adjustment with line and plane features, achieving map refinement as a back-end for conventional LiDAR SLAM systems. PLC SLAM by Zhou *et al.* [15] introduced planes, lines, and cylinders in the LiDAR SLAM system. Despite the geometric features mentioned above, advanced learning techniques [16] can extract semantic information in LiDAR scans and semantics can help improve LiDAR mapping. A semantic-aided LiDAR SLAM system was presented in [17], and semantic information was utilized for local odometry estimation and global loop closure detection.

In this study, we propose a LiDAR mapping framework that is partially inspired by previous works, including bundle adjustment and semantic-aided mapping. However, instead of using dense feature points, we directly use sparser lines and planes for both global map merging and refinement, making the entire mapping system more lightweight.

B. Multi-robot and Multi-Session Mapping

Multi-robot and multi-session localization and mapping attracts lots of research interest in recent years [18]. There are primarily two lines of research in this area: online and distributed multi-robot mapping [4], [19], and centralized map servers for handling multi-session maps such as Maplab [6], [20]. DiSCo-SLAM [4] used Scan Context descriptor [21] and ICP for loop closure detection and global scan matching. Pairwise consistent measurement set maximization (PCM) [22] was also introduced for outlier rejection in pose graph optimization (PGO). Kimera-Multi [19] proposed to use metric-semantic representations to build distributed SLAM system and graduated non-convexity (GNC) was introduced for robust PGO. In centralized multi-session mapping, Maplab system in 2018 [20] is an open source framework for visual-inertial mapping and localization, which includes map merging, batch optimization, and loop closing.. Recent Maplab 2.0 [6] integrates multiple sensor modalities into the map server, making centralized mapping more flexible. In this study, we propose a mapping framework that focuses on designing LiDAR-based global map merging and refinement for multi-session mapping in urban environments. The goal of this work is to make LiDAR mapping more lightweight and consistent with multi-session data.

We would also like to mention the existence of semantic-based map-merging techniques. Yue *et al.* [23] formulated a probabilistic fusion method to merge semantic point cloud maps. Guo *et al.* [24] proposed to use semantic descriptors and design a graph-matching method to align maps. In this study, we propose a novel approach that employs semantic information in the form of lines and planes for map merging. The proposed technique operates on the Grassmannian manifold, which is distinct from conventional Euclidean space-based methods.

III. METHODOLOGY

An overview of the system is depicted in Figure 2. Subsequently, this paper presents several subsections outlining the semantic mapping module using lines and planes, the proposed global map merging module, and a detailed description of the novel bundle adjustment.

A. From Point Clouds to Lines and Planes

1) LiDAR Odometry and Semantic Segmentation: The LiDAR odometry technique is well-studied in the community. In this study, we directly use it as the front end to obtain the odometry data and raw point cloud sub-maps for each session. Specifically, keyframes of sub-maps are generated based on variations in rotation and translation. We use open-source methods to obtain semantic labels of LiDAR scans. We will introduce the settings in the experimental section. The dense semantic point clouds are used for the following feature extraction and parameterization.

2) Semantic Feature Extraction: Several specific types of semantic landmarks, such as poles, roads, buildings, and fences, are selected as map elements based on prior knowledge of urban scenarios. These elements commonly exist in urban environments and have compact geometric representations. In this study, poles can be represented as infinite lines, while other landmarks are represented as infinite planes. To extract these line and plane features from dense semantic point clouds, a clustering algorithm [25] and a voxel-based segmentation algorithm are used. Essentially, line and plane features are low-dimensional landmarks in this study. These line and plane features have fewer parameters compared to dense point clouds, while still preserving the original geometric information, making them beneficial for map management in large-scale urban environments.

Specifically, for line feature extraction, clustering results provide several point cloud clusters that described different pole-like objects. The direction and centroid of each object are obtained using the principal component analysis (PCA) algorithm. For plane features, the semantic point cloud is divided into voxels based on position and encoded using a hash function, which is from [14]. Points within each voxel

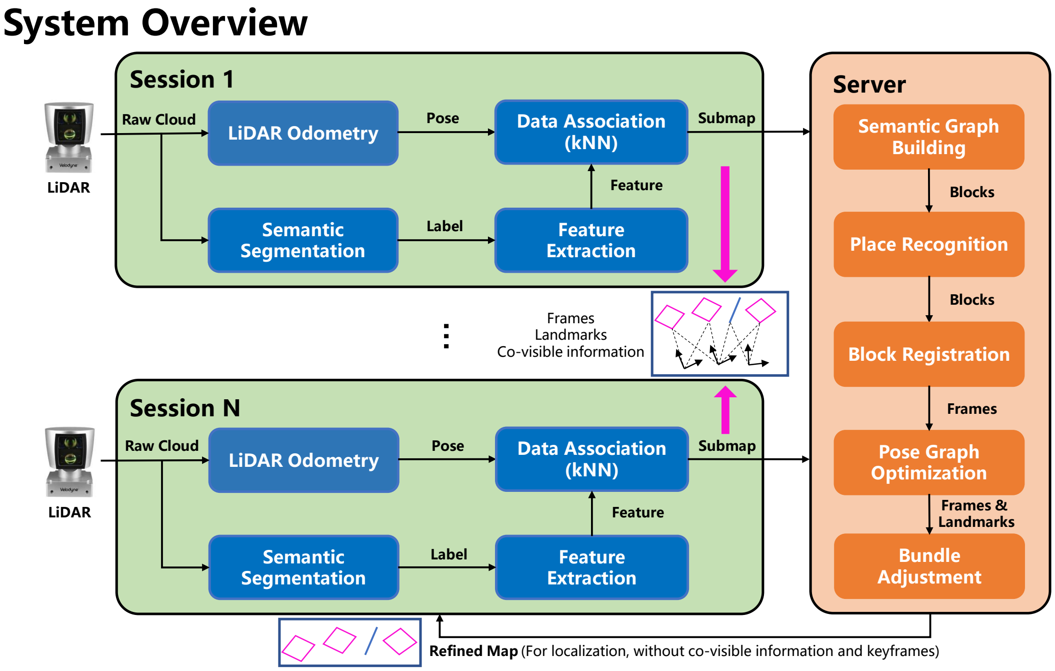

Fig. 2. System Overview. Online mapping and a centralized map server are depicted in green and orange blocks, respectively. The sub-maps comprise lightweight landmarks, including lines and planes, as well as co-visibility connections between keyframes and landmarks. The map server achieves multi-session mapping from scratch in a coarse-to-fine manner, starting with global map merging and subsequently local refinement.

are checked using the PCA algorithm, and plane features are extracted. The centroid and covariance matrix of each feature are saved in their host frame.

