Linear RGB-D SLAM for Structured Environments

Kyungdon Joo, *Member, IEEE,* Pyojin Kim, *Member, IEEE,* Martial Hebert, *Member, IEEE,*
In So Kweon, *Member, IEEE,* and Hyoun Jin Kim, *Member, IEEE*

Abstract—We propose a new linear RGB-D simultaneous localization and mapping (SLAM) formulation by utilizing planar features of the structured environments. The key idea is to understand a given structured scene and exploit its structural regularities such as the Manhattan world. This understanding allows us to decouple the camera rotation by tracking structural regularities, which makes SLAM problems free from being highly nonlinear. Additionally, it provides a simple yet effective cue for representing planar features, which leads to a linear SLAM formulation. Given an accurate camera rotation, we jointly estimate the camera translation and planar landmarks in the global planar map using a linear Kalman filter. Our linear SLAM method, called L-SLAM, can understand not only the Manhattan world but the more general scenario of the Atlanta world, which consists of a vertical direction and a set of horizontal directions orthogonal to the vertical direction. To this end, we introduce a novel tracking-by-detection scheme that infers the underlying scene structure by Atlanta representation. With efficient Atlanta representation, we formulate a unified linear SLAM framework for structured environments. We evaluate L-SLAM on a synthetic dataset and RGB-D benchmarks, demonstrating comparable performance to other state-of-the-art SLAM methods without using expensive nonlinear optimization. We assess the accuracy of L-SLAM on a practical application of augmented reality.

Index Terms—Linear SLAM, Manhattan World, Atlanta World, RGB-D Image, Bayesian Filtering, Scene Understanding.

1 Introduction

Visual simultaneous localization and mapping (visual SLAM) is the problem of estimating the six degrees of freedom (DoF) rotational and translational camera motion while simultaneously building a map of a surrounding unknown environment from a sequence of images. Visual SLAM methods have been widely studied within the robotics and computer vision communities for several decades [1]. They are the fundamental building blocks for various computer vision applications such as autonomous robots and virtual and augmented reality (VR/AR) [2]–[4].

Typical visual SLAM approaches, such as DVO-SLAM [5] and ORB-SLAM2 [6], have shown promising results in general environments with rich texture. They usually rely on low-level features such as point features in the vicinity of the texture. Thus, they fare poorly in texture-less or feature-less scenes, which are commonly encountered in indoor environments with large planar structures. To alleviate this limitation, recent SLAM methods [7]–[9] utilize additional high-level geometric primitives such as planar features of structured indoor environments.

Most indoor environments not only consist of planar structures but also exhibit formal and regular forms. For instance, from the indoor structure of the building (*e.g.*, room layout) to object (*e.g.*, furniture), they can be represented by a set of cuboid structures of various sizes (see Fig. 1). In computer vision, most of these structures, in the shape of cuboid, are commonly approximated using the Manhattan world (MW) assumption [11], which is defined by three orthogonal directions. In addition to the MW assumption, there exist several structural assumptions defined according to the level of constraint, *e.g.*, the Atlanta world (AW) assumption [12] based on a semi-orthogonality, and a mixture of Manhattan frames [13] composed of multiple Manhattan structures. By its orthogonality and simplicity, the structural assumptions have been exploited in various computer vision applications, such as 3D reconstruction [14], and scene

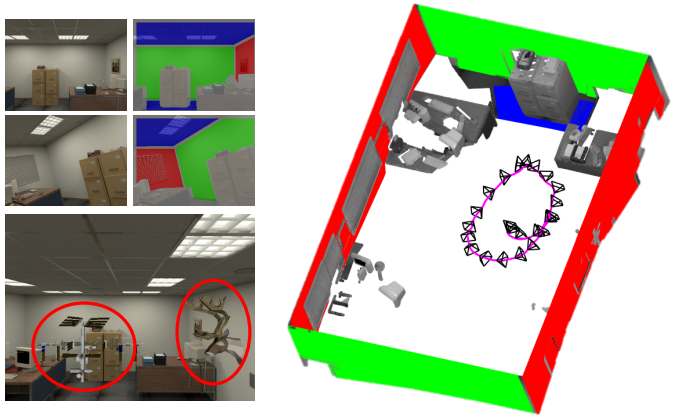

Fig. 1. **Linear SLAM on the ICL-NUIM dataset** [10]. The proposed L-SLAM generates a consistent global planar map using a linear Kalman filter framework instead of an expensive pose graph optimization. *Left top*: The tracked planar features following the Manhattan structure are overlaid on top of the RGB images. *Left bottom*: AR application results in the red circles with the international space station and Elk's head 3D models. *Right*: Global planar map and non-planar regions are rendered by back-projecting the RGB-D images from the estimated camera trajectory with L-SLAM. We omit the ceiling planar features for visibility.

- *K. Joo is with the Artificial Intelligence Graduate School and the Department of Computer Science and Engineering, UNIST, Ulsan, Republic of Korea. E-mail: kdjoo369@gmail.com, kyungdon@unist.ac.kr*
- *P. Kim is with the Department of Mechanical Systems Engineering, Sookmyung Women's University, Seoul, Republic of Korea. E-mail: pjinkim@sookmyung.ac.kr*
- *M. Hebert is with the School of Computer Science and the Robotics Institute, CMU, Pittsburgh, USA. E-mail: hebert@cs.cmu.edu*
- *I. S. Kweon is with the School of Electrical Engineering, KAIST, Daejeon, Republic of Korea. E-mail: iskweon77@kaist.ac.kr*
- *H. J. Kim is with the School of Mechanical and Aerospace Engineering, Seoul National University, Seoul, Republic of Korea. E-mail: hjinkim@snu.ac.kr*
- *(Corresponding author: Pyojin Kim.)*

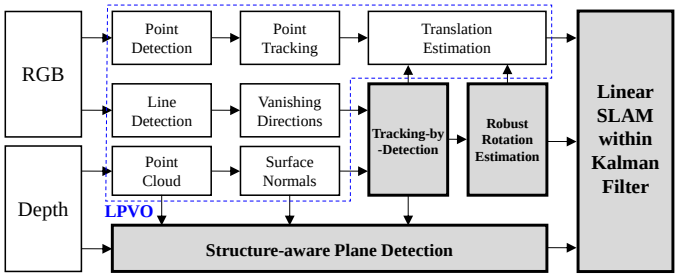

Fig. 2. **Overview of the proposed L-SLAM algorithm**. We highlight key components (boxes filled in grey) of the proposed approach for our main contributions. Blue-dashed box indicates the components of LPVO [18].

layout inference [15, 16]. In particular, visual odometry (VO) and SLAM approaches [17, 18] can achieve drift-free rotational motion of the camera by utilizing the structural patterns observed repeatedly and consistently, resulting in an improved location estimation accuracy, which is our motivation.

In this work, we propose a novel *linear RGB-D SLAM* approach, referred to as L-SLAM[1], which efficiently utilizes dominant directions of a given structured indoor scene – structural regularities. Based on the understanding of the underlying structural regularities (*i.e.*, dominant directions), we jointly estimate camera position and planar landmarks in the global planar map within a linear Bayesian filter, as shown in Figs. 1 and 2. Concretely, we first recognize and track the dominant directions of the structured scene – dominant directions satisfying the Manhattan or Atlanta worlds. It allows us to decompose the rotational motion, which is the main source of nonlinearity in SLAM formulation. Given the absolute camera rotation, L-SLAM identifies the horizontal and vertical planes supporting the structured environments and measures the distance to these planes from the current camera pose. With the distance measurements, we simultaneously update the 3-DoF camera translation and the 1-D distance of the associated planar landmarks in the map within a linear Kalman filter (KF) framework. The following is a summary of our main contributions:

- We propose the first linear and unified RGB-D SLAM approach for two different structural regularities – the Manhattan and Atlanta worlds.
- We introduce a novel tracking-by-detection scheme that estimates the underlying unknown structural regularities (*i.e.*, Atlanta structure) of a scene, which allows us to calculate the camera rotation robustly.
- By exploiting efficient parametrization for the structural regularities, we compactly represent an observed plane as the association to the supporting direction and 1-D distance (*i.e.*, planar landmark), enabling us to formulate the measurement model as a linear model.
- Our method has been validated through extensive experiments on both synthetic and real-world RGB-D benchmark datasets. In addition, we show the applicability of L-SLAM to augmented reality (AR).

1. In general, we call the proposed linear SLAM as L-SLAM, and according to the structured patterns employed, we will denote L-SLAM for the Manhattan world (MW) as L_{MW}-SLAM, and L-SLAM for the Atlanta world (AW) as L_{AW}-SLAM throughout in this paper.

This paper presents a unified linear SLAM approach to our previous conference works [4, 19], utilizing two different structural assumptions – the Manhattan and Atlanta world assumptions, respectively. Specifically, we describe and analyze the two different structured assumptions – their definition and representation (parametrization). Based on this fact, we seamlessly integrate two SLAM approaches into a unified method and show its effectiveness through qualitative and quantitative evaluations and AR applications.

2 Related Work

Visual SLAM methods have been actively studied within the robotics and computer vision communities for the past two decades owing to its importance in various applications, from autonomous UAV to AR. From the vast literature on the visual SLAM, we provide a brief overview of state-of-the-art typical approaches and some SLAM methods utilizing planar structures.

Many successful SLAM algorithms have been developed using either point features (*i.e.*, indirect approach) or high gradient pixels (*i.e.*, direct method). Representatives of these are direct LSD-SLAM [20], DSO [21], and feature-based ORB-SLAM2 [6]; however, their performance can be severely degraded in challenging low-texture environments.

Some works in the early years of SLAM research exploited planes as additional feature within an extended Kalman filter (EKF)-based SLAM approaches [22]. In [23, 24], tracked points lying on the same plane were reformulated as planar features to reduce the state size in EKF-SLAM. Servant *et al*. [25] included planar features in the EKF state vector with a priori structural information. Martínez-Carranza and Calway [26] proposed a unified parametrization for both points and planes within an EKF-based monocular SLAM. Weingarten and Siegwar [27] used planar features extracted from 2D laser scanner in an EKF-based SLAM. However, these EKF-SLAM methods utilizing planar features have some problems. They cannot avoid local linearization error [28] because the combined estimation of camera rotation and translation results in non-linearity of the measurement model. In addition, because both distance and orientation are used to represent the planar features, the size of the state vector and covariance matrix (computational complexity) grows rapidly over time, which limits applications to a room-scale environment.

Several recent planar SLAM studies have applied graph-based SLAM [29]–[31], which is a nonlinear and non-convex optimization problem [17]. To avoid singularities in pose graph optimization, Kaess [32] presented a minimal plane representation of infinite planes. Ma *et al*. [33] tracked keyframe camera pose and global plane model by performing direct image alignment and global graph optimization. Yang *et al*. [7] performed graph-based SLAM with the plane measurements coming from scene layout understanding using convolutional neural networks (CNN). In [9], a keyframe-based factor graph optimization was performed to achieve real-time operation on a CPU only. Although these approaches demonstrate superior estimation results in structured environments, they require expensive and difficult pose graph optimization since they estimate the

camera translation and rotation together, which is a main source of nonlinearity in SLAM formulation [17].

In general, planar features in man-made environments have structural forms whose properties can be utilized as a priori information or structural regularities in SLAM and navigation processes [34]–[36]. Struab *et al*. [37] utilized the Manhattan structure to estimate rotation robust against angular drift. Kim *et al*. [18, 38] tracked the Manhattan frame of a given scene to estimate drift-free rotational motion, which enables the de-coupling of translational motion. Le and Košecka [8] proposed a planar RGB-D SLAM method that estimates the camera rotation by identifying local Manhattan frames between subsequent frames and then infers 2-DoF camera translation in a graph SLAM framework. Recently, Li *et al*. [35] leveraged the structural regularity of the Atlanta world for a monocular SLAM approach. Zou *et al*. [36] proposed a visual-inertial odometry that utilizes Atlanta structure by using line features with prior orientation. Although a few studies have recently attempted to utilize the structural regularities for visual navigation, there is a lack of sufficient experiments and validation results in various structured environments.

To the best of our knowledge, this is the first *linear RGB-D SLAM* approach that fully utilizes the structural features such as the Manhattan and Atlanta worlds without any prior knowledge. The most relevant planar SLAM approach to the proposed L-SLAM is [8], which first estimates the 3-DoF camera rotation by recognizing the piece-wise planar models, and utilizes graph SLAM optimization to recover the 2-DoF camera translation. However, in contrast to the proposed L-SLAM which, estimates full 6-DoF camera motion, there is an assumption that the translational motion of the camera is always planar.

3 Problem Statement

In this section, we briefly present an overview of the proposed L-SLAM (Sec. 3.1) and provide preliminary information required for the proposed method (Sec. 3.2).

3.1 Overview

Given an RGB-D sequence as input, we propose a linear RGB-D SLAM framework for structured environments (L-SLAM) that jointly estimates camera pose and planar landmarks in the global planar map. Specifically, we first recognize the unknown structural regularities (dominant directions satisfying the Manhattan or Atlanta worlds) of a given scene using the tracking-by-detection scheme, which enables us to robustly estimate and decompose the rotational motion (Sec. 4). Based on this structural understanding, we identify the horizontal and vertical planes supporting the structural regularities and formulate a linear SLAM framework using the distance to these planes from the current camera pose (Sec. 5).

3.2 Preliminary

Before the technical details, we introduce structural assumptions and their representations (*i.e.*, parametrizations) in Sec. 3.2.1, which are key factors in the proposed L-SLAM. We then give a brief description of the previous LPVO algorithm [18] that estimates the rotational motion and initial translational motion under the Manhattan world in Sec. 3.2.2. Our L-SLAM utilizes several modules in LPVO and extend them into a linear SLAM framework for the Atlanta world as well as the Manhattan world.

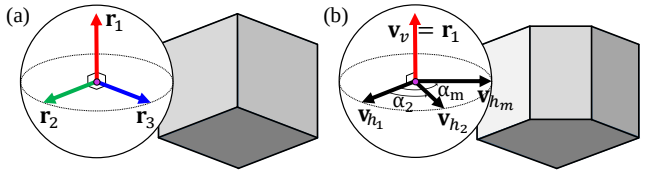

Fig. 3. **Manhattan world vs. Atlanta world**: (a) Manhattan frame represented by a rotation matrix $\mathbf{R} = [\mathbf{r}_1, \mathbf{r}_2, \mathbf{r}_3] \in SO(3)$ and the corresponding scene satisfying the Manhattan world. (b) Atlanta frame modeled by a rotation matrix $\mathbf{R}$ and an angle set $\{\alpha_m\}_{m=2}^{M}$ [42] and a scene structure following the Atlanta world.

3.2.1 Structural Representation

Manhattan World (Manhattan Frame). Man-made environments such as indoor objects and buildings have structural forms composed of orthogonal and parallel planes, which are commonly based on the Manhattan world (MW) assumption [11] in computer vision and robotics. Under the MW assumption, three orthogonal directions are used to describe a scene structure. These are referred to as the *Manhattan frame (MF)* or *Manhattan directions* and can be simply represented by a rotation matrix $\mathbf{R} = [\mathbf{r}_1, \mathbf{r}_2, \mathbf{r}_3] \in SO(3)$, as shown in Fig. 3(a). We denote MF as $\mathbf{R}_{\text{MF}}$.

By virtue of its simplicity, MF estimation of structured environments is commonly utilized as an essential part in high-level computer vision tasks such as scene understanding [16, 39], and layout estimation [40, 41]. In particular, given a scene following the Manhattan structure, tracking MF enables the estimation of drift-free rotation in SLAM framework and VO approach [4, 18]. However, the MW assumption is not suitable for a wide range of man-made structures whose horizontal directions are not orthogonal to each other such as the non-orthogonal walls of the Pentagon.

Atlanta World (Atlanta Frame). To alleviate the limited expression of the Manhattan world, Schindler *et al*. [12] proposed Atlanta world (AW), in which the horizontal directions are orthogonal to the vertical (typically gravity) direction; however, in contrast to the Manhattan world, these horizontal directions do not have to be orthogonal to each other. Thanks to these geometric characteristics, the Atlanta structure is a minimal model for representing the maximum range of man-made environments, permitting the handling of a wider range of scenes.

We can represent an Atlanta structure as a set of unit direction vectors $\mathcal{V} = \{\mathbf{v}_v, \mathbf{v}_{h_1}, \mathbf{v}_{h_2}, \cdots, \mathbf{v}_{h_M}\} = \{\mathbf{v}_m\}_{m=1}^{M+1}$ that consists of a vertical vector $\mathbf{v}_v = \mathbf{v}_1$ and a set of M horizontal vectors $\mathbf{v}_{h_m} = \mathbf{v}_{m+1}$, where $\mathbf{v}_v \perp \mathbf{v}_{h_m}$ (called the AW constraint) for $m \in \{1, \cdots, M\}$. We refer to this direction set $\mathcal{V}$ as the *Atlanta frame (AF)* or *Atlanta directions*. To represent AF, we follow the efficient AF parametrization proposed by Joo *et al*. [42, 43]. Specifically, their AF representation leverages the rotation matrix $\mathbf{R} = [\mathbf{r}_1, \mathbf{r}_2, \mathbf{r}_3] \in SO(3)$ to represent the vertical direction and the first horizontal direction by $\mathbf{r}_1$ and $\mathbf{r}_2$, *i.e.*, $\mathbf{v}_v = \mathbf{r}_1$ and $\mathbf{v}_{h_1} = \mathbf{r}_2$,

where $\mathbf{v}_{h_1}$ acts as a reference location. Then, each additional horizontal direction $\mathbf{v}_{h_m}$ can be defined as a single angle parameter α_m by rotating $\mathbf{v}_{h_1}$ by the angle α_m around the axis $\mathbf{v}_v$, as depicted in Fig. 3(b). Thus, we can represent an AF $\mathcal{V}$ by $\{\mathbf{R}, \{\alpha_m\}_{m=2}^{M}\}$, which includes the AW constraint without explicit conditions and allows for the formulation of a linear SLAM adapted to the AW assumption (*cf*., Sec. 5.2).

3.2.2 Line and Plane-based Visual Odometry

We briefly summarize the *Line and Plane based Visual Odometry* (LPVO) [18]. LPVO has two main steps: 1) structural regularities – limited to Manhattan – are tracked to obtain the drift-free camera rotation with an $\mathrm{SO}(3)$-manifold constrained mean shift algorithm, and 2) estimation of camera translation by minimizing a de-rotated reprojection error from tracked points.

Rotation Estimation. The core of the drift-free rotation estimation in LPVO is to track the MF jointly from both lines and planes by exploiting structural regularities. Lines from RGB images and surface normal vectors from depth images are simultaneously used to estimate the drift-free camera orientation accurately and stably even when looking at only one or two planes. Given the density distribution of vanishing directions from lines and surface normals from depth on the Gaussian sphere $\mathbb{S}^2$, LPVO infers the mean of the directional vector distribution around each dominant Manhattan direction through a mean shift algorithm with a Gaussian kernel in the tangent plane $\mathbb{R}^2$ [44]. LPVO applies the Riemann exponential map to transform the mean shift results back to the Gaussian sphere $\mathbb{S}^2$ from the tangential plane $\mathbb{R}^2$.

The three modes $\hat{\mathbf{r}}_1, \hat{\mathbf{r}}_2, \hat{\mathbf{r}}_3$ ($\hat{\mathbf{r}}_i \in \mathbb{R}^3$) found by the mean shift algorithm are projected onto the $\mathrm{SO}(3)$ manifold because each Manhattan direction is independently updated. To satisfy the MW constraints (*i.e.*, full-orthogonality), LPVO employs the singular value decomposition (SVD):

$$\mathbf{R}_{\mathrm{MF}} = \mathbf{U}\mathbf{V}^\top, \qquad (1)$$
$$[\mathbf{U}, \mathbf{D}, \mathbf{V}] = \mathrm{SVD}(\begin{bmatrix}\lambda_1\hat{\mathbf{r}}_1 & \lambda_2\hat{\mathbf{r}}_2 & \lambda_3\hat{\mathbf{r}}_3\end{bmatrix}),$$

where λ_i is a weighting factor of how certain the observation of a direction is [44]. In this manner, LPVO can keep track of the Manhattan structures in every frame, allowing the drift-free 3-DoF rotational motion of the camera with respect to the world coordinate of the Manhattan world.

Translation Estimation. LPVO transforms feature correspondences between consecutive frames into a pure translation by making use of the drift-free rotation estimation in the previous step. The 3-DoF translation motion, which minimizes the residual vectors of all tracked feature points with and without depth, can be obtained by solving the following optimization problem:

$$\mathbf{t}^* = \arg\min_{\mathbf{t}} \sum_{i=1}^{N_k} \left(r_{i_1}(\mathbf{t})\right)^2 + \left(r_{i_2}(\mathbf{t})\right)^2 + \sum_{i=1}^{N_u} \left(r'_i(\mathbf{t})\right)^2, \qquad (2)$$

where $r_{i_1}(\mathbf{t}), r_{i_2}(\mathbf{t})$ and $r'_i(\mathbf{t})$ are the de-rotated reprojection error with the number of tracked features with known N_k and unknown N_u depth information, respectively (for further detailed derivation of Eq. (2), see [18]). LPVO can obtain the 3-DoF translational motion of the camera by minimizing the de-rotated reprojection error from the tracked points, which is only a function of the translational camera motion $\mathbf{t}$. We will use the estimated translational motion as an initial translation in the proposed linear SLAM formulation.

It should be noted that LPVO estimates only camera pose under the *MW assumption*. We basically utilize the overall pipeline of LPVO and extend it into a novel linear SLAM framework for the general structured environments (the Atlanta structure), as shown in Fig. 2.

4 Structural Regularities Understanding

This section describes how to recognize the unknown structural regularities of structured environments and robustly estimate camera rotation from the structural regularities. The MW assumption, a subset of the AW assumption, consists of three orthogonal directions (only two horizontal directions); that is, their orientation relation is fixed, and only a tracking module is required once recognized like LPVO [18]. In contrast, the AW assumption does not impose any pre-determined number of horizontal directions nor any prior in their relative orientations. Thus, estimating the Atlanta structure of a scene itself is a challenging and non-trivial task. To this end, we first introduce a tracking-by-detection algorithm for the Atlanta structure, where the tracking algorithm is complemented by a detection module that identifies new or missing directions and makes the tracking robust (Sec. 4.1). Based on the estimated Atlanta structure by the tracking-by-detection, we then estimate camera rotational motion (Sec. 4.2).

4.1 Tracking-by-Detection Algorithm

We propose a new tracking-by-detection framework that estimates the underlying Atlanta structure of a scene, called the *global Atlanta frame* $\mathcal{V}_G$ (see Fig. 4). The global AF $\mathcal{V}_G$ contains all of the Atlanta directions observed and their activation label $\mathbf{l}_G$[2].

Concretely, given the surface normals and vanishing directions of the k-th frame, denoted as $\mathcal{N}^k$, we first independently track and detect the AFs. We then associate the tracked and detected AFs into one unified AF. We call this unified AF *local Atlanta frame* $\mathcal{V}_L^k$ describing the Atlanta structure of the current (local) frame. The local AF $\mathcal{V}_L^k$ includes its association label $\mathbf{l}_L^k$ that stores the index of the corresponding Atlanta direction in the global AF. Given the estimated local AF $\mathcal{V}_L^k$, we update the global AF $\mathcal{V}_G$ and its associated activation label $\mathbf{l}_G$ and compute the association label $\mathbf{l}_L^k$. The detailed procedure of the proposed tracking-by-detection is formalized in Alg. 1. In the Manhattan world, we perform the tracking module only after initializing the Manhattan structure.

4.1.1 Atlanta Frame Tracking

We track each Atlanta direction via a mean shift algorithm in the tangent plane with a Gaussian kernel similarly to the density-based dominant direction tracking [44]. Given the surface normals and vanishing directions distributed on a unit sphere $\mathcal{N}^k$ and an initial Atlanta direction $\mathbf{v}_L^l \in \mathcal{V}_L^{k-1}$,

2. A binary activation label is associated with each direction in the global AF. Each label is set to 1 if its corresponding Atlanta direction in the global AF is activated, otherwise 0.

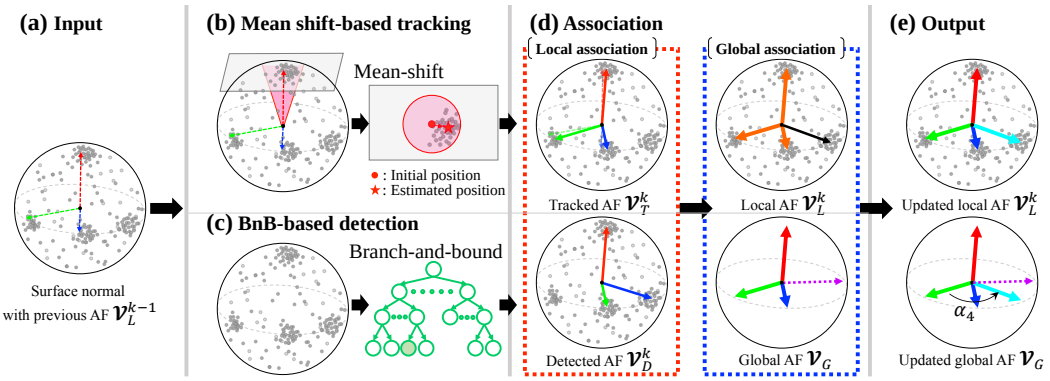

Fig. 4. **Overview of the tracking-by-detection scheme**. (a) Given a surface normal distribution $\mathcal{N}^k$ at the k-th frame with the previous local AF $\mathcal{V}_L^{k-1}$ (dashed arrows), we independently perform (b) mean shift-based AF tracking and (c) BnB-based AF detection. (d) We then apply the association step; associate tracked AF $\mathcal{V}_T^k$ and detected AF $\mathcal{V}_D^k$ into the local AF $\mathcal{V}_L^k$ (local association), where the orange arrows indicate the associated directions and the black arrow denotes a potential direction. In the global association, we associate the local AF $\mathcal{V}_L^k$ with the global AF $\mathcal{V}_G$, in which the typical arrows represent activated directions and the dashed purple arrow describes the de-activated direction. (e) As output, we obtain the updated local AF and global AF, where a new horizontal direction (cyan arrow) is born.

Algorithm 1 Tracking-by-detection for Atlanta frame

Input: Surface normal sequence $\mathcal{N}^k$.
Output: Local AF $\mathcal{V}_L^k$ with its association label $\mathbf{l}_L^k$ and global AF $\mathcal{V}_G$ with its activation label $\mathbf{l}_G$.

Detect AF $\mathcal{V}_D^1$ on $\mathcal{N}^1$
$\mathcal{V}_G \leftarrow \mathcal{V}_D^1$, $\mathcal{V}_L^1 \leftarrow \mathcal{V}_D^1$ ▷ Initialize $\mathcal{V}_G$ and $\mathcal{V}_L^1$
Initialize $\mathbf{l}_G$ and $\mathbf{l}_L^1$
$c_D = 1$ ▷ Initialize non-detection counter
for surface normal $\mathcal{N}^k (k \geq 2)$ **do**
  $\mathcal{V}_T^k \leftarrow \mathcal{V}_L^{k-1}$ ▷ Initialize seed for tracking
  Do mean shift-based AF tracking $\mathcal{V}_T^k$ on $\mathcal{N}^k$ ▷ *Tracking* step
  if $|\mathcal{V}_T^k| < 2$ **or** $c_D > 30$ **then**
    Detect AF $\mathcal{V}_D^k$ on $\mathcal{N}^k$ ▷ *Detection* step
    $c_D = 0$ ▷ Reset non-detection counter
  end if
  Do local association (*cf.*, Alg. 2)
  Do global association (*cf.*, Alg. 3)
  Force Atlanta world constraint on $\mathcal{V}_L^k$
  $c_D = c_D + 1$ ▷ Count accumulated non-detection frame
end for

we project the surface normals and vanishing directions into the tangent plane at $\mathbf{v}_L^l$, and infer the mean of the directional vector distribution around the projection of each Atlanta direction $\mathbf{v}_L^l$, as depicted in Fig. 4(b). After applying a mean shift [45], we estimate the tracked Atlanta direction $\mathbf{v}_T^t$ by projecting the mean estimate onto the unit sphere (for more details, see [44]). If $\mathbf{v}_T^t$ does not cover a given normal distribution with a certain percentage of the total number of normals, we discard this direction, which will be de-activated during the association. We denote the tracked AF as $\mathcal{V}_T^k = \{\mathbf{v}_T^t\}_{t=1}^{N_T}$, where N_T is the number of tracked Atlanta directions.

4.1.2 Robust Atlanta Frame Detection

We run an AF detection process separately from the tracking, as shown in Fig. 4(c). Through the detection step, we can find potential directions corresponding to lost Atlanta directions or new Atlanta directions. This strategy results in more stable tracking and enables sustainable long-term tracks [46]. Given the input normal distribution $\mathcal{N}^k$, we detect the AF $\mathcal{V}_D^k = \{\mathbf{v}_D^d\}_{d=1}^{N_D}$ that best describes the current normal distribution by means of the number of inliers. For this purpose, we utilize a branch-and-bound based AF estimation method [42] that guarantees global optimality and robustly estimates the AF even in a high noise situation. For the sake of computational efficiency, we detect the AF with only two horizontal directions ($M = 2$) and run the detection step only when the number of tracked Atlanta directions is less than 2 or periodically with a pre-defined frame interval (> 30 in all of our experiments).

4.1.3 Atlanta Frame Association

The association stage consists of three steps: local association, global association, and maintaining the AW constraint. In the local association step, we associate the tracked AF $\mathcal{V}_T^k$, and the detected AF $\mathcal{V}_D^k$ into one unified local AF $\mathcal{V}_L^k$ for the current frame. In the global association step, we associate the local AF $\mathcal{V}_L^k$ with the global AF $\mathcal{V}_G$, *i.e.*, updating the global AF $\mathcal{V}_G$ and its activation labels, where the global AF is updated by three operations: birth, revival, and death (see example in Fig. 6(a)). In the last step, we enforce the AW constraint on the local AF $\mathcal{V}_L^k$.

Local Association. Considering the tracked AF $\mathcal{V}_T^k$ as the initial local AF, we associate the detected AF $\mathcal{V}_D^k$ with the local AF and find potential directions for updating the global AF. For each detected direction $\mathbf{v}_D^d \in \mathcal{V}_D^k$, we measure its angular distance to the tracked AF, *i.e.*, $\angle(\mathbf{v}_D^d, \mathbf{v}_T^t)$, where $\angle(\cdot,\cdot)$ denotes the angle between two vectors. We then find the closest tracked direction and associate two directions $\mathbf{v}_D^d$ and $\mathbf{v}_T^t$ if their distance is less than an association threshold θ (we set θ to 5°). Otherwise, we consider $\mathbf{v}_D^d$ as a potential direction (potential direction set $\mathcal{V}_P$) for the global association. For instance, the orange and black directions in Fig. 4(d) denote the associated directions and the potential direction, respectively. If the detection step is not performed, we directly consider the tracked AF as the local AF. The procedure of the local association is formulated in Alg. 2.

Global Association. Through the tracking step, the correspondences between the local AF $\mathcal{V}_L^k$ and the global AF $\mathcal{V}_G$ (*i.e.*, local association $\mathbf{l}_L^k$) are determined. Thus, based on these relations, we can estimate the rotational motion of the camera in the referential of the global AF (this will be discussed in Sec. 4.2). In this referential, similarly to the

Algorithm 2 Local association

Input: Tracked AF $\mathcal{V}_T^k$=$\{\mathbf{v}_T^t\}$, detected AF $\mathcal{V}_D^k$=$\{\mathbf{v}_D^d\}$, and association threshold θ.
Output: Local AF $\mathcal{V}_L^k$ with its association label $\mathbf{l}_L^k$ and potential direction $\mathcal{V}_P$.

$\mathcal{V}_L^k \leftarrow \mathcal{V}_T^k$ ▷ Initialize local AF as tracked AF
$\mathbf{l}_L^k \leftarrow \mathbf{l}_L^{k-1}$ ▷ Initialize association label
Discard tracking failed direction in $\mathbf{l}_L^k$
$\mathcal{V}_P \leftarrow \emptyset$ ▷ Initialize $\mathcal{V}_P$
for each $\mathbf{v}_D^d$ **do**
  $\tilde{\mathbf{v}}^t$=$\arg\min_{\mathbf{v}_T^t} \angle(\mathbf{v}_D^d, \mathbf{v}_T^t)$ ▷ Find the closest tracked direction
  if $\angle(\mathbf{v}_D^d, \tilde{\mathbf{v}}^t) \leq \theta$ **then**
    Associate $\mathbf{v}_D^d$ into $\mathcal{V}_L^k$
  else
    $\mathcal{V}_P \leftarrow \{\mathcal{V}_P \cup \mathbf{v}_D^d\}$ ▷ Add $\mathbf{v}_D^d$ to $\mathcal{V}_P$
  end if
end for

Algorithm 3 Global association

Input: Local AF $\mathcal{V}_L^k$ with its association label $\mathbf{l}_L^k$, global AF $\mathcal{V}_G$with its activation label $\mathbf{l}_G$, potential directions $\mathcal{V}_P$=$\{\mathbf{v}_P^p\}$, rotation S^* between global Atlanta and current coordinates, and association threshold θ.
Output: local AF $\mathcal{V}_L^k$ with the updated association label $\mathbf{l}_L^k$ and global AF $\mathcal{V}_G$ with the updated activation label $\mathbf{l}_G$.

Deactivate non-tracked direction in $\mathcal{V}_G$ ▷ *Death* operation
$\bar{\mathcal{V}}_G = S^*\mathcal{V}_G$ ▷ Rotate $\mathcal{V}_G$ by to the current camera coordinate
for each $\mathbf{v}_P^p$ **do**
  $\tilde{\mathbf{v}}^g$=$\arg\min_{\bar{\mathbf{v}}_G^g} \angle(\mathbf{v}_P^p, \bar{\mathbf{v}}_G^g)$ ▷ Find the closest global direction
  if $\angle(\mathbf{v}_P^p, \tilde{\mathbf{v}}^g) \leq \theta$ **then** ▷ *Revival* operation
    Revive $\tilde{\mathbf{v}}^g$
    Activate the corresponding activation label in $\mathbf{l}_G$
    $\mathcal{V}_L^k \leftarrow \{\mathcal{V}_L^k \cup \mathbf{v}_P^p\}$ ▷ Add revived direction in $\mathcal{V}_L^k$
    Add the revived direction label in $\mathbf{l}_L^k$
  else if $\angle(\mathbf{v}_P^p, \tilde{\mathbf{v}}^g) > \theta$ **then** ▷ *Birth* operation
    if $\mathbf{v}_P^p$ lies on the horizon **then**
      Compute angle α to define the new direction
      Add the new direction by α in $\mathcal{V}_G$
      Add the new label in $\mathbf{l}_G$ ▷ Update $\mathbf{l}_G$
      $\mathcal{V}_L^k \leftarrow \{\mathcal{V}_L^k \cup \mathbf{v}_P^p\}$ ▷ Add the new direction in $\mathcal{V}_L^k$
      Add the new direction label in $\mathbf{l}_L^k$
    end if
  end if
end for

local association, we associate the local AF with the global AF using their angular distance.

The detailed process is as follows (see Alg. 3). We first deactivate non-associated directions in the global AF *w.r.t.* the local AF (*death* operation); that is, we update the activation label $\mathbf{l}_G$ by setting the label of the non-tracked direction as 0. We then transform the global AF $\mathcal{V}_G$ into the current camera coordinate using the estimated rotation S^*. We denote the transformed global AF as $\bar{\mathcal{V}}_G$. Similarly to the local association, we find the closest direction in $\bar{\mathcal{V}}_G$ for each potential direction $\mathbf{v}_P^p$. We then revive the associated direction in the global AF if it was not activated and their angle distance is less than the association threshold θ (*revival* operation). If a potential direction $\mathbf{v}_P^p$ is associated with none of the directions in $\bar{\mathcal{V}}_G$ and lies on the horizon of $\bar{\mathcal{V}}_G$, we birth a new Atlanta direction in the global AF and make it the valid one in the local AF (*birth* operation). To insert the new horizontal direction into an existing global AF, we project the new direction on the horizon and compute α by measuring the angle with the first horizontal direction (α_4 in Fig. 4(e) for example).

4.1.4 *Atlanta World Constraint*

After the association step, we initially refine each direction in the local AF $\mathcal{V}_L^k$ via mode seeking during the tracking step, after which we enforce the AW constraint. The AW constraint forces the local AF to satisfy the AW assumption (*i.e.*, $\mathbf{v}_v \perp \mathbf{v}_{h_m}$), and allows for complete exploitation of the structural regularities of the Atlanta world in the proposed SLAM framework. It should be noted that without the AW constraint module, we cannot represent the tracked or detected Atlanta directions (local AF) using the efficient AF parametrization by $\{\mathbf{R}, \{\alpha_m\}_{m=2}^M\}$. In addition, tracking-by-detection without the AW constraint will induce a drift for each direction in the local AF and result in inaccurate rotation estimation.

To this end, we first determine an accurate representative vertical direction,[3] which defines the horizon. We project the dominant horizontal direction, which has the maximum density (having maximum inliers), onto the horizon and then sequentially refine the other horizontal directions while maintaining each relative alpha angle *w.r.t.* the dominant horizontal direction. We repeat this procedure until the local AF converges.

4.2 Robust Rotation Estimation

Using our association strategy, we have the updated global AF $\mathcal{V}_G$=$\{\mathbf{v}_G^g\}$ and the local AF $\mathcal{V}_L^k$=$\{\mathbf{v}_L^l\}$ with their association label $\mathbf{l}_L^k$ at the current frame. With a minimum of two matched Atlanta directions, the camera rotation can be estimated in the global AF referential as in Sec. 3.2.2. If more directions are available, a set of rotation matrices can be computed by sampling all possible combinations of triplets of directions, as:

$$\mathbf{R}_i = \mathbf{V}_L^i \, \mathbf{V}_G^{i^{-1}}, \tag{3}$$

where $\mathbf{V}_L^i \in \mathbb{R}^{3\times3}$ and $\mathbf{V}_G^i \in \mathbb{R}^{3\times3}$ represent the i-th sampled local AF and the corresponding sampled global AF, respectively. More concretely, given $\mathcal{V}_G$ and $\mathcal{V}_L^k$ with known association, we sample all possible combinations of triplets of directions. We then concatenate these sampled directions to form a 3×3 matrix. For instance, we can construct $\mathbf{V}_L^1 = [\mathbf{v}_L^1 \, \mathbf{v}_L^2 \, \mathbf{v}_L^3]$ and $\mathbf{V}_G^1 = [\mathbf{v}_G^1 \, \mathbf{v}_G^3 \, \mathbf{v}_G^4]$, where each global-local pair ($\mathbf{v}_L^1 \leftrightarrow \mathbf{v}_G^1$, $\mathbf{v}_L^2 \leftrightarrow \mathbf{v}_G^3$, and $\mathbf{v}_L^3 \leftrightarrow \mathbf{v}_G^4$) is associated. Note that we always include the vertical direction by default in the sampling and orthogonalize $\mathbf{V}_L^i$ and $\mathbf{V}_G^i$ to prevent a degenerate solution in Eq. (3).[4]

We generate a set of candidate rotations using Eq. (3) and estimate the camera rotation by means of a single rotation averaging [47, 48]. We use L^p-mean rotation *w.r.t.* $d(\cdot,\cdot)$:

$$S^* = \arg\min_{S\in SO(3)} \sum_{i=1} d(\mathbf{R}_i, S)^p, \tag{4}$$

where $p = 1$ in our case to increase the degree of robustness against outliers. This facilitates an accurate estimation of the camera rotation given the candidate rotations.

3. To ensure an accurate vertical direction well supported by the horizontal directions, we generate a set of virtual vertical direction hypotheses from the cross products of combinations of horizontal direction pairs. We then compute a representative vertical direction using the weighted sum of the initial vertical direction and virtual direction set.

4. If the vertical direction does not exist in the tracked AF, we temporally generate the virtual vertical direction by computing the cross product of any two horizontal directions.

5 Linear RGB-D SLAM Formulation

Based on the understanding of the structural regularities (*cf*., Sec. 4), we present a novel linear SLAM framework for structured environments. For this purpose, we introduce a method of detecting structure-aware planes in structured environments in Sec. 5.1, which plays an important role in the proposed linear SLAM approach. We then introduce a novel SLAM approach using planar features within a linear KF framework in Sec. 5.2. For a smooth transition and better understanding, we first describe a linear SLAM formulation for the Manhattan world (L_{MW}-SLAM) and then extend it to the Atlanta world (L_{AW}-SLAM). The overview of the proposed L-SLAM is illustrated in Fig. 2.

5.1 Structure-aware Plane Detection

Once the structural regularities of the scene *w.r.t*. the camera pose (*i.e*., local AF) has been established, we can easily identify the dominant planes supporting the current structured environments. Given the surface normals for each pixel, we find the relevant normal vectors inside a conic section of each Atlanta direction. We perform the plane RANSAC [50] with the pixels corresponding to the surface normals near each direction of the local AF. We model the plane [51] as follows:

$$n_x u + n_y v + n_z = w \quad (u = \frac{X}{Z}, v = \frac{Y}{Z}, w = \frac{1}{Z}), \tag{5}$$

where X, Y, and Z denote the 3D coordinates, u, v, and w correspond to the normalized image coordinates and the measured disparity at that coordinate, and n_x, n_y, and n_z are the model parameters representing the distance and orientation of the plane. The error function of the plane RANSAC is the distance between the 3D point and the plane. We fit the plane to the given inlier 3D points from the plane RANSAC in a least-square fashion.

If the angle difference between the normal vector of the plane and one of the Atlanta directions is less than 5°, we refit this plane to a set of measured disparity values (w) subject to the constraint that it must be parallel to the corresponding Atlanta direction. We compute the optimal scale factor in a least-square manner that minimizes:

$$s^* = \arg\min_s \left\| s\left(r_x u + r_y v + r_z\right) - w \right\|, \tag{6}$$

where s is the scale factor representing the reciprocal of the distance from the plane to the center of the camera, and $\mathbf{r}_i = [r_x,\ r_y,\ r_z]^\top \in \mathbb{R}^{3\times 1}$ denotes the x, y, and z component of the unit vector of the corresponding Atlanta direction $\mathbf{r}_i$. In this manner, we can find the structure-aware planar features in the scene whose normals are aligned with the local AF, as shown in Fig. 5.

It is worth noticing that this 1-D distance (scale factor) from the camera position to each plane along the Atlanta direction plays an important role as the measurement in the proposed L-SLAM (*cf*., Sec. 5.2.3), linearizing the complex SLAM formulation.

5.2 Linear RGB-D SLAM for Structured Environments

5.2.1 KF State Vector Definition

The state vector in the KF consists of the current 3-DoF translational motion of the camera and 1-D representation of the structure-aware planar features (*i.e*., planar landmarks) in the global planar map. We denote the state vector by $\mathbf{x}$ with its associated covariance $\mathbf{P}$:

$$\begin{aligned} \mathbf{x} &= \left[\mathbf{p}^\top,\ m_1,\ \cdots,\ m_n\right]^\top \in \mathbb{R}^{(3+n)\times 1} \text{ and} \\ \mathbf{P} &= \begin{bmatrix} \mathbf{P_{pp}} & \mathbf{P_{pm}} \\ \mathbf{P_{mp}} & \mathbf{P_{mm}} \end{bmatrix} \in \mathbb{R}^{(3+n)\times(3+n)}, \end{aligned} \tag{7}$$

where $\mathbf{p} = [x,\ y,\ z]^\top \in \mathbb{R}^{3\times 1}$ denotes the 3-DoF camera translation in the global map frame where the rotation of the camera is completely compensated for. In contrast to previous planar SLAM approaches, we do not include the camera orientation in the state vector, which is the main factor that increases the nonlinearity in the SLAM problem [17] because we already have accurate and drift-free camera rotation in Sec. 4.2. The map $m_j \in \mathbb{R}^1$ represents the 1-D distance (offset) between the structure-aware planar feature and the origin in the global map frame,[5] and n is the number of structure-aware planes in the global map. Although each structure-aware planar feature in Sec. 5.1 consists of the 1-D distance and the alignment for the AF, we only track and update the distance since the alignment of the structure-aware planes does not change over time. A newly detected structure-aware planar feature is additionally augmented after the last map component of the state vector. It should be noted that there are no variables related to the camera or plane orientation in the state vector $\mathbf{x}$, resulting in a linear KF formulation.

One of the problems of using the KF in SLAM is the quadratic update complexity in the number of features that can limit the ability to use multiple measurements [52]. Because we model only large and dominant planar structures, such as a wall or floor with a single variable per plane, the size of the state vector $\mathbf{x}$ is very small compared to the size obtained via other EKF-SLAM approaches, as shown in Table 1. While other EKF-SLAM methods [23]–[26] represent the plane using a 3 to 10-D vector, the proposed method models the planar landmark with only one parameter (offset), resulting in a very low complexity. If the number of the planar features (n) is 10, the state size of the proposed method is approximately ten times smaller than that of Martinez's EKF-SLAM method [26], meaning the EKF update is expected to be ∼100 times faster.

5.2.2 Process Model

We predict the next state based on the 3-DoF translational movement between the consecutive frames (*cf*., Sec. 3.2.2). We propagate the 3-DoF camera translation, and assume the map does not change. Then, our process model can be written as follows:

$$\mathbf{x}_k = \mathbf{F}\mathbf{x}_{k-1} + \begin{bmatrix}\triangle\mathbf{p}_{k,k-1}^\top & \mathbf{0}_{1\times n}\end{bmatrix}^\top, \tag{8}$$

where $\mathbf{F}$ denotes the identity matrix, and $\triangle\mathbf{p}_{k,k-1}$ is the estimated 3-DoF translational movement between the k and $k-1$ image frame. The covariance matrix consists of the process noise of the LPVO approach, which indicates the error of the estimated 3-DoF translational movement from

5. It should be noted that 1-D structure-aware distance m_j differs from the structure-aware distance in Sec. 5.1 in that the reference of m_j is the origin of the global map coordinate, *i.e*., not the current center of the camera.

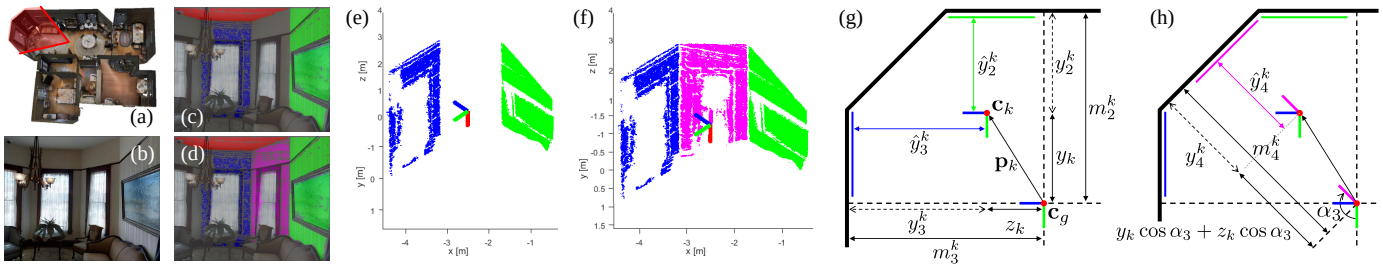

Fig. 5. **Example of the Kalman filter components for L-SLAM**. (a) Example of 3D indoor scene [49] with a specific view point (red color), and (b) the corresponding RGB image. Results of plane detections supporting (c) the Manhattan frame and (d) the Atlanta frame are overlaid on top of the RGB images, respectively, and (e,f) the corresponding color-coded planar features are drawn in a 3-D space, where we exclude a set of planes describing the vertical direction for better visualization purpose. The detailed descriptions of each variable, the definition of the state vector, and the measurement model for (g) the Manhattan world and (h) the Atlanta world in Kalman filter.

TABLE 1
Advantages of L-SLAM over existing EKF-SLAM methods.

	L-SLAM (ours)	[23]	[24]	[25]	[26]
State size	$3+n$	$7+7n$	$7+9n$	$15+3n$	$12+10n$
Linearity	Linear	Nonlinear	Nonlinear	Nonlinear	Nonlinear

the LPVO. Currently, the process noise is manually tuned to 1 cm empirically.

5.2.3 Measurement Model

We update the state vector in the KF by observing the distance between the currently detected structure-aware planar features and the current camera pose. For better understanding, we first describe a measurement model for the Manhattan world; thereafter, we present a general measurement model for the Atlanta world while maintaining the linear property. For the sake of presentation, we assume that the planar features supporting the vertical direction $\mathbf{v}_v$ $(=\mathbf{r}_1)$ and the first horizontal direction $\mathbf{v}_{h_1}$ $(=\mathbf{r}_2)$ of the global AF correspond to the x-axis and y-axis of the world coordinate system, respectively. This regime is directly applicable to the Manhattan world, as shown in Fig. 5.

Measurement Model for Manhattan World. In the Manhattan case, the simplest structure, each Manhattan direction directly corresponds to the x-axis, y-axis, and z-axis of the world coordinate according to the above assumption. This allows us to project the state vector into the observed space by a simple arithmetic operation – subtraction. In other words, a measurement model $\mathbf{y}$ for the m_j is defined by:

$$\mathbf{y} = \mathbf{H}\mathbf{x} = \begin{bmatrix} m_1 - x \\ m_2 - y \\ m_3 - z \\ \vdots \end{bmatrix} \in \mathbb{R}^{q\times 1}, \tag{9}$$

$$\mathbf{H} = \begin{bmatrix} -1 & 0 & 0 & 1 & 0 & 0 & \cdots \\ 0 & -1 & 0 & 0 & 1 & 0 & \cdots \\ 0 & 0 & -1 & 0 & 0 & 1 & \cdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots \end{bmatrix} \in \mathbb{R}^{q\times(3+n)}, \tag{10}$$

where $\mathbf{H}$ is the observation model that maps the state space into the observed space, and q is the number of matched structure-aware planar features; here, the planar features are limited to Manhattan-support one. For example, the second row of $\mathbf{H}$ indicates that the matched plane supports the y-axis. This enables us to observe an orthogonal distance by subtracting m_2 from the current position (see y_2^k in Fig. 5(g)).

Measurement Model for Atlanta World. In contrast to the Manhattan case, additional horizontal directions $\mathbf{v}_{h_m}$ $(m \geq 2)$ in the AF cannot be directly represented as a simple coordinate (*e.g.*, x-axis); that is, a 1-D distance of the structure-aware plane related to the additional horizontal direction cannot be represented using the axis of the single coordinate. This may cause non-linearity in the proposed KF framework.

To resolve this problem, we fully exploit the efficient parametrization of the AF (*cf.*, Sec. 3.2.1). Specifically, as a key contribution, we represent planar features supporting additional horizontal directions by the 1-D angle α and the corresponding 1-D distance, where the y-axis is rotated by α around the x-axis, for each horizontal direction. This makes the horizontal direction representation linear without additional non-linear variables in the state variable and enables a linear formulation for the Atlanta case. Thus, we can seamlessly extend the measurement model for the Manhattan case in Eqs. (9) and (10) into the general structured environments – the Atlanta world. We formally define the observation model $\mathbf{H}$ and measurement model $\mathbf{y}$ as:

$$\mathbf{y} = \mathbf{H}\mathbf{x} = \begin{bmatrix} m_1 - x \\ m_2 - y \\ m_3 - z \\ m_4 - y\cos\alpha_3 - z\sin\alpha_3 \\ \vdots \end{bmatrix} \in \mathbb{R}^{q\times 1}, \tag{11}$$

$$\mathbf{H} = \begin{bmatrix} -1 & 0 & 0 & 1 & 0 & 0 & 0 & \cdots \\ 0 & -1 & 0 & 0 & 1 & 0 & 0 & \cdots \\ 0 & 0 & -1 & 0 & 0 & 1 & 0 & \cdots \\ 0 & -\cos\alpha_3 & -\sin\alpha_3 & 0 & 0 & 0 & 1 & \cdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots \end{bmatrix} \in \mathbb{R}^{q\times(3+n)}, \tag{12}$$

where α_m is an angle defining the m-th horizontal direc-

tion ($m \geq 2$). The entities of $\mathbf{y}$ are the observed 1-D distances from the structure-aware planar features computed using the current state vector. In particular, we can linearly observe the 1-D distance even for an additional horizontal direction using α_m. For example, we can compute (observe) an orthogonal distance by a simple geometric relation using the current position $\mathbf{p}_k$ and α_3, as described in Fig. 5(h) and Eq. (12) – the fourth row of $\mathbf{H}$ in Eq. (12).

From the measurement model $\mathbf{y}$ in Eq. (11), we compute the residual between $\mathbf{y}$ and the measurements $\hat{\mathbf{y}}$ (1-D distance from the current camera coordinate). We then update the state vector of the KF for all associated structure-aware planes with the planar landmarks in the global planar map. Because all formulas and calculations are *perfectly* linear from Eqs. (7) to (12), there is no local linearization error, and we can easily calculate the optimal Kalman gain [53]. In this manner, we can consistently track the 3-DoF camera translation and 1-D planar map position efficiently and reliably.

Our SLAM algorithm relies on the drift-free rotation estimates in Sec. 4.2, which shows accurate and stable rotation tracking performance (about 0.2° error in average) in structured environments. We treat this small orientation error as the measurement noise by the Kalman filter, which removes the need to explicitly consider the correlations [54]. The measurement noise includes not only the orientation error but also the distance measurement noise of the RGB-D camera. Currently, the measurement error is manually tuned to 2 cm.

5.2.4 Planar Map Management

At the beginning of L-SLAM, we initialize a state vector and its covariance with the structure-aware planar features detected at the first frame. When constructing a global planar map, we only utilize the structure-aware planes that have a sufficiently large area in order to accurately recognize the dominant structural characteristics such as walls, floor, and ceiling in the current structured environments. We perform plane matching using the distance (offset) and alignment from the currently detected structure-aware planar features and the global plane map in the state vector. If the metric distance between the two planes is less than a certain length (in our experiments, 10 cm), and they have the same alignment, the detected planar feature is associated with an existing global planar map to update the state vector. The global planar map can be extended incrementally as new structure-aware planes are detected.

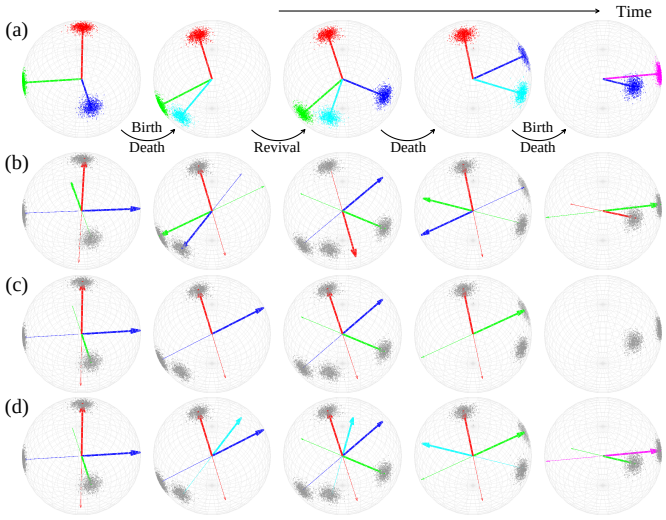

Fig. 6. **Evaluation of the proposed tracking-by-detection on the synthetic sequence**. (a) Example of generated ground truth data with possible scenarios such as birth, death, and revival. The corresponding results: (b) BnB-based AF detection method [42], (c) MF tracking method (Manhattan only), and (d) the proposed method. In contrast to the other comparison methods, our approach maintains a consistent association between consecutive frames.

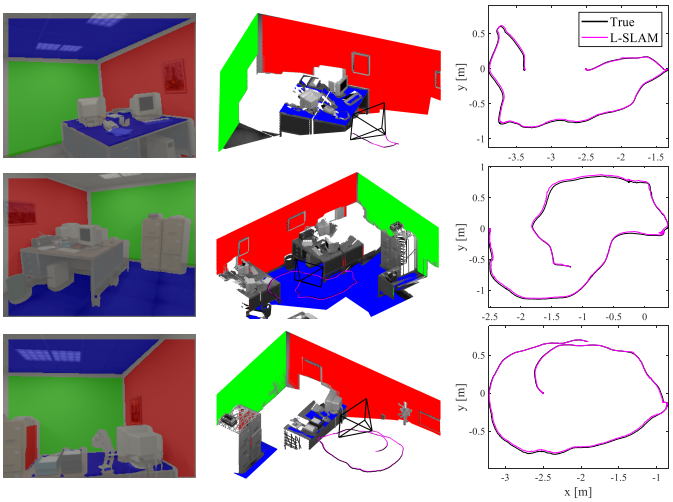

Fig. 7. **Selected motion estimation results of the proposed algorithm in the ICL-NUIM dataset**. The first and second columns show the structure-aware planar features for mapping and localizing the camera position in the proposed L-SLAM algorithm. Vertical surfaces are red or green and horizontal surfaces are blue depending on their orientation. The magenta and black lines in the third column represent the estimated and the ground-truth trajectories, respectively.

6 Evaluation

We first validate the understanding of structural regularities for the Atlanta world on synthetic sequences. We qualitatively compare the proposed tracking-by-detection scheme and quantitatively evaluate the robust rotation estimation using different noise cases. We then evaluate the proposed L-SLAM on various RGB-D datasets from room-size (∼10 m) to building-size (∼100 m) structured environments:

- **ICL-NUIM** [10] is a room-size RGB-D dataset providing RGB and depth images rendered in a synthetic living room and office with ground-truth camera trajectories. It is challenging to accurately estimate the camera pose because of the low-texture and artificial noise in the depth images.
- **TUM RGB-D** [55] is the de facto standard RGB-D dataset for VO/visual SLAM evaluation composed of ground-truth camera poses and RGB-D images captured in room-scale environments.
- **TAMU RGB-D** [56] consists of large-scale man-made environments such as stairs and corridors inside a building. In particular, it includes a planar structured scene following the Atlanta world.
- **Author-collected RGB-D dataset** contains RGB and depth images at 30 Hz in large building-scale planar environments with an Asus Xtion RGB-D camera. We start

TABLE 2
Computational time analysis.

Module	Runtime
Plane detection	1 ms
Plane fitting	0.1 ms
Atlanta tracking	16 ms
Atlanta detection	180 ms

and end at the same position to evaluate loop closures and for the sake of consistency because ground-truth trajectories and maps are not available.

We compare our L-SLAM to other state-of-the-art RGB-D SLAM and planar SLAM approaches, namely ORB-SLAM2 [6], DVO-SLAM [5], CPA-SLAM [33], KDP-SLAM [9], DPP-SLAM [8], InfiniTAM [57], BundleFusion [58], ElasticFusion [59], BAD SLAM [60], and Structure SLAM [61]. In contrast to the proposed L-SLAM, which is based on a linear formulation, they all perform a high-dimensional nonlinear pose graph optimization, and some of them require GPUs for extensive computation and network inference. We also show an improvement compared to LPVO [18], which our new SLAM approach builds on. We deactivate the capability to detect loop closures via image retrieval in ORB-SLAM2 for a fair comparison. We test each SLAM method with the original source code provided by the authors while including the result of CPA-SLAM and KDP-SLAM taken directly from [9] and [61].

6.1 Implementation Details

We implement the proposed L-SLAM method in an unoptimized MATLAB code for fast prototyping. We run both L-SLAM throughout the sequence on a desktop computer with an Intel Core i7-4790K 4.0GHz CPU and 32GB of RAM.

Computational Time. To validate the computational trade-off of the proposed SLAM approach, we analyze the computational time of key components on the ICL NUIM dataset (see Table 2); 1) plane detection and fitting and 2) tracking-by-detection.

Plane detection and fitting are essential modules in conventional plane-based RGB-D SLAM frameworks [8, 27, 32]. In our L-SLAM framework, plane detection and fitting take $\sim$1 ms and $\sim$0.1 ms, respectively, which is reasonable. Moreover, considering that we utilize a structural assumption – the MW or the AW assumptions, we can efficiently estimate planes supporting the structural assumption.

Regarding tracking-by-detection scheme, the tracking takes $\sim$16 ms. We applied manifold-constrained mean shift algorithm [44] on the limited search space by known dominant directions of the Atlanta world, which efficiently tracks each dominant direction. On the contrary, the detection takes $\sim$180 ms, which guarantees global optimality but might be a computational bottleneck. However, we perform the detection when it satisfies specific conditions (*cf*., Sec. 4.1.2). Thus, it does not hinder the computational load of the proposed SLAM (this is discussed in detail in Sec. 7.2). Owing to the above reasons, the proposed L-SLAM works in $10 \sim 15$ Hz (near real-time)[6], even though it is implemented in MATLAB.

6. Our L-SLAM operates at above 20 Hz in L_{MW}-SLAM , and $10 \sim 15$ Hz in case of L_{AW}-SLAM due to the AF detection module.

TABLE 3
Robustness of rotation estimation according to variant noises. We generate a set of synthetic sequences and compute the mean angular error between the estimate and ground truth.

Noise type	Angular error
S	0.526°
L + W	0.502°
S + W + O	0.515°
S + W + D	0.534°
S + W + O + D	0.506°

6.2 Validation of Structural Regularities

Synthetic Data Generation. To evaluate the understanding of structural regularities under the AW assumption, we generate synthetic data including possible scenarios in the association such as death, birth, and revival, as shown in Fig. 6(a). Specifically, we define the global AF with $M(=4)$ horizontal directions in the world coordinate (five Atlanta directions in total) and generate the surface normal distributions supporting the global AF, where each surface normal distribution has a small variance. To mimic natural motions, we generate sequential 3D rotations (600 frames in total) and smoothen the results continuously using a rotation-smoothing method [62]. We also set various activation labels to evaluate the association. Given the rotations and activation labels, we rotate the global AF including the surface normal distributions to generate the ground truth of the local AF. We call this synthetic sequence *base sequence*.

In addition, we generate different types of noises that may happen in the real world, such as white noise, different distribution variance, drift, and outlier (*i.e*., normal distribution not following the Atlanta world). We name each noise type using the first alphabet; for instance, 'S' for small variance, 'L' for large variance, 'W' for white noise, 'O' for outlier, and 'D' for drift. According to the difficulty, we combine these noises to the base sequence and generate a set of synthetic sequences, called *noise sequences* (the first column in Table 3).

Evaluation. We first qualitatively evaluate the proposed tracking-by-detection method on the base sequence. We compare the proposed approach with the BnB-based AF detection [42] and MF tracking [18]. As shown in Fig. 6(b), the AF detection method independently estimates AF while maximizing the number of inliers, thus exhibiting fluctuations and an absence of consistent associations between consecutive frames. In contrast to the detection approach, MF tracking relatively shows stable tracking along the sequence, even though it fails to track when it encounters a non-Manhattan frame, as shown in Fig. 6(c). Our tracking-by-detection method, however, shows a stable tracking result while identifying new or missed Atlanta directions, which demonstrates its effectiveness in structured environments, as shown in Fig. 6(d).

In addition, we quantitatively validate the robustness of our system against the noise sequences in rotation estimation. We perform the tracking-by-detection on each noise sequences and then estimate the rotation using the proposed robust rotation estimation. As shown in Table 3, our approach shows stable and accurate rotation estimation results

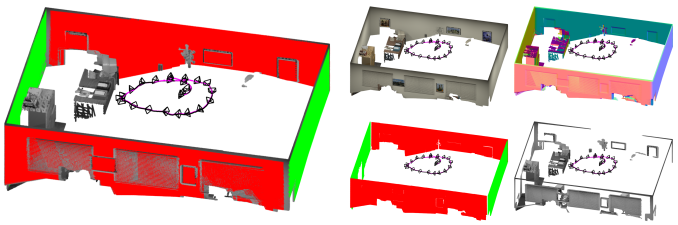

Fig. 8. **Qualitative result of office room sequences from the ICL-NUIM dataset**. Left: Synthetic scene 3D reconstruction of an office room from the ICL-NUIM dataset, displaying both planar and non-planar regions with the estimated (magenta) and the ground-truth (black) trajectories. Right: Color output, surface normal map, non-planar regions only with gray scale, and orthogonal planar regions only with RGB scale in clock-wise order. The ceilings are not shown for visibility.

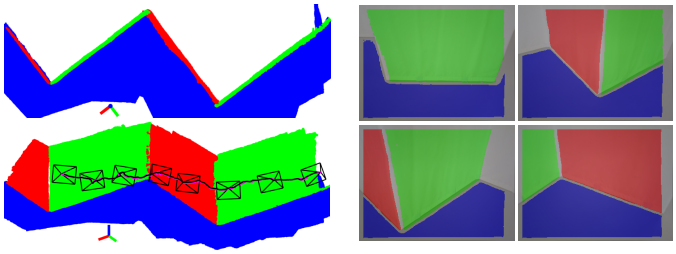

Fig. 9. **Qualitative result on *fr3/str_notex_near* of the TUM RGB-D dataset**. Top and side views of the global 3D planar map generated by the proposed L-SLAM algorithm from *fr3/str_notex_near* (left). The structure-aware planar features are overlaid on top of the original images of the respective scenes in clockwise order (right).

regardless of noise types, which means that our tracking-by-detection scheme and robust rotation estimation is robust to various noises in the real-world. For a detailed qualitative comparison, readers can refer to the supplementary video.

6.3 ICL-NUIM Dataset

We report the root mean square error (RMSE) of the absolute trajectory error (ATE) [55] for the resulting camera trajectories of all living room and office sequences with noise in Table 4. We highlight the smallest error for each sequence, and $\times$ means the estimation failure for the corresponding sequence. The results of CPA-SLAM, KDP-SLAM, BundleFusion, and ElasticFusion for the office are not available, marked as $-$ in Table 4. Note that we directly quote the results of CPA-SLAM, KDP-SLAM, BundleFusion, and ElasticFusion for a fair comparison.

CPA-SLAM, InfiniTAM, and BAD SLAM show the best quantitative results in some living room and office sequences, but they all require GPU hardware for extensive computation. In addition, InfiniTAM and BAD SLAM easily fail to estimate the camera motion and diverge quickly when looking at only one or two planes in the RGB and depth images denoted as $\times$ in Table 4. In contrast, our L-SLAM can continue estimating the camera motion stably by exploiting lines and planes together and presents estimation results comparable to other state-of-the-art methods without the help of GPU computation. We plot the estimated camera trajectories using L-SLAM (see Fig. 7), showing that L-SLAM is comparable to other state-of-the-art SLAM approaches without a nonlinear pose graph optimization. It is noteworthy that the RMSE of L-SLAM is sometimes larger than the RMSE of LPVO because of the inaccuracy of planar distance measurements caused by a simulated sensor noise in depth images in the ICL-NUIM dataset; however, that difference is very minor (about 2 $\sim$ 5 cm), and overall, the proposed L-SLAM approach has a tendency to show more accurate 6-DoF estimation results in most test cases.

In the office sequences, L-SLAM achieves more accurate or similar performance to other SLAM methods because the office environments consist of sufficient planar features. Reconstruction results of the office room sequences are shown in Fig. 8. Although InfiniTAM performs the best owing to sufficient texture and planar features in *of-kt3n*, the proposed L-SLAM also performs nearly as well. Among the CPU-only RGB-D and planar SLAM methods (except for CPA-SLAM, InfiniTAM, BAD SLAM, and Structure SLAM, which require a GPU), L-SLAM presents the lowest average trajectory error. The resulting camera trajectories with L-SLAM are shown in Fig. 7, demonstrating that L-SLAM, with an efficient and linear KF, is comparable to other recent SLAM approaches especially for highly-planar environments.

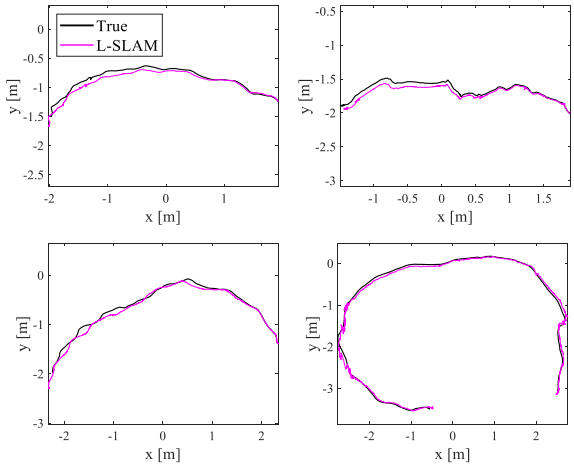

Fig. 10. **The estimated camera trajectories on the TUM RGB-D dataset**. The estimated camera trajectories with L-SLAM (magenta) and ground-truth (black) for the TUM RGB-D dataset in clockwise order: fr3/str_notex_far, fr3/str_notex_near, fr3/large_cabinet, and fr3/str_tex_far.

It is noteworthy that both L_{MW}-SLAM and L_{AW}-SLAM were demonstrated in this experiment, but considering that we do not know which structural assumption a given scene follows, L_{AW}-SLAM is more appropriate in real-world indoor applications. In this regard, the performance gap between L_{AW}-SLAM and other existing methods may become smaller. However, our approach still shows comparable results and has its own advantages, such as linear formulation and generation of planar maps supporting the Atlanta world.

6.4 TUM RGB-D Dataset

We choose several RGB-D sequences in the environments where the planar features are sufficiently present in the TUM RGB-D dataset [55]. Table 5 compares estimation results of the SLAM approaches. ORB-SLAM2, BundleFusion, and BAD SLAM show good quantitative results in texture-rich scenes such as *fr3/str_tex_far*, which is entirely expected as L-SLAM utilizes a much cheaper approach. While L-SLAM shows a comparable performance even in *fr3/str_notex_near* poorly-featured environments, as shown

TABLE 4
Evaluation Results of ATE RMSE (unit: m) on ICL-NUIM Benchmark. ATE RMSEs are measured. Lower is better (unit: meter).

Sequence	lr-kt0n	lr-kt1n	lr-kt2n	lr-kt3n	of-kt0n	of-kt1n	of-kt2n	of-kt3n
ORB-SLAM2	0.010	0.185	0.028	0.014	0.049	0.079	0.025	0.065
DVO-SLAM	0.108	0.059	0.375	0.433	0.244	0.178	0.099	0.079
CPA-SLAM	**0.007**	0.006	0.089	**0.009**	–	–	–	–
KDP-SLAM	0.009	0.019	0.029	0.153	–	–	–	–
InfiniTAM	×	0.006	**0.013**	×	0.042	0.025	×	**0.010**
BundleFusion	0.009	0.012	0.013	0.013	–	–	–	–
ElasticFusion	0.009	0.009	0.014	0.106	–	–	–	–
BAD SLAM	×	**0.005**	0.014	×	×	**0.013**	**0.019**	0.013
Structure SLAM	–	0.016	0.045	0.046	–	×	0.031	0.065
LPVO	0.015	0.039	0.034	0.102	0.061	0.052	0.039	0.030
L_{MW}-SLAM	0.012	0.027	0.053	0.143	**0.020**	0.015	0.026	0.011
L_{AW}-SLAM	0.014	0.035	0.027	0.117	0.036	0.022	0.027	0.025

TABLE 5
Evaluation Results of ATE RMSE (unit: m) on TUM RGB-D Benchmark.

Sequence	fr3/str_notex_far	fr3/str_notex_near	fr3/str_tex_far	fr3/str_tex_near	fr3/cabinet	fr3/large_cabinