Linear Four-Point LiDAR SLAM for Manhattan World Environments

Eunju Jeong[1], Jina Lee[1], Suyoung Kang[2], and Pyojin Kim[3]

***Abstract*—We present a new SLAM algorithm that utilizes an inexpensive four-point LiDAR to supplement the limitations of the short-range and viewing angles of RGB-D cameras. Herein, the four-point LiDAR can detect distances up to 40 m, and it senses only four distance measurements per scan. In open spaces, RGB-D SLAM approaches, such as L-SLAM, fail to estimate robust 6-DoF camera poses due to the limitations of the RGB-D camera. We detect walls beyond the range of RGB-D cameras using four-point LiDAR; subsequently, we build a reliable global Manhattan world (MW) map while simultaneously estimating 6-DoF camera poses. By leveraging the structural regularities of indoor MW environments, we overcome the challenge of SLAM with sparse sensing owing to the four-point LiDARs. We expand the application range of L-SLAM while preserving its strong performance, even in low-textured environments, using the linear Kalman filter (KF) framework. Our experiments in various indoor MW spaces, including open spaces, demonstrate that the performance of the proposed method is comparable to that of other state-of-the-art SLAM methods.**

***Index Terms*—Vision-Based Navigation, Computer Vision for Transportation, Sensor Fusion, RGB-D Perception.**

I. Introduction

SIMULTANEOUS localization and mapping (SLAM) is a fundamental problem in robotics that involves building a map of an unknown environment and simultaneously estimating a 6-DoF pose of a robot within that map. Typical feature-based RGB-D SLAM methods such as ORB-SLAM3 [1] and DROID-SLAM [2] demonstrate robust performance in rich-texture environments. However, these SLAM methods can be degraded in low-texture environments. To circumvent this issue, L-SLAM [3] presents a linear RGB-D SLAM by tracking the Manhattan frame (MF) [4], [5] using the structural regularities of the scene. However, owing to a short measuring distance (up to 5 m) and limitation in the viewing angle of depth cameras, the performance of RGB-D SLAM is degraded in wide and open spaces, which are commonly encountered in structured indoor environments as shown in Fig 1.

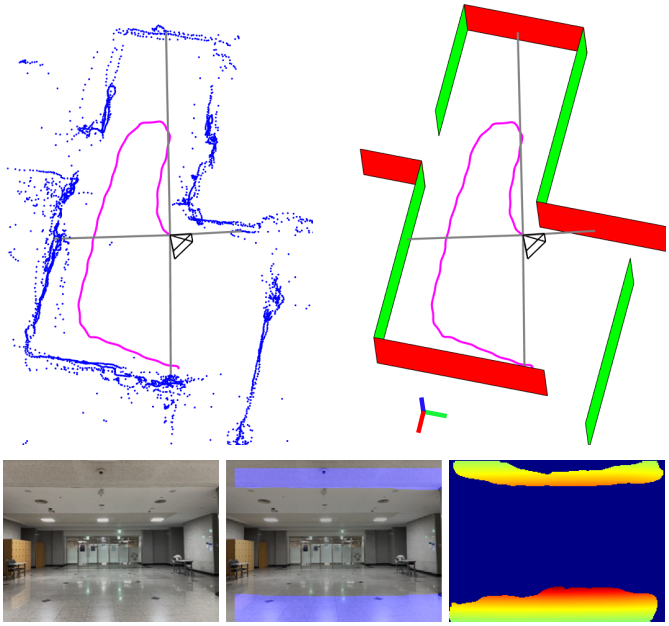

Fig. 1. Accumulated point cloud built using four radially-spaced single-point LiDARs (gray lines) and commercial visual-inertial odometry (magenta) (left). Depth images from an RGB-D camera are ineffective in wide and open spaces (bottom). We overcome SLAM with sparse sensing problems caused by the four-point LiDARs utilizing the Manhattan world (MW) assumption (right).

LiDAR odometry and SLAM [6], [7] can be an alternative to overcome the limitations of RGB-D cameras, showing promising SLAM results. However, conventional LiDAR sensors such as Velodyne VLP-16 are too heavy and expensive to attach to micro-UAVs such as Crazyflie 2.0 [8] or small autonomous robots, and thus cannot be applied universally.

To address these issues, we propose linear four-point LiDAR SLAM (FL-SLAM) for structural environments, which extends the previous L-SLAM to achieve consistent performance even in wide and open spaces. We employ the inexpensive four single-point LiDARs that can detect only four ranges in a single scan with a maximum range of 40 m to perceive surrounding orthogonal walls beyond the short-range of the RGB-D camera. The four-point LiDARs can obtain a significantly smaller amount of sparse range measurements than the typical LiDAR with more than 180 dense range measurements per scan, resulting in a SLAM with a sparse sensing problem. We overcome this challenging problem by effectively utilizing the Manhattan world (MW) structural regularities [9] of the indoor environments, and line RANSAC to represent detected

Manuscript received: April, 5, 2023; Revised July, 13, 2023; Accepted August, 4, 2023.

This paper was recommended for publication by Editor Pascal Vasseur upon evaluation of the Associate Editor and Reviewers' comments. This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (No. NRF-2021R1F1A1061397). *(Corresponding author: Pyojin Kim.)*

[1]Eunju Jeong and Jina Lee are with Department of Mechanical Systems Engineering, Sookmyung Women's University, Seoul 04310, South Korea. {eunju0316,jinaleeci}@sookmyung.ac.kr

[2]Suyoung Kang is with Department of Electronics Engineering, Sookmyung Women's University, Seoul 04310, South Korea. 1913084@sookmyung.ac.kr

[3]Pyojin Kim is with the School of Mechanical Engineering, Gwangju Institute of Science and Technology (GIST), Gwangju 61005, South Korea. pjinkim@gist.ac.kr

Digital Object Identifier (DOI): see top of this page.

surrounding walls as orthogonal lines directly modeled as landmarks with insufficient range measurements. We evaluate the proposed method in various indoor MW environments and demonstrate that FL-SLAM produces comparable 6-DoF camera poses to other state-of-the-art SLAM approaches. Our main contributions are as follows:

- We utilize MW structural regularity to build a reliable global MW map with sparse sensing of inexpensive four-point LiDARs.
- We seamlessly integrate with a linear KF-based RGB-D SLAM (L-SLAM) based on the global MW map built using four-point LiDARs to enable its effective performance in wide and open spaces.
- We evaluate the proposed FL-SLAM on the author-collected building-scale indoor MW environments, demonstrating comparable estimation results compared to other state-of-the-art SLAM methods.

II. Related Work

Depth-based and RGB-D SLAM methods have been actively studied in robotics and computer vision communities. In wide and open spaces in indoor environments, the performance of RGB-D SLAM, such as L-SLAM [3], [10], is severely degraded because it cannot identify the surrounding walls due to the limited sensing range of an RGB-D camera. ORB-SLAM3 [1], a traditional SLAM that uses expensive SLAM techniques (loop closure, pose graph optimization), is also degraded in wide and open spaces due to the limitations of the RGB-D camera. Recently, deep learning-based RGB-D SLAM methods have been developed [2], [11]–[13]. [11] utilizes points and line segments together with plane detection similar to those of L-SLAM, but they use CNNs to detect planes. However, deep learning-based RGB-D SLAM algorithms require expensive additional devices such as GPUs for training and long training times.

Other LiDAR odometry and SLAM studies [6], [7], [14] use Velodyne LiDAR, which can measure up to 100 m with high accuracy and a horizontal field of view of 360 degrees, showing promising odometry and SLAM results in open spaces. However, conventional LiDAR sensors (Velodyne VLP-16 and Ouster OS1) are too expensive and heavy to attach to micro-UAVs such as Crazyflie 2.0 [8] or small autonomous robots, and thus cannot be applied universally.

We utilize inexpensive four-point LiDARs that can only measure four ranges with a maximum range of 40 m per scan, resulting in SLAM with a sparse sensing problem. Performing SLAM with sparse sensing presents a significant challenge, as the system simultaneously constructs the precise map and estimates the camera pose using limited and sparse measurements with increased uncertainty. Only a few studies have focused on solving this problem. [15]–[17] use the Rao-Blackwellized particle filter (RBPF) [18], and [19] employs the pose graph optimization to solve the problem of SLAM with sparse sensing. [15] extracts line features as landmarks from the consecutive observation, including loop-closure detection to refine the results. [16] also uses loop-closure detection to refine the map and the estimated pose while assuming that the walls are orthogonal. We also adopt the MW assumption [20] to build an accurate map from sparse sensing, but we can produce a reliable 3D map and accurately estimate the 6-DoF camera pose without the loop-closure detection technique within a linear KF framework. Similar to us, [17], [19] use a micro UAV [8] with a four-point LiDAR, which can sense four ranges with a maximum range of 4 m per scan. [19] is the first to apply a graph-based approach to the problem of SLAM with sparse sensing. They replace scan-matching as the frontend for sparse range data and propose an approximate match heuristic for efficient loop-closure detection. However, all these SLAM with sparse sensing algorithms are for 2D SLAM with the assumption that the translational motion of the camera is always planar. By contrast, we build a 3D structured map and estimate the full 6-DoF camera motion.

III. Proposed Method

Our proposed FL-SLAM method builds on the previous linear RGB-D SLAM (L-SLAM) [3]. However, while L-SLAM fails to track accurate 6-DoF egomotion of the camera in wide and open spaces due to the limited sensing range of the depth camera, we expand it using sparse range measurements from four inexpensive single-point LiDARs. Fig 2 shows an overview of the proposed FL-SLAM method.

A. Linear RGB-D SLAM (L-SLAM)

We summarize the L-SLAM briefly (for full details, refer to [3]). L-SLAM has three main steps: 1) it tracks structural regularities (MF) to obtain the 3-DoF drift-free rotation and detects dominant orthogonal planar features in the scene using the tracked MF; 2) it estimates 3-DoF translation by minimizing the de-rotated reprojection error from the tracked points; and 3) it measures the 1-D distances to the identified orthogonal planes [22] from the currently tracked camera pose and updates the 3-DoF translation and 1-D distance of the associated planes in the global structure map within a linear Kalman filter (KF) framework.

The key idea of the drift-free rotation estimation in L-SLAM is to track the MF, the set of three orthogonal axes commonly found in structured indoor environments. L-SLAM utilizes the vanishing directions (VDs) [23] from image lines in RGB images and surface normals from depth images [24] to track the MF of the structured environments. Once the absolute orientation of the current scene has been established, L-SLAM can identify the dominant orthogonal planes whose normals are aligned with the tracked MF. Wide and open spaces, however, do not contain dominant vertical planes (walls) that can be detected by the RGB-D camera due to the limited sensing range, resulting in highly degraded rotational motion tracking. In addition, L-SLAM can only update the height value of the camera motion (Z-axis direction in MF) among the 3-DoF translational motion since only the horizontal planes (floors or ceilings) are observable in open spaces as shown in Fig 1. Due to the nature of VO, L-SLAM cannot avoid accumulated drift error over time in X and Y translational motion in wide and open spaces.

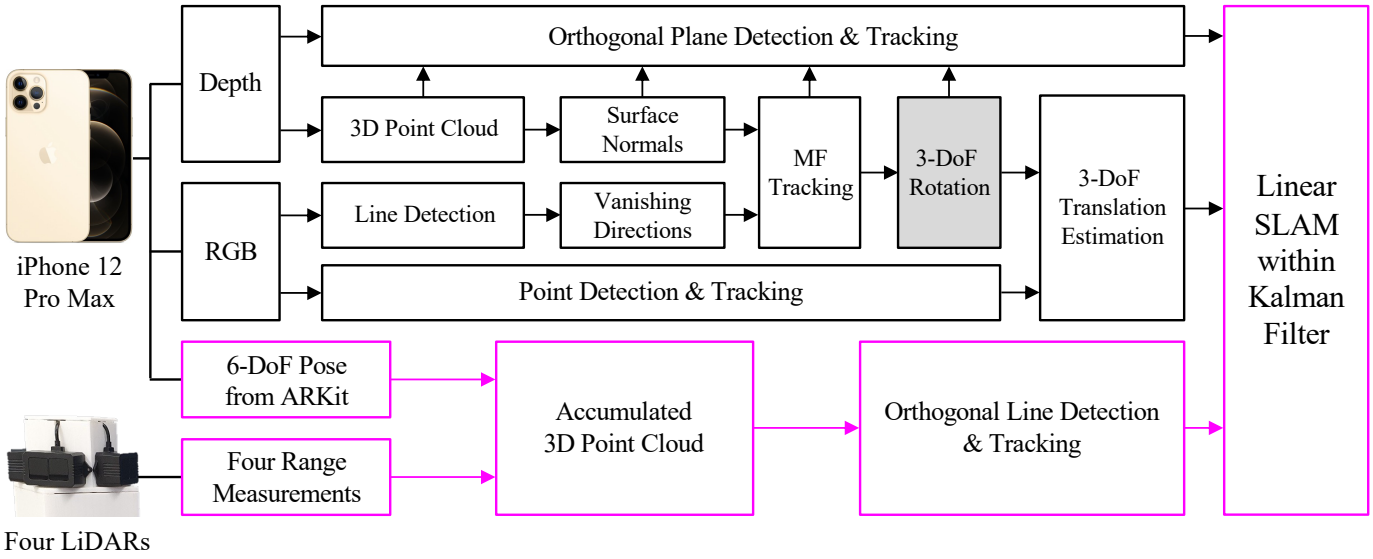

Fig. 2. Overview of the proposed FL-SLAM algorithm. We highlight key components (magenta) of the proposed approach for our main contributions. Four radially-spaced LiDARs give only four range readings in a single scan. We obtain an accumulated point cloud with Apple ARKit, a commercial visual-inertial odometry (VIO) known to be the most stable and accurate [21]. We build a global MW map by projecting accumulated point clouds onto the floor in the MW and detecting/tracking orthogonal lines. Note that we utilize 3-DoF rotation from ARKit when L-SLAM fails to track in open spaces (a box filled in gray).

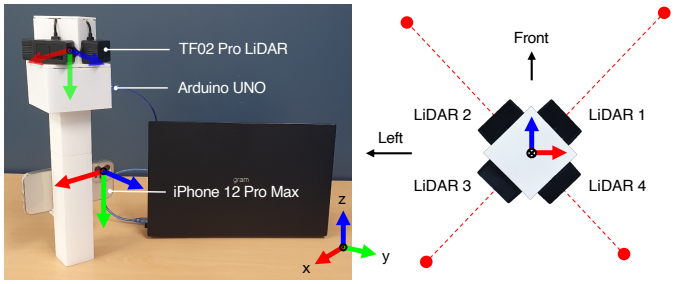

Fig. 3. The custom-built rig with four LiDARs and iPhone 12 Pro Max (left). We connect four TF02 Pro LiDARs to an Arduino UNO, and obtain four range measurements at 15 Hz. We illustrate the body frame of four LiDAR sensors and the camera frame. Top view of four radially-spaced LiDARs (right). We place the four ToF LiDAR sensors orthogonal to each other.

B. Orthogonal Wall Detection with Four LiDARs

To mitigate drift error in the X and Y translation in open spaces, it is crucial to detect surrounding walls in the structured environments. To achieve this, we designed a custom-built rig using a 3D printer to fix four LiDARs and an RGB-D camera (iPhone 12 Pro Max) in a single rigid body with the four-point LiDAR placed diagonally as shown in Fig. 3. As in a previous study on SLAM with sparse sensing [19], placing the four-point LiDAR in four directions (front, right, left, and back) can result in inaccurate measurements and noisy data in long corridors.

The overall procedure of the proposed FL-SLAM is shown in Fig. 2. We gradually extend the global MW map by fitting the lines to the accumulated point cloud obtained through current detected points from the four LiDARs and the ARKit 6-DoF camera poses from the iPhone. The global MW map accumulates walls detected from four-point LiDAR and RGB-D images, and each wall consists of three components: 1) 1-D distance (offset) from the origin of the global Manhattan frame, 2) alignment for the global Manhattan frame, and 3) two endpoints of the line or plane.

1) Converting Range Measurements to Point Cloud: To obtain a 3D point cloud of the structured environment with four range measurements as shown in Fig. 4, we convert each range measurement from a body frame b to a global Manhattan frame g using the following equation. To ensure a reliable point cloud, we only convert range measurements between 0.4 and 30 m into 3D Cartesian coordinates:

$$\begin{bmatrix} X_g \\ Y_g \\ Z_g \\ 1 \end{bmatrix} = T_{gb} \begin{bmatrix} X_b \\ Y_b \\ Z_b \\ 1 \end{bmatrix}, \; T_{gb} = \begin{bmatrix} R_{gb} & \mathbf{t}_{gb} \\ \mathbf{0}_{1\times 3} & 1 \end{bmatrix} \in \mathbb{R}^{4\times 4} \tag{1}$$

where X_g, Y_g, and Z_g represent the detected 3D point expressed in the global Manhattan frame, while X_b, Y_b, and Z_b represent the same point expressed in the camera body frame. The global Manhattan frame can be determined at the first camera frame in the FL-SLAM. The 4×4 rigid body transformation matrix $T_{gb} \in \mathrm{SE}(3)$ represents the 6-DoF pose of the camera, with $\mathbf{t}_{gb} \in \mathbb{R}^3$ representing the 3-DoF translational motion of ARKit and $R_{gb} \in \mathrm{SO}(3)$ representing the 3-DoF rotational motion of ARKit in the global Manhattan frame. We employ Apple ARKit 6-DoF poses to build an accurate and consistent 3D point cloud [21] with four range measurements in a single scan.

2) Manhattan World Mapping in Real-time: Algorithm 1 outlines the procedure for building a global MW map parallel to the X and Y axes of the global Manhattan frame using the four-point LiDARs as shown in Fig. 4. We project an accumulated 3D point cloud onto the floor in the MW, thus utilizing X_g and Y_g obtained from Eq. (1) as input.

Lines 1 to 7 describe extending the length of the already detected lines (walls) stored in the global MW map as new points are detected per scan. If the distance between the newly detected point and the line stored in the global MW map is less than 20 cm, we treat the currently detected point corresponds to the line stored in the global MW map and update the endpoint of the line.

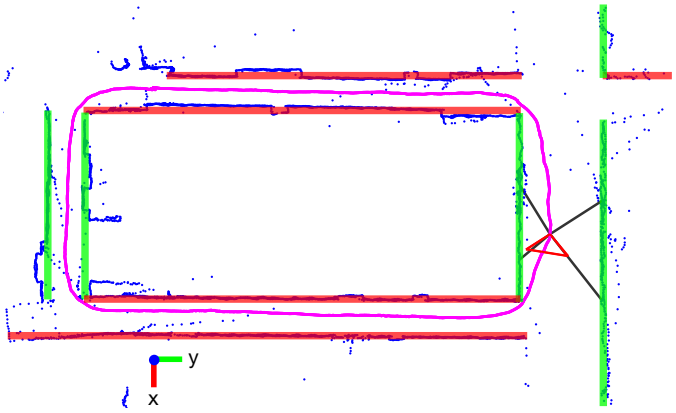

Fig. 4. Accumulated point cloud (blue dots) with sparse range measurements and Apple ARKit 6-DoF poses (magenta). We only utilize ARKit 6-DoF poses to build an accurate and consistent point cloud. We plot a global MW map (red and green) built with four-point LiDARs using the line RANSAC.

Algorithm 1 MW Mapping with Four-Point LiDARs

Input: X and Y coordinates of the accumulated point cloud in the global Manhattan map frame
Output: Global MW Map

1: **for** k time steps **do**
2: **if** already detected walls exist **then**
3: **if** currently detected points correspond to already detected walls **then**
4: Update endpoints of each wall
5: Remove the points used to extend each wall
6: **end if**
7: **end if**
8: **while** true **do**
9: Line fitting with line RANSAC
10: **if** the number of inlier points $<$ threshold **then**
11: **break**
12: **else**
13: Structure alignment in MW
14: Remove the points used to generate the new wall
15: **if** newly detected wall corresponds to one of the already detected walls **then**
16: Combine with the existing wall
17: **end if**
18: **end if**
19: **end while**
20: **end for**

Lines 8 to 20 describe detecting new lines and incrementally extending the global MW map as new dominant lines are detected per scan. We fit lines to the accumulated 2D point cloud using the line model-based RANSAC framework [25], which is suitable for robustly estimating the line model even in the presence of noise and outliers from sparse sensing. To accurately recognize dominant structural features such as walls in the structured environments, we add the newly detected line to the global MW map only if the number of inlier points of the line estimated by line RANSAC is more than 40. Then, if the angle difference between the currently detected line and one of the X and Y axes of the global Manhattan map frame is less than 5 degrees, we refit the slope of the line to orthogonal to the corresponding axis. We consider two lines as the same wall landmark if their offset difference is less than a certain threshold (in our experiments, 10 to 20 cm is appropriate) and they have the same alignment. We then merge the newly detected line with the corresponding line in the global MW map. To avoid detecting already detected lines, we remove the points used to generate the orthogonal lines from the accumulated 2D point cloud and repeat line RANSAC until we can no longer find a reliable new line from the current scan. In this way, we can build an accurate global MW map per scan from sparse sensing with the inexpensive four single-point LiDARs as shown in Fig. 4.

Note that the global MW map stores not only the walls detected by the four-point LiDARs, but also the walls, floor, and ceiling detected by RGB-D images in the same form of lines (offset, alignment, and endpoints), and there are no overlapping walls in the global MW map. Detecting orthogonal planes from RGB-D images for each frame can be referenced from L-SLAM [3]. By using a low-cost four-point LiDAR, we can detect walls beyond the range of detection of the RGB-D camera and supplement the global MW map of the previous L-SLAM, thereby producing a complete 3D global MW map.

C. Linear Four-Point LiDAR SLAM (FL-SLAM)

1) State Vector Definition: The state vector in the KF consists of the 3-DoF translational camera motion and 1-D distances (offset) of the orthogonal planar features in the global MW map with respect to the origin of the global Manhattan map frame C_g. We express the state vector $\mathbf{x}$ as follows:

$$\mathbf{x} = \left[\mathbf{p}^\top, m_1, ..., m_n\right]^\top \in \mathbb{R}^{3+n} \tag{2}$$

where $\mathbf{p} = \left[x\ y\ z\right]^\top \in \mathbb{R}^3$ denotes the 3-DoF camera translation expressed in the global Manhattan map frame C_g. The map $\mathbf{m}_i = \left[o_i\right]^\top \in \mathbb{R}^1$ denotes the 1-D distance (offset) of the orthogonal planes such as walls, floor, and ceiling, and n is the number of orthogonal planar features in the global MW map. When the new orthogonal planes (mostly newly detected walls) are discovered with either four-point LiDARs or an RGB-D camera, they are added to the global MW map and corresponding offset $\mathbf{m}_{new}$ is augmented to the end of the state vector $\mathbf{x}$.

2) Propagation Step: We predict the next step based on the 3-DoF translation estimated by LPVO [26] between the consecutive image frames. We propagate the 3-DoF translation with the process model $\mathbf{x}_k = \mathbf{x}_{k-1} + \left[\triangle\mathbf{p}_{k,k-1}^\top\ \mathbf{0}_{1\times n}\right]^\top$ where $\triangle\mathbf{p}_{k,k-1}$ is the estimated 3-DoF translation from LPVO. We assume the 1-D distance of orthogonal planes in the global MW map does not change in the process model.

3) Correction Step with Global MW Map: Algorithm 2 outlines the state update process when the current scene is an open space (Lines 2 to 5) and when it is not (Lines 6 to 9). We utilize 3-DoF rotational motion from ARKit when the scene is an open space where accurate and drift-free rotation cannot be obtained due to MF tracking failure. Since we define open space as a situation where we cannot detect vertical

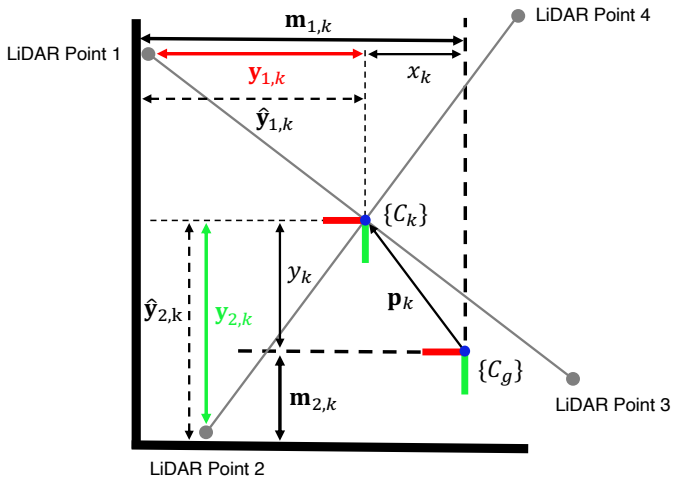

Fig. 5. Illustration of the Kalman filter components for FL-SLAM: the state vector ($\mathbf{p}_k$, $\mathbf{m}_k$) and observation model ($\mathbf{y}_k$) at k-th camera frame. C_g and C_k represent the origin of the global MW map frame and center of k-th camera frame, respectively.

Algorithm 2 State Update in Linear KF

for k time steps **do**
  if the current scene is an open space **then**
    Update X,Y translation with currently observed points from the four-point LiDARs
    Update Z translation with currently detected horizontal planes from the RGB-D images
    Add new landmarks to the end of the state
  else
    Update translation with currently detected planes from the RGB-D images
    Add new landmarks to the end of the state
  end if
end for

planes (walls) using an RGB-D camera, we determine each camera frame whether the current scene is an open space or not using the number of surface normal vectors (SNV) for each pixel used to track MF over time. If the number of SNV corresponding to the X and Y global Manhattan frame axes is less than the certain threshold (in our experiment, 10000 to 15000 is appropriate), the dominant walls of the two axes cannot be identified in the current RGB-D scene. In this way, the proposed FL-SLAM can obtain accurate and stable 3-DoF rotation in the entire trajectory so that we can utilize a linear KF [27], [28] with the state vector of the camera translation and 1-D offset of the walls.

In this open space, FL-SLAM updates the estimated X,Y translation from the LPVO of the current state through the points currently observed with the four-point LiDARs as shown in Fig. 5. A plane orthogonal to the Z-axis is always detectable using the RGB-D camera. Therefore, in open space, only X,Y translation needs to be corrected with LiDAR points observed through the four-point LiDARs. To update the current state vector, we match the currently observed point and the nearest wall in the global MW map only if the distance between them is less than 10 cm. For example, as shown in Fig. 5, LiDAR points 1 and 2 match each global wall, and we consider LiDAR points 3 and 4 as unreliable observations and do not use them for updating the state vector. We perform the KF update so that the residual, which is the difference between the actual measurement from LiDAR and the predicted measurement from the state vector, is zero for all matched points with global walls in the global MW map. The residual $\mathbf{r}_k$ at k-th camera frame and the predicted measurement model $\hat{\mathbf{y}}_k$ are expressed as:

$$\mathbf{r}_k = \mathbf{y}_k - \hat{\mathbf{y}}_k \tag{3}$$

$$\hat{\mathbf{y}}_k = \begin{bmatrix} \mathbf{m}_{1,k} - x_k \\ \mathbf{m}_{2,k} - y_k \\ \mathbf{m}_{3,k} - z_k \\ \vdots \end{bmatrix} \in \mathbb{R}^m \tag{4}$$

where $\mathbf{y}_k$ is the actual measurement from the LiDAR sensor, and x_k, y_k, and z_k are 3-DoF camera translation in the state vector. $\mathbf{m}_{3,k}$ is a global map orthogonal to the Z-axis of the global MW map frame, and corresponding $\mathbf{y}_{3,k}$ is an actual measurement for the floor or ceiling detected by an RGB-D camera. m is the number of matched orthogonal planar features in the global MW map. We set the measurement noise in the KF to 7 cm in the open spaces, reflecting the range measurement noise of four LiDARs.

When the current RGB-D scene is not an open space where we can detect surrounding walls with the RGB-D camera, we update the state vector in the KF by observing the distance between the orthogonal planes currently detected through the RGB-D camera, which is the same as the correction step of L-SLAM. When the vertical planes (walls) cannot be detected in the RGB-D images in wide and open spaces, we update 3-DoF translational motion and the global MW maps with currently observed range measurements from four LiDARs. In this manner, FL-SLAM can effectively update the 3-DoF translation and 1-D map in open spaces.

IV. Evaluation

We both qualitatively and quantitatively evaluate the proposed FL-SLAM on various author-collected RGB-D and corresponding sparse range measurement datasets obtained by the custom-built sensor rig as shown in Fig. 3. Since none of the existing datasets comprise both RGB-D images and sparse range measurements, we obtain the datasets by ourselves. We obtain RGB and depth images of 256 $\times$ 192 at 15 Hz using the Stray Scanner app on the iPhone 12 Pro Max while simultaneously acquiring the four range measurements corresponding to each RGB-D image using the four-point LiDAR. We connect four TF02-Pro LiDARs to a single Arduino Uno, connecting it to a laptop with Intel Core i5 (2.42 GHz) to obtain real-time distance measurements. We implement the proposed FL-SLAM in MATLAB on a desktop computer with an Intel Core i7 (2.90 GHz) and 16 GB memory.

To demonstrate improved performance compared with the L-SLAM, we collect real-world building-scale datasets that include various open spaces commonly found in typical university buildings or offices as shown in Fig. 6. As we cannot

TABLE I
EVALUATION RESULTS OF FDE (UNIT: M) ON AUTHOR-COLLECTED CLOSED-LOOP DATASETS.

Experiment	FL-SLAM (Ours)	L-SLAM	Graph-SLAM	DROID-SLAM	ORB-SLAM3	Length (m)
Square Corridor	<u>0.350</u>	38.426	1.055	**0.225**	0.588	92.266
Open Hallway 1	0.297	1.580	0.191	**0.070**	<u>0.077</u>	24.604
Open Hallway 2	0.190	×	0.802	<u>0.079</u>	**0.060**	44.160

TABLE II
EVALUATION RESULTS OF ATE RMSE (UNIT: M) ON AUTHOR-COLLECTED OPEN-LOOP DATASETS.

Experiment	FL-SLAM (Ours)	L-SLAM	Graph-SLAM	DROID-SLAM	ORB-SLAM3	Length (m)
L-shaped Corridor	**0.660**	1.990	29.373	<u>0.845</u>	1.510	52.032
U-shaped Corridor	**0.738**	<u>1.476</u>	3.177	2.657	3.846	64.361
Open Hallway 3	<u>0.390</u>	7.164	0.406	**0.326**	0.699	34.564

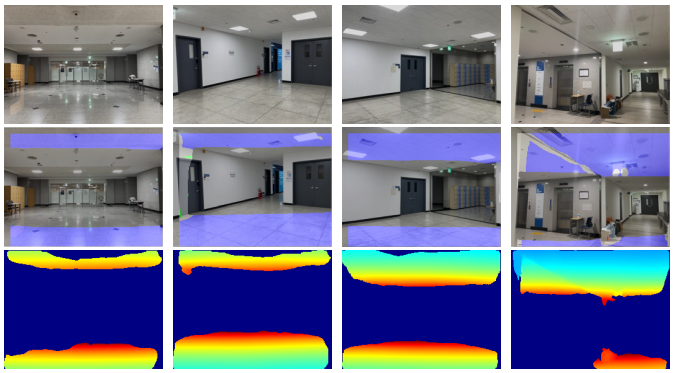

Fig. 6. Example images of open spaces in author-collected datasets. We overlay the tracked MF (floor and ceiling) on top of the RGB images in the second row. In the third row, there are few valid depth values in the depth images other than nearby areas such as the floor and ceiling.

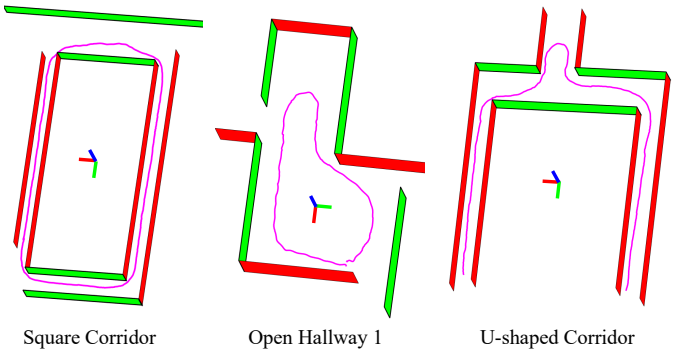

Fig. 7. Selected 3D global MW map built with four-point LiDARs, and the camera trajectory results (magenta) of the proposed FL-SLAM. The red and green walls are orthogonal to the X and Y axes of the global MW map frame, respectively. The floor and ceiling in blue are not shown for visibility.

obtain a ground truth trajectory, we experiment with three closed-loop sequences where the start and end points of the trajectories are the same to evaluate whether the proposed FL-SLAM accurately updates the camera translation based on the global MW map. We experiment with three other open-loop sequences to evaluate the entire trajectory by comparing it to the ARKit trajectory as ground truth. Although we cannot obtain absolute ground truth, it is well-known that Apple ARKit is very stable and accurate at short distances [21].

We compare our FL-SLAM with the state-of-the-art SLAM methods, namely DROID-SLAM [2], ORB-SLAM3 [1], and Graph-SLAM [19]. DROID-SLAM and ORB-SLAM3 are RGB-D SLAM methods, and Graph-SLAM is a 2D LiDAR SLAM approach with sparse sensing for the Crazyflie [8] nano-quadrotor. We utilize RGB-D images as input data for DROID-SLAM, ORB-SLAM, and L-SLAM. As the input data of Graph-SLAM, we use the same range measurements obtained by four-point LiDARs and X, Y translation, and orientation as the proposed FL-SLAM utilizes to update camera poses. For a fair comparison, we test each SLAM method made publicly available by the original authors from their official GitHub pages with default parameter settings.

We quantitatively evaluate the performance of five SLAM algorithms using the final drift error (FDE) metric, which is the end-point position error in meters at the three closed-loop datasets as shown in Table I. Since FL-SLAM utilizes ARKit orientation in open spaces for three open-loop datasets, we use the absolute trajectory error (ATE) metric for translational motion comparison at the three other open-loop datasets as shown in Table II. For Graph-SLAM, which is 2D SLAM, we compare only the X and Y translation with the ARKit trajectory. The best results are in bold, and the second-best results are underlined. As shown in Table I and Table II, FL-SLAM demonstrates comparable performance to other state-of-the-art SLAM algorithms that use loop detection by successfully updating the estimated translation from VO based on the global MW map constructed through four-point LiDAR. For a detailed analysis, we focus on three sequences: Square Corridor, Open Hallway 1, and U-shaped corridor.

A. Square Corridor

The Square Corridor is a long corridor over 92 meters in total length, which includes open spaces and a narrow corridor advantageous for L-SLAM to detect surrounding walls. Fig. 8 shows the open spaces that can be seen in the sequence. When the orthogonal planar features are insufficient to track the MF due to the short-range measurement of the RGB-D camera, L-SLAM fails significantly in estimating the camera trajectory. In contrast, our proposed FL-SLAM can successfully build

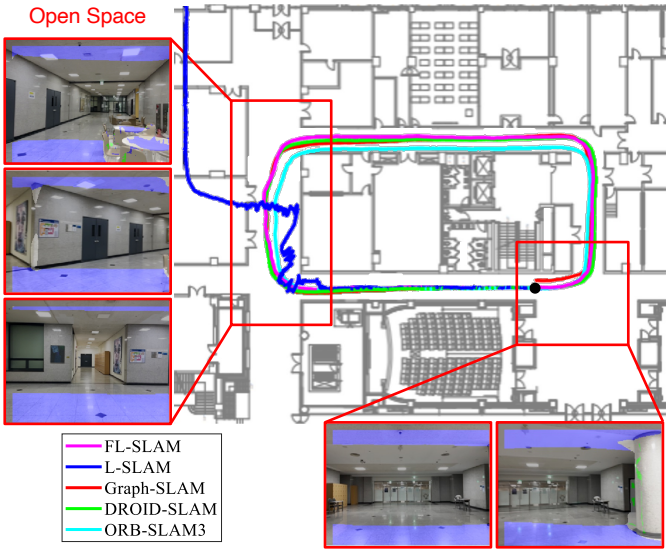

Fig. 8. Qualitative comparisons of the Square Corridor sequence. We overlay estimated trajectories with five SLAM methods on the floor plan, showing the location of the open spaces in the sequence. A black dot indicates the beginning and end of the sequence to evaluate the loop-closing performance.

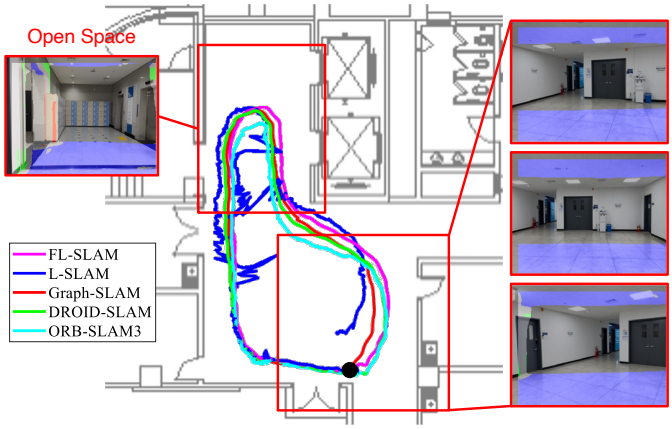

Fig. 9. Estimated trajectories with the proposed (magenta) and other SLAM approaches on the Open Hallway 1 sequence. Since an RGB-D camera looks at open spaces where vertical walls are not observable (red boxes), L-SLAM fails to estimate camera motion accurately while FL-SLAM does not.

a global MW map even in such open spaces using a low-cost four-point LiDAR as shown in Fig. 7 (left), allowing for accurate camera translation updates within the linear KF framework. We can obtain stable trajectories by using the rotation of ARKit in open spaces. We show comparable performance to ORB-SLAM and DROID-SLAM, which use the loop-detection technique by returning 0.350 m to the position starting with the Square Corridor sequence without additional use of loop-detection.

B. Open Hallway 1

Open Hallway 1 is a wide space commonly found indoors, consisting mostly of an open space environment. The global MW map generated using our four-point LiDAR and the resulting trajectories of FL-SLAM and other algorithms are shown in Fig. 7 (middle) and Fig. 9. Similar to the Square Corridor sequence, FL-SLAM achieves almost successful loop

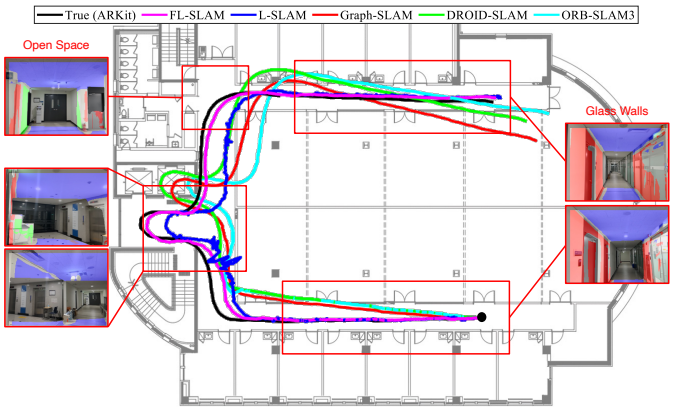

Fig. 10. Qualitative comparisons of the U-shaped Corridor sequence, overlaying estimated trajectories with six VIO/SLAM approaches on the floor plan. We employ Apple ARKit for the ground-truth (black), and the black dot represents the starting point of the sequence. The estimated path for FL-SLAM (magenta) matches the floor plan most consistently.

detection and estimates reasonable trajectories compared to other state-of-the-art SLAM algorithms.

C. U-shaped Corridor

The U-shaped Corridor is a long corridor with a total length of 64 meters, commonly found in indoor structured environments. One side of the long corridor of the sequence is almost glass, while the other side has white cement walls with few textures. Due to the glass walls, we obtain a noisy point cloud, which leads to performance degradation in Graph-SLAM. On the other hand, the proposed FL-SLAM effectively uses the MW assumption despite this inaccurate and noisy point cloud, allowing for the construction of a reliable global MW map as shown in Fig. 7 (right) and Fig. 11. The map produced by the four-point LiDAR for each frame is effective not only in open spaces but also in updating the short trajectory translation estimated by VO effectively in a long corridor by detecting far distant walls using the four-point LiDAR. The performance of feature-based SLAM such as ORB-SLAM3 and DROID-SLAM is degraded due to the lack of texture in indoor environments.

D. Ablation Study on the Use of Four-Point LiDARs

The blue-dotted trajectory in Fig. 11 is the result of simply replacing the 3-DoF rotation of L-SLAM with the 3-DoF rotation of the ARKit in open spaces where L-SLAM obtains incorrect rotational camera motion. This blue-dotted trajectory is the result of building a global MW map and updating the 3-DoF translation using only the orthogonal planes detected via RGB-D camera, such as L-SLAM, without using orthogonal walls detected by four-point LiDAR. The magenta color trajectory is our proposed complete FL-SLAM algorithm, which uses the rotation of ARKit in open spaces while also using the detected orthogonal walls by utilizing the four-point LiDAR to update the estimated translation by VO at every frame. Through this ablation study of the use of four-point LiDAR, we demonstrate that the good performance of the proposed FL-SLAM is not simply because of the rotation of ARKit,

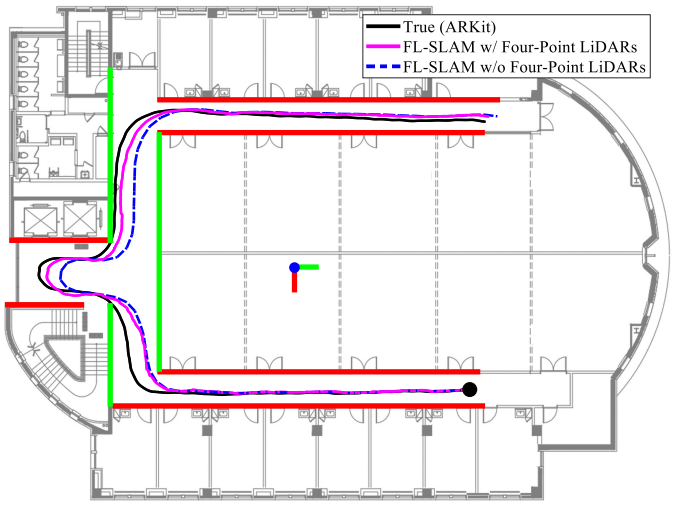

Fig. 11. Ablation study on the use of four-point LiDARs for FL-SLAM. We overlay the trajectories of FL-SLAM, ARKit, and the global MW map on the floor plan. We can build a consistent global MW map similar to the floor plan using the inexpensive four-point LiDARs with the proposed FL-SLAM.

but because we effectively map using four-point LiDAR and update the translation accurately based on it.

As shown in Fig. 11, with an RGB-D camera, it is possible to detect walls only within 5 m. When passing through a narrow corridor, it has only been possible to update the translation estimated by VO based on the side walls, and the front and back of the trajectory cannot be corrected. Therefore, using only an RGB-D camera makes it difficult to compensate for the drift caused by the nature of VO, as seen in the blue-dotted trajectory. However, since we use a four-point LiDAR capable of measuring up to 40 m, we can detect orthogonal walls even in the far distance and update the short trajectory translation estimated by VO to make it similar to the actual trajectory. Comparing the blue-dotted trajectory in Fig. 11 with the trajectory of L-SLAM (blue) in Fig. 10, the use of the rotation of ARKit also slightly helps to correct the shortened trajectory, but updating the translation through the wall detected by the four-point LiDAR has a significant impact.

V. Conclusion

We propose a new linear KF-based SLAM algorithm that complements the short measurement range limitations of conventional RGB-D cameras by effectively utilizing a low-cost four-point LiDAR that can measure up to 40 m and sense only four distance measurements per scan, resulting in SLAM with sparse sensing problem. We overcome this challenging problem by effectively using MW assumption to build a reliable global MW map with the four-point LiDARs. Through experiments in various indoor MW environments, we demonstrate that FL-SLAM significantly expands the application range of L-SLAM by accurately detecting walls beyond the range of detection of the RGB-D camera using the four-point LiDARs. Furthermore, we demonstrate comparable performance to other state-of-the-art SLAM methods without using a loop detection algorithm.

References

[1] C. Campos, R. Elvira, J. J. G. Rodríguez, J. M. Montiel, and J. D. Tardós, "Orb-slam3: An accurate open-source library for visual, visual–inertial, and multimap slam," *IEEE TRO*, 2021.
[2] Z. Teed and J. Deng, "Droid-slam: Deep visual slam for monocular, stereo, and rgb-d cameras," *Advances in neural information processing systems*, 2021.
[3] P. Kim, B. Coltin, and H. J. Kim, "Linear rgb-d slam for planar environments," in *ECCV*, 2018.
[4] J. Straub, O. Freifeld, and Rosman, "The manhattan frame model—manhattan world inference in the space of surface normals," *IEEE TPAMI*, 2017.
[5] P. Kim, H. Li, and K. Joo, "Quasi-globally optimal and real-time visual compass in manhattan structured environments," *IEEE Robotics and Automation Letters*, 2022.
[6] J. Zhang and S. Singh, "Loam: Lidar odometry and mapping in real-time." in *Robotics: Science and Systems*, 2014.
[7] 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.
[8] W. Giernacki, M. Skwierczyński, W. Witwicki, P. Wroński, and P. Kozierski, "Crazyflie 2.0 quadrotor as a platform for research and education in robotics and control engineering," in *International Conference on MMAR*, 2017.
[9] A. Flint, D. Murray, and I. Reid, "Manhattan scene understanding using monocular, stereo, and 3d features," in *ICCV*, 2011.
[10] T. Schops, T. Sattler, and M. Pollefeys, "Bad slam: Bundle adjusted direct rgb-d slam," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019.
[11] F. Shu, J. Wang, A. Pagani, and D. Stricker, "Structure plp-slam: Efficient sparse mapping and localization using point, line and plane for monocular, rgb-d and stereo cameras," in *ICRA*, 2023.
[12] Z. Zhu, S. Peng, V. Larsson, W. Xu, and H. Bao, "Nice-slam: Neural implicit scalable encoding for slam," in *CVPR*, 2022.
[13] A. Rosinol, J. J. Leonard, and L. Carlone, "Nerf-slam: Real-time dense monocular slam with neural radiance fields," *arXiv:2210.13641*, 2022.
[14] F. Moosmann and C. Stiller, "Velodyne slam," in *IEEE IV*, 2011.
[15] K. R. Beevers and W. H. Huang, "Slam with sparse sensing," in *ICRA*, 2006.
[16] T. N. Yap and C. R. Shelton, "Slam in large indoor environments with low-cost, noisy, and sparse sonars," in *ICRA*, 2009.
[17] T. Chathuranga, M. Padmal, D. Bibile, P. Jayasekara, and N. Kottege, "Sensor deck development for sparse localization and mapping for micro uavs to assist in disaster response."
[18] K. Murphy and S. Russell, "Rao-blackwellised particle filtering for dynamic bayesian networks," *Sequential Monte Carlo methods in practice*, 2001.
[19] H. Zhou, Z. Hu, S. Liu, and S. Khan, "Efficient 2d graph slam for sparse sensing," in *IROS*, 2022.
[20] J. M. Coughlan and A. L. Yuille, "Manhattan world: Compass direction from a single image by bayesian inference," in *Proceedings of the seventh IEEE international conference on computer vision*, 1999.
[21] P. Kim, J. Kim, M. Song, Y. Lee, M. Jung, and H.-G. Kim, "A benchmark comparison of four off-the-shelf proprietary visual–inertial odometry systems," *Sensors*, 2022.
[22] C. J. Taylor and A. Cowley, "Parsing indoor scenes using rgb-d imagery," in *Robotics: Science and Systems*, 2013.
[23] J.-C. Bazin, C. Demonceaux, P. Vasseur, and I. S. Kweon, "Motion estimation by decoupling rotation and translation in catadioptric vision," *CVIU*, 2010.
[24] D. Holz, S. Holzer, R. B. Rusu, and S. Behnke, "Real-time plane segmentation using rgb-d cameras," in *Robot Soccer World Cup*, 2011.
[25] M. A. Fischler and R. C. Bolles, "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography," *Communications of the ACM*, 1981.
[26] P. Kim, B. Coltin, and H. J. Kim, "Low-drift visual odometry in structured environments by decoupling rotational and translational motion," in *ICRA*, 2018.
[27] P. Kim, H. Lim, and H. J. Kim, "Visual inertial odometry with pentafocal geometric constraints," *International Journal of Control, Automation and Systems*, 2018.
[28] J. M. Pak and C. K. Ahn, "State estimation algorithms for localization: A survey," *International Journal of Control, Automation and Systems*, 2023.