To build a lightweight mapping framework, a line observation is described using two terminal points $\mathbf{f}^{\mathcal{L}} := \langle \mathbf{p}_a, \mathbf{p}_b \rangle$, which are determined using the singular values and singular vectors from the PCA result. Similarly, a plane observation is described by the centroid and four terminal points $\mathbf{f}^{\mathcal{S}} := \langle \mathbf{p}_a, \mathbf{p}_b, \mathbf{p}_c, \mathbf{p}_d \rangle$, which also depended on the PCA result. The four points are the vertices of a rhombus and lie in the two directions of the singular vectors corresponding to the larger two singular values.

3) Lightweight Map Structure: We then initialize and update line and plane landmarks during the online mapping procedure, where data association based on the centroid-based nearest neighbor searching method constructs the co-visibility structure. We define a line landmark $\mathbf{l}^{\mathcal{L}} := \langle l_s, \mathbf{c}^{\mathcal{L}}, \mathbf{n}^{\mathcal{L}}, \mathbf{p}^{\mathcal{L}}, \{\mathbf{f}_i^{\mathcal{L}}\} \rangle$ and a plane landmark $\mathbf{l}^{\mathcal{S}} := \langle l_s, \mathbf{c}^{\mathcal{S}}, \mathbf{n}^{\mathcal{S}}, \mathbf{p}^{\mathcal{S}}, \{\mathbf{f}_i^{\mathcal{S}}\} \rangle$, including a semantic label, centroid, normal, minimum parameter block, and their observation in different keyframes, similar to the visual bundle structure.

However, the point-normal form, which represents normal vectors $\mathbf{n}$ and random points $\mathbf{c}$ in Euclidean space, is overparameterized for lines and planes and is not friendly for constructing optimization problems. To address this issue, we propose to represent an infinite line as $\mathbf{p}^{\mathcal{L}} := \langle \alpha, \beta, x, y \rangle \in \mathbb{R}^4$, where α and β represent the direction, and x and y represent the offset translation on the xOy plane. Similarly, an infinite plane is formulated as $\mathbf{p}^{\mathcal{S}} := \langle \alpha, \beta, d \rangle \in \mathbb{R}^3$, where α and β represent the direction, and d represents the offset translation on the z-axis, which are the minimum parameter blocks mentioned before.

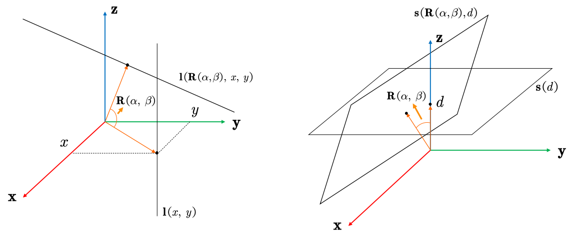
z
y
x
x
y
$\mathbf{R}(\alpha, \beta)$
$\mathbf{l}(\mathbf{R}(\alpha,\beta), x, y)$
$\mathbf{l}(x, y)$
$\mathbf{s}(\mathbf{R}(\alpha,\beta),d)$
z
$\mathbf{R}(\alpha, \beta)$
d
$\mathbf{s}(d)$
y
x

Fig. 3. Formulation of line and plane landmarks.

$$\mathbf{R}(\alpha, \beta) = \begin{bmatrix} \cos(\beta) & 0 & -\sin(\beta) \\ \sin(\alpha)\sin(\beta) & \cos(\alpha) & \sin(\alpha)\cos(\beta) \\ \cos(\alpha)\sin(\beta) & -\sin(\alpha) & \cos(\alpha)\cos(\beta) \end{bmatrix} \quad (1)$$

In fact, it is possible to transform all infinite lines and infinite planes from the original line $\mathbf{u}_z$ and original plane xOy in two steps, as demonstrated in Figure 3. Regarding line features, we first apply an offset (x, y) to the original line $\mathbf{u}_z$ to obtain the translated line $\mathbf{l}(x, y)$. Next, we rotate the nearest point (x, y) using a 2 degrees of freedom (DoF) rotation matrix $\mathbf{R}(\alpha, \beta)$ to obtain the final line $\mathbf{l}(\mathbf{R}(\alpha, \beta), x, y)$. The 2 DoF rotation matrix $\mathbf{R}(\alpha, \beta)$ is defined by equation (1). Concerning plane features, the original plane xOy is translated along the z axis and rotated using a 2 DoF rotation matrix, resulting in the final plane $\mathbf{s}(\mathbf{R}(\alpha, \beta), z)$.

The minimum line or plane representation is devised for the optimization framework, necessitating a residual related to the pose and landmark. While point-to-line and point-to-plane residuals are available, they are constructed using the point-normal form. Hence, it is essential to establish the mapping relations between the point-normal form and infinite line/plane form, and our proposed ones are stated as follows:

$$\begin{bmatrix}\mathbf{n}^{\mathcal{L}}\\ \mathbf{c}^{\mathcal{L}}\end{bmatrix} = \begin{bmatrix}\mathbf{R}(\alpha,\beta)\mathbf{u}_z\\ \mathbf{R}(\alpha,\beta)(\mathbf{u}_x x + \mathbf{u}_y y)\end{bmatrix} \tag{2}$$

$$\begin{bmatrix}\mathbf{n}^{\mathcal{S}}\\ d^{\mathcal{S}}\end{bmatrix} = \begin{bmatrix}\mathbf{R}(\alpha,\beta)\mathbf{u}_z\\ d\end{bmatrix} \tag{3}$$

B. Global Map Merging

1) Semantic Graph Building: To merge sub-maps at different locations, place recognition and relative pose estimation are critical challenges that must be solved globally, i.e., without an initial guess. Traditional methods typically employ full laser scan data to construct handcrafted or learning-based global descriptors. In our case, we implement the GraffMatch algorithm [26], which is a global descriptor-free method based on the open-sourced data association framework [27] to identify the overlapping parts between two sub-maps. However, since each sub-map contains numerous landmarks, the graph matching problem has an infeasible dimensionality, leading to unmanageable solution times.

To overcome this issue, we propose a carefully designed two-step method. Firstly, we partition the sub-map into blocks along the vehicle trajectory instead of directly matching the two sub-maps. Each block comprises a host keyframe and a set of landmarks around it. Secondly, to reduce the number of features, we cluster the plane landmarks that lie on the same infinite plane into a single plane. Specifically, we represent each pole as a line node $\mathbf{v}^L = \langle \mathrm{l_s}, \mathbf{c}, \mathbf{n}, \mathbf{l}^L \rangle$, and cluster the plane landmarks to form larger planes. These larger planes are represented as plane nodes $\mathbf{v}^S = \langle \mathrm{l_s}, \mathbf{c}, \mathbf{n}, \{\mathbf{l}_i^S\} \rangle$. The elements of the graph node include semantic labels, centroids, normals, and the corresponding landmark(s). The semantic graph consists of two types of nodes, namely $\mathcal{G} = \langle \{\mathbf{v}_i^L\}, \{\mathbf{v}_j^S\} \rangle$.

2) Place Recognition: In the GraffMatch algorithm, line and plane features are extracted to formulate the k-dim subspaces, and their Graff coordinates Y are shown as (4).

$$Y = \begin{bmatrix} A & b/\sqrt{\|b\|^2+1} \\ 0 & 1/\sqrt{\|b\|^2+1} \end{bmatrix} \in \mathbb{R}^{(n+1)\times(k+1)} \tag{4}$$

In our situation, A and b are the orthonormal basis and orthonormal displacement of the line or plane extracted from pole instances and clustered plane instances. Grassmannian metric $\mathrm{d_{Graff}}(Y_1, Y_2)$ can be used to compute the distance of two subspaces, such as:

$$\begin{aligned} b_{02} &= b_2 - b_1 \\ Y_1^{'} &= \begin{bmatrix} A_1 & 0 \\ 0 & 1 \end{bmatrix} \\ Y_2^{'} &= \begin{bmatrix} A_2 & b_{02}/\sqrt{\|b_{02}\|^2+1} \\ 0 & 1/\sqrt{\|b_{02}\|^2+1} \end{bmatrix} \\ \{\sigma_i\} &= \mathrm{SVD}((Y_1^{'})^T Y_2^{'}) \\ \mathrm{d_{Graff}}(Y_1, Y_2) &= \sum_i \arccos^2(\sigma_i) \end{aligned} \tag{5}$$

Then this metric is used to construct the affinity matrix for the following global matching framework. We recommend interested readers refer to CLIPPER [26] for a more detailed explanation.

3) Block Registration: Thanks to the CLIPPER framework, which gives the corrected line and plane correspondences $\mathcal{C}^{\mathcal{L}}$ and $\mathcal{C}^{\mathcal{S}}$ between semantic graphs $\mathcal{G}_{b_i}$ and $\mathcal{G}_{b_j}$ (b_i and b_j are two individual blocks). Then we can build a hybrid registration with lines and planes to solve the relative pose, which is formulated as

$$\begin{aligned} \min_{\mathbf{T}_{b_j}^{b_i}} & \sum_{(k,l)\in\mathcal{C}^{\mathcal{L}}} \rho(\|(\mathbf{I} - \mathbf{n}_k\mathbf{n}_k^T)(\mathbf{T}_{b_j}^{b_i}\mathbf{c}_l - \mathbf{c}_k)\|^2_{\Sigma_{\mathcal{L}}}) \\ & \sum_{(k,l)\in\mathcal{C}^{\mathcal{S}}} \rho(\|\mathbf{n}_k^T(\mathbf{T}_{b_j}^{b_i}\mathbf{c}_l - \mathbf{c}_k)\|^2_{\Sigma_{\mathcal{S}}}) \end{aligned} \tag{6}$$

where $\mathbf{T}_{b_j}^{b_i}$ is the relative transformation between block i and block j, while $\rho(\cdot)$ denotes a robust kernel function. However, due to the limitations of CLIPPER, it may not always be possible to find all the corresponding node pairs. Therefore, to obtain a more accurate estimate of the relative pose, an optimization-based refinement based on knn-search-based iterative closest landmark is performed, which is formulated as follows:

$$\begin{aligned} \min_{\mathbf{T}_{f_j}^{f_i}} & \sum_{(k,l)\in\mathcal{N}^{\mathcal{L}}} \rho(\|(\mathbf{I} - \mathbf{n}_k\mathbf{n}_k^T)(\mathbf{T}_{f_j}^{f_i}\mathbf{p}_l - \mathbf{p}_k)\|^2_{\Sigma_{\mathcal{L}}}) \\ & \sum_{(k,l)\in\mathcal{N}^{\mathcal{S}}} \rho(\|\mathbf{n}_k^T(\mathbf{T}_{f_j}^{f_i}\mathbf{p}_l - \mathbf{p}_k)\|^2_{\Sigma_{\mathcal{S}}}) \end{aligned} \tag{7}$$

where (k, l) is the nearest-neighbor pair between two blocks. To achieve a more robust registration result, the optimization problem is solved iteratively for improved matching of landmarks, leading to a more stable solution. Compared to descriptor-based place recognition methods such as Scan Context [21], GraffMatch is descriptor-free but computationally more intensive. Once all pairs of blocks have been registered, we use the pairwise consistent measurement (PCM) algorithm [22] to identify false loop candidates. The function $C(\mathbf{T}_{i_k}^{j_k}, \mathbf{T}_{i_l}^{j_l})$ measures the consistency of the relative pose of the block registration:

$$\begin{aligned} \delta\mathbf{T} &= (\mathbf{T}_{i_k}^{j_k})^{-1} \cdot \hat{\mathbf{T}}_{i_k}^{i_l} \cdot \mathbf{T}_{i_l}^{j_l} \cdot \hat{\mathbf{T}}_{j_l}^{j_k} \\ C(\mathbf{T}_{i_k}^{j_k}, \mathbf{T}_{i_l}^{j_l}) &= [\mathbf{r}(\delta\mathbf{R}), \mathbf{r}(\delta\mathbf{p})]^T \\ &= [\|\mathrm{Log}(\mathbf{R}(\delta\mathbf{T}))\|_2, \|\mathbf{p}(\delta\mathbf{T})\|_2]^T \end{aligned} \tag{8}$$

If $\mathbf{r}(\delta\mathbf{R})$ and $\mathbf{r}(\delta\mathbf{p})$ are small enough, the pair of relative pose $\mathbf{T}_{i_k}^{j_k}$ and $\mathbf{T}_{i_k}^{j_k}$ are considered to be consistent. The problem of solving the internal-consistent set of relative poses $\mathcal{L}_{\mathcal{O}}$ is a maximum clique problem that can be solved by the open-source parallel maximum clique library [28], such that all the remaining relative poses will be constructed as relative pose residual (9) and added to the following pose graph optimization.

$$\begin{aligned} &\mathbf{r}_{\mathcal{L}_{\mathcal{O}}}\left(\mathbf{T}_{L_k}^W, \mathbf{T}_{L_{k+1}}^W, \hat{\mathbf{T}}_{L_{k+1}}^{L_k}\right) = \\ &\begin{bmatrix} \mathbf{R}_{L_k}^{W\,T}\left(\mathbf{p}_{L_{k+1}}^W - \mathbf{p}_{L_k}^W\right) - \hat{\mathbf{p}}_{L_{k+1}}^{L_k} \\ (\hat{\mathbf{R}}_{L_{k+1}}^{L_k})^T (\mathbf{R}_{L_k}^W)^T \mathbf{R}_{L_{k+1}}^W \end{bmatrix} \end{aligned} \tag{9}$$

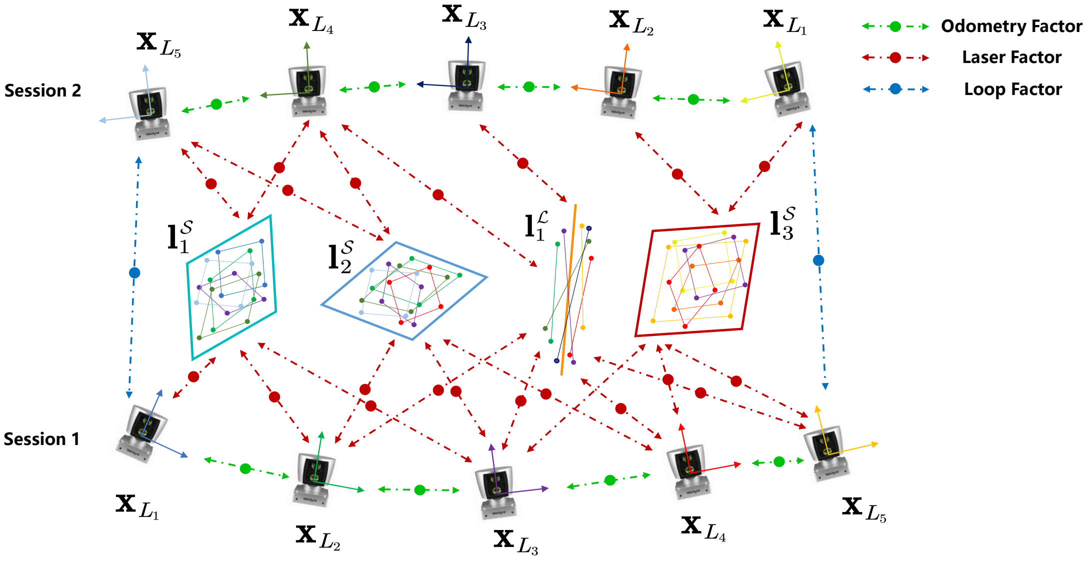

Fig. 4. Landmark definition and factors for multi-session LiDAR mapping: odometry factors (in green) obtained using LOAM, loop factors (in blue) obtained using GraffMatch, and laser factors (in red) obtained through the proposed bundle adjustment. Distinct colors are employed for lines and planes to enhance the description, with respect to its connected robot pose in the two sessions.

C. Map Refinement

1) Pose Graph Optimization: Generally, based on the front-end LiDAR odometry data, the odometry pose residuals $\mathbf{r}_{\mathcal{O}}$ have the same formulation as $\mathbf{r}_{\mathcal{L}_{\mathcal{O}}}$. Thus, the classical pose graph optimization problem is formulated as

$$\begin{aligned}\min_{\mathbf{T}^W_{L_i}\in\mathcal{F}} \sum_{(k,k+1)\in\mathcal{O}} &\rho(\|\mathbf{r}_{\mathcal{O}}\left(\mathbf{T}^W_{L_k},\mathbf{T}^W_{L_{k+1}},\hat{\mathbf{T}}^{L_k}_{L_{k+1}}\right)\|^2_{\Sigma_{\mathcal{O}}})+\\ \sum_{(i,j)\in\mathcal{L}_{\mathcal{O}}} &\rho(\|\mathbf{r}_{\mathcal{L}_{\mathcal{O}}}\left(\mathbf{T}^W_{L_i},\mathbf{T}^W_{L_j},\hat{\mathbf{T}}^{L_i}_{L_j}\right)\|^2_{\Sigma_{\mathcal{L}_{\mathcal{O}}}})\end{aligned} \tag{10}$$

Pose graph optimization provides a higher-precision global pose for keyframes and landmarks. However, it is possible to have landmarks that are repeatedly included in multiple sub-maps. To reduce the size of the map and the dimension of subsequent optimization, the instances of these landmarks in multiple sub-maps will be merged depending on either the graph-matching result or the centroid distance.

2) Bundle adjustment for lines and planes: After merging the overlap landmarks between sub-maps, we introduce a new bundle adjustment formulation to jointly optimize the keyframes' poses, line landmarks, and plane landmarks to improve the map accuracy.

Inspired by visual bundle adjustment methods, we construct the residual between the observation in the keyframe and the corresponding landmark in the world. Every line observation has two point-to-infinite-line residuals, which can be expressed as follows:

$$\mathbf{r}_{\mathcal{L}}\left(\mathbf{T}^W_L,\mathbf{l}^{\mathcal{L}}\right)=\left(\mathbf{I}-\mathbf{n}\mathbf{n}^T\right)\left(\mathbf{T}^W_L\mathbf{p}^L-\mathbf{q}\right) \tag{11}$$

where $\mathbf{T}^W_L$ is the pose of the keyframe, $\mathbf{l}^L$ is the line landmark whose point-normal form is $(\mathbf{n},\mathbf{q})$, and $\mathbf{p}_L$ represents one of the two observation points in the keyframe.

In the same way, every plane observation has four point-to-infinite-plane residuals, such as

$$\mathbf{r}_{\mathcal{S}}\left(\mathbf{T}^W_L,\mathbf{l}^{\mathcal{S}}\right)=\mathbf{n}^T\left(\mathbf{T}^W_L\mathbf{p}^L-\mathbf{q}\right) \tag{12}$$

With (2), (3), (11) and (12), we could easily calculate the jacobian matrices with respect to line parameters $\langle\alpha,\beta,x,y\rangle$ and plane parameters $\langle\alpha,\beta,d\rangle$. Then we formulate the LiDAR bundle adjustment formulation as

$$\begin{aligned}\min_{\mathcal{X}} \sum_{(k,k+1)\in\mathcal{O}} &\rho(\|\mathbf{r}_{\mathcal{O}}\left(\mathbf{T}^W_{L_k},\mathbf{T}^W_{L_{k+1}},\hat{\mathbf{T}}^{L_k}_{L_{k+1}}\right)\|^2_{\Sigma_{\mathcal{O}}})+\\ \sum_{(i,j)\in\mathcal{L}_{\mathcal{O}}} &\rho(\|\mathbf{r}_{\mathcal{L}_{\mathcal{O}}}\left(\mathbf{T}^W_{L_i},\mathbf{T}^W_{L_j},\hat{\mathbf{T}}^{L_i}_{L_j}\right)\|^2_{\Sigma_{\mathcal{L}_{\mathcal{O}}}})+\\ \sum_{(k,i)\in\mathcal{C}_{\mathcal{L}}} &\rho(\|\mathbf{r}_{\mathcal{L}}\left(\mathbf{T}^W_{L_k},\mathbf{l}^{\mathcal{L}}_i\right)\|^2_{\Sigma_{\mathcal{L}}})+\\ \sum_{(k,j)\in\mathcal{C}_{\mathcal{S}}} &\rho(\|\mathbf{r}_{\mathcal{S}}\left(\mathbf{T}^W_{L_k},\mathbf{l}^{\mathcal{S}}_i\right)\|^2_{\Sigma_{\mathcal{S}}})\end{aligned} \tag{13}$$

where $\mathcal{X}$ includes all the keyframes' poses and landmarks.

Finally, all the multi-session poses and landmarks are optimized jointly in our proposed mapping framework. It is worth noting that the pipeline of global map merging and refinement only involves lightweight lines and planes. The effectiveness and efficiency of this map type will be validated in the following experimental sections.

IV. Experiments

We first introduce the experimental settings and then present the extensive mapping results qualitatively and quantitatively.

A. Experimental settings

We implement our framework in C++ and employ the Ceres Solver [29] to solve the nonlinear optimization problems. To assess the efficacy of our framework, we select the KITTI dataset [30], a real-world dataset, and the CARLA simulator [31], a virtual dataset. Both datasets provide a large number of semantic-aided scans and ground truth poses that can be used to build and evaluate our mapping framework.

For the KITTI dataset, we choose Seq.00, Seq.05, and Seq.08, which belong to urban scenarios and have several overlapping parts to generate multi-session data. We use an open-source 3D semantic segmentation method[1] [32] to obtain semantic labels in KITTI datasets. For one sequence in KITTI, we divide it into multiple sessions and each session is with an overlapping area with some of the other sessions. As for CARLA, the multi-session data collection process in the simulator is convenient, and the number of sessions is not limited. We use Town03 in CARLA as the data collection environment, and generate ten driving sessions with random start locations. Since users can extract all precise LiDAR data from CARLA, we add Gaussian noise to the point cloud with a variance $\sigma = 4$cm. Additionally, 20% of the semantic labels are randomly altered to incorrect ones to create noise. For both the KITTI dataset and CARLA, we employ LOAM[2] [1] a classical LiDAR mapping framework, as the front end.

B. Effectiveness of Bundle Adjustment

We first present an individual validation of the effectiveness of our bundle adjustment method in improving local mapping consistency. Since our sub-maps do not have internal loops, we evaluate the consistency of the map by comparing the relative pose error (RPE) between the raw trajectory obtained from A-LOAM and the optimized trajectory obtained from bundle adjustment that uses the initial odometry pose from A-LOAM.

We use the KITTI Seq.00 dataset for validation, which is divided into six sessions. Our bundle adjustment algorithm could help the relative pose error of each trajectory, as shown in Table I, which means that the local map consistency is improved. Actually, local map consistency is essential for map merging tasks since an internally inconsistent map will destroy the source map structure during merging. In our experiments, we observed that A-LOAM is not always stable when the vehicle experiences large acceleration or rotation velocity, leading to large odometry drift. Our proposed bundle adjustment could effectively correct the drift and reduce inconsistency.

C. Multi-Session map merging and consistency

In this subsection, we demonstrate that our multi-session mapping framework can merge multiple sub-maps from scratch and construct a globally consistent map. To construct a globally consistent map, the multi-session mapping system is desired to extract loop candidates, eliminate all the false loop candidates and optimize all the keyframes and landmarks. In our experiments, our framework could merge almost all the sub-maps correctly[3]. A case study of map merging is presented in Figure 5.

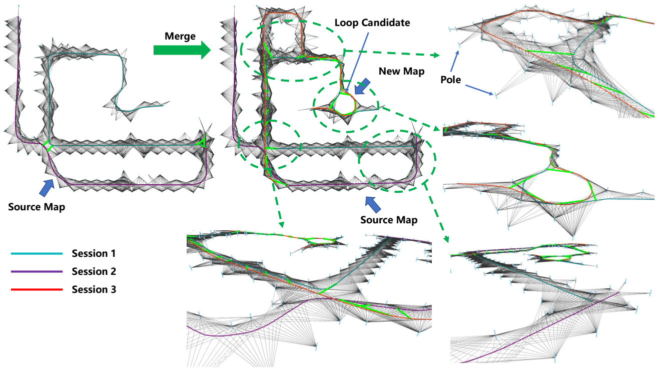

Fig. 5. A case of map merging and the co-visible connection with bird's-eye-view on CARLA simulator. The source map is from two sub-maps (Session 1 and Session 2, top-left) and a new map (Session 3, top-middle) is merged into the source map. The loop candidates (bold green lines) are provided by GraffMatch, and pruned by PCM and all the remaining loop candidates are reliable. The observations between the keyframes and landmarks are rendered as black lines (only poles are visualized for a clearer visualization).

To evaluate the mapping accuracy, we use the evaluation results in a LiDAR bundle adjustment (LiDAR-BA) paper [35] and the quantitative results of global trajectories are shown in Table II. Considering that A-LOAM has no loop closure module, we evaluate the global accuracy of SC-A-LOAM[4] that has a loop detector [21] and a global pose graph optimization as back end. In KITTI Seq.00, our framework provides a more accurate global translation and rotation estimation with our bundle adjustment. In Seq.05, our results are also comparable to other frameworks. However, for Seq.08, we believe that two reasons may have contributed to the inferior results. First, there is not enough overlapping area in Seq.08, resulting in a trajectory dependent solely on odometry. Second, A-LOAM is not as advanced as other LiDAR SLAM systems, and thus estimates a worse trajectory, introducing large errors in our proposed mapping framework. We also present the mapping results on CARLA datasets, shown in Figure 6.

D. Map Lightweightness

Despite the consistency of maps, we also evaluate the lightweightness of our proposed map representation compared to the traditional point cloud map. To this end, we conduct experiments on the KITTI dataset and compare the storage requirements of our lightweight map to those of dense point cloud maps that have varying downsampling resolutions r. If our map is designed solely for localization without frames or co-visible information, it will include only line and plane landmarks, and we emphasize this with (L) label. The results are summarized in Table III.

[1] https://github.com/mit-han-lab/spvnas

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

[3] A supplementary video presents the global map merging process.

[4] https://github.com/gisbi-kim/SC-A-LOAM

TABLE I

RMSE of the RPE ($^\circ/m$) On KITTI Seq. 00 Dataset

Methods	Session 0	Session 1	Session 2	Session 3	Session 4	Session 5
A-LOAM	0.1547/0.0223	0.1318/0.0212	0.1433/0.0232	0.1260/0.0229	0.1318/0.0220	0.1375/0.0235
A-LOAM+ Ours BA	**0.1489/0.0195**	**0.1318/0.0193**	**0.1317/0.0220**	**0.1203/0.0202**	**0.1260/0.0197**	**0.1260/0.0215**

TABLE II

RMSE of the APE ($^\circ/m$) On KITTI and CARLA Datasets

Methods	Seq. 00	Seq. 05	Seq. 08	Carla T3
SC-A-LOAM	1.0/1.3	0.7/0.8	2.6/4.3	/
CT-ICP [33]	**0.7**/1.7	**0.5**/0.8	1.2/2.5	/
MULLS [13]	**0.7**/1.1	0.6/1.0	1.3/2.9	/
LiTAMIN2 [34]	0.8/1.3	0.7/0.6	**0.9/2.1**	/
LiDAR-BA [35]	**0.7**/0.8	**0.5/0.4**	1.3/2.7	/
Ours(w/o BA)	**0.7**/0.9	0.6/0.6	2.7/4.5	0.229/0.512
Ours(w BA)	**0.7/0.7**	0.6/0.5	2.7/4.5	**0.143/0.432**

TABLE III

Map Storage (MB) of Different Representations

Representation	KITTI Seq. 00	KITTI Seq. 05	KITTI Seq. 08
Cloud (r=0.1m)	1773.40	1236.75	1254.92
Cloud (r=0.3m)	23.48	16.77	18.56
Cloud (r=0.5m)	8.33	6.01	6.57
Ours	1.85 (0.50/km)	1.22 (0.55/km)	1.56 (0.48/km)
Ours (L)	**0.77** (0.21/km)	**0.52** (0.24/km)	**0.64** (0.20/km)

Our lightweight map requires significantly less storage than the point cloud maps while maintaining map consistency. This feature allows for reduced data transmission loading between the robot and the server, leading to more efficient communication.

E. Map re-use for online localization

After obtaining the globally and locally consistent map, we test our lightweight map for online vehicle localization. To avoid the influence of global mapping error, we replace the estimated frame poses with ground truth poses and reconstruct the map based on the co-visible information. With this consistent map constructed by the ground truth poses, the global errors for evaluation are only caused by the localization procedure.

We adopt a semantic-aided point-to-landmarks method to achieve online localization as described in Equation (14).

$$\min_{\mathbf{T}_L^W} \sum_{(k,l)\in\mathcal{M}^{\mathcal{L}}} \rho(\|(\mathbf{I}-\mathbf{n}_k\mathbf{n}_k^T)(\mathbf{T}_L^W\mathbf{p}_l-\mathbf{p}_k)\|^2_{\Sigma_{\mathcal{L}}}) \\ \sum_{(k,l)\in\mathcal{M}^{\mathcal{S}}} \rho(\|\mathbf{n}_k^T(\mathbf{T}_L^W\mathbf{p}_l-\mathbf{p}_k)\|^2_{\Sigma_{\mathcal{S}}}) \tag{14}$$

TABLE IV

Lightweight LiDAR Localization on KITTI Dataset: APE ($^\circ/m$)

KITTI Seq. 00	KITTI Seq. 05	KITTI Seq. 08
0.243/0.035	0.237/0.045	0.376/0.060

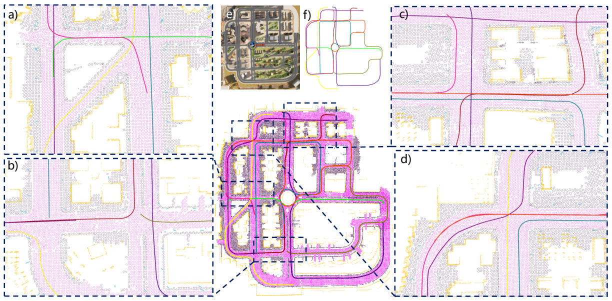

Fig. 6. Mapping results with bird's-eye-view on CARLA simulator. The multi-session data is collected in a town using a LiDAR-equipped vehicle. (a)-(d): zoomed views of the semantic lines and planes, as well as colored multi-session trajectories. (e): aerial view of the town seen from the sky. (f): trajectories at different sessions indicated by different colors.

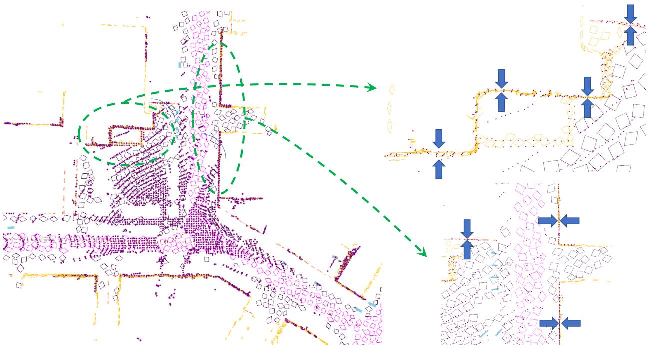

Fig. 7. Visualization of online localization on KITTI dataset. Semantics are in different colors in the zoom-in view. The points are registered accurately to the map, demonstrating that our lightweight map is efficient and reliable for online localization.

where $\mathcal{M}^{\mathcal{L}}$ and $\mathcal{M}^{\mathcal{L}}$ refer to the sets of line landmarks and plane landmarks, respectively, in the map. The variable $\mathbf{p}_l$ denotes the point in the current laser scan, whereas $\mathbf{p}_k$ represents the centroid of the nearest landmark with respect to the current estimated pose $\mathbf{T}_L^W$. All quantitative results for the KITTI dataset are presented in Table IV. The final localization error is at the centimeter level, which is sufficiently precise for localization-oriented applications, such as autonomous driving in urban environments. Moreover, the average latency for solving the scan-to-map problem 14 using a single thread on KITTI sequences is approximately 61ms (CPU: Intel i7-8700K), which is sufficient for real-time localization applications. We present a visualization result in Figure 7 to help understand our proposed online localization on the lightweight map.

V. Conclusion

In this paper, we propose and validate a multi-session, localization-oriented, and lightweight LiDAR mapping framework in urban environments. The framework includes both global map merging and local refinement, and only uses semantic lines and planes in the pipeline. The generated maps are lightweight compared to point cloud maps and can support online robot localization. There are several promising directions for future work that could improve and extend the proposed framework. Our ultimate goal is to achieve efficient crowd-sourced mapping in city-scale environments.

References

[1] J. Zhang and S. Singh, "Loam: Lidar odometry and mapping in real-time." in *Robotics: Science and Systems*, vol. 2, no. 9. Berkeley, CA, 2014, pp. 1–9.

[2] K. Chen, B. T. Lopez, A.-a. Agha-mohammadi, and A. Mehta, "Direct lidar odometry: Fast localization with dense point clouds," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 2000–2007, 2022.

[3] P.-Y. Lajoie, B. Ramtoula, Y. Chang, L. Carlone, and G. Beltrame, "Door-slam: Distributed, online, and outlier resilient slam for robotic teams," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 1656–1663, 2020.

[4] Y. Huang, T. Shan, F. Chen, and B. Englot, "Disco-slam: distributed scan context-enabled multi-robot lidar slam with two-stage global-local graph optimization," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 1150–1157, 2021.

[5] Y. Chang, K. Ebadi, C. E. Denniston, M. F. Ginting, A. Rosinol, A. Reinke, M. Palieri, J. Shi, A. Chatterjee, B. Morrell *et al.*, "Lamp 2.0: A robust multi-robot slam system for operation in challenging large-scale underground environments," *IEEE Robotics and Automation Letters*, vol. 7, no. 4, pp. 9175–9182, 2022.

[6] A. Cramariuc, L. Bernreiter, F. Tschopp, M. Fehr, V. Reijgwart, J. Nieto, R. Siegwart, and C. Cadena, "maplab 2.0–a modular and multi-modal mapping framework," *IEEE Robotics and Automation Letters*, vol. 8, no. 2, pp. 520–527, 2022.

[7] K. Ebadi, L. Bernreiter, H. Biggie, G. Catt, Y. Chang, A. Chatterjee, C. E. Denniston, S.-P. Deschênes, K. Harlow, S. Khattak *et al.*, "Present and future of slam in extreme underground environments," *arXiv preprint arXiv:2208.01787*, 2022.

[8] M. Labussière, J. Laconte, and F. Pomerleau, "Geometry preserving sampling method based on spectral decomposition for large-scale environments," *Frontiers in Robotics and AI*, vol. 7, p. 572054, 2020.

[9] H. Yin, Y. Wang, L. Tang, X. Ding, S. Huang, and R. Xiong, "3d lidar map compression for efficient localization on resource constrained vehicles," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 2, pp. 837–852, 2020.

[10] P. C. Lusk and J. P. How, "Global data association for slam with 3d grassmannian manifold objects," in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2022, pp. 4463–4470.

[11] F. Pomerleau, F. Colas, R. Siegwart *et al.*, "A review of point cloud registration algorithms for mobile robotics," *Foundations and Trends® in Robotics*, vol. 4, no. 1, pp. 1–104, 2015.

[12] T. Shan and B. 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)*. IEEE, 2018, pp. 4758–4765.

[13] Y. Pan, P. Xiao, Y. He, Z. Shao, and Z. Li, "Mulls: Versatile lidar slam via multi-metric linear least square," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021, pp. 11 633–11 640.

[14] Z. Liu and F. Zhang, "Balm: Bundle adjustment for lidar mapping," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 3184–3191, 2021.

[15] L. Zhou, G. Huang, Y. Mao, J. Yu, S. Wang, and M. Kaess, "Plc-lislam: Lidar slam with planes, lines, and cylinders," *IEEE Robotics and Automation Letters*, vol. 7, no. 3, pp. 7163–7170, 2022.

[16] A. Milioto, I. Vizzo, J. Behley, and C. Stachniss, "Rangenet++: Fast and accurate lidar semantic segmentation," in *2019 IEEE/RSJ international conference on intelligent robots and systems (IROS)*. IEEE, 2019, pp. 4213–4220.

[17] L. Li, X. Kong, X. Zhao, W. Li, F. Wen, H. Zhang, and Y. Liu, "Sa-loam: Semantic-aided lidar slam with loop closure," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021, pp. 7627–7634.

[18] S. Saeedi, M. Trentini, M. Seto, and H. Li, "Multiple-robot simultaneous localization and mapping: A review," *Journal of Field Robotics*, vol. 33, no. 1, pp. 3–46, 2016.

[19] Y. Tian, Y. Chang, F. H. Arias, C. Nieto-Granda, J. P. How, and L. Carlone, "Kimera-multi: Robust, distributed, dense metric-semantic slam for multi-robot systems," *IEEE Transactions on Robotics*, vol. 38, no. 4, 2022.

[20] T. Schneider, M. Dymczyk, M. Fehr, K. Egger, S. Lynen, I. Gilitschenski, and R. Siegwart, "maplab: An open framework for research in visual-inertial mapping and localization," *IEEE Robotics and Automation Letters*, vol. 3, no. 3, pp. 1418–1425, 2018.

[21] G. Kim and A. Kim, "Scan context: Egocentric spatial descriptor for place recognition within 3d point cloud map," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2018, pp. 4802–4809.

[22] J. G. Mangelson, D. Dominic, R. M. Eustice, and R. Vasudevan, "Pairwise consistent measurement set maximization for robust multi-robot map merging," in *2018 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2018, pp. 2916–2923.

[23] Y. Yue, C. Zhao, Z. Wu, C. Yang, Y. Wang, and D. Wang, "Collaborative semantic understanding and mapping framework for autonomous systems," *IEEE/ASME Transactions on Mechatronics*, vol. 26, no. 2, pp. 978–989, 2020.

[24] X. Guo, J. Hu, J. Chen, F. Deng, and T. L. Lam, "Semantic histogram based graph matching for real-time multi-robot global localization in large scale environment," *IEEE Robotics and Automation Letters*, vol. 6, no. 4, pp. 8349–8356, 2021.

[25] M. Ester, H.-P. Kriegel, J. Sander, X. Xu *et al.*, "A density-based algorithm for discovering clusters in large spatial databases with noise." in *kdd*, vol. 96, no. 34, 1996, pp. 226–231.

[26] P. C. Lusk, D. Parikh, and J. P. How, "Graffmatch: Global matching of 3d lines and planes for wide baseline lidar registration," *IEEE Robotics and Automation Letters*, vol. 8, no. 2, pp. 632–639, 2022.

[27] P. C. Lusk, K. Fathian, and J. P. How, "Clipper: A graph-theoretic framework for robust data association," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021, pp. 13 828–13 834.

[28] R. A. Rossi, D. F. Gleich, A. H. Gebremedhin, M. M. A. Patwary, and M. Ali, "A fast parallel maximum clique algorithm for large sparse graphs and temporal strong components," *CoRR, abs/1302.6256*, 2013.

[29] S. Agarwal, K. Mierle, and T. C. S. Team, "Ceres Solver," 3 2022. [Online]. Available: https://github.com/ceres-solver/ceres-solver

[30] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? the kitti vision benchmark suite," in *2012 IEEE conference on computer vision and pattern recognition*. IEEE, 2012, pp. 3354–3361.

[31] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, and V. Koltun, "Carla: An open urban driving simulator," in *Conference on robot learning*. PMLR, 2017, pp. 1–16.

[32] H. Tang, Z. Liu, S. Zhao, Y. Lin, J. Lin, H. Wang, and S. Han, "Searching efficient 3d architectures with sparse point-voxel convolution," in *European Conference on Computer Vision*, 2020.

[33] P. Dellenbach, J.-E. Deschaud, B. Jacquet, and F. Goulette, "Ct-icp: Real-time elastic lidar odometry with loop closure," in *2022 International Conference on Robotics and Automation (ICRA)*. IEEE, 2022, pp. 5580–5586.

[34] M. Yokozuka, K. Koide, S. Oishi, and A. Banno, "Litamin2: Ultra light lidar-based slam using geometric approximation applied with kl-divergence," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021, pp. 11 619–11 625.

[35] X. Liu, Z. Liu, F. Kong, and F. Zhang, "Large-scale lidar consistent mapping using hierarchical lidar bundle adjustment," *IEEE Robotics and Automation Letters*, vol. 8, no. 3, pp. 1523–1530, 2023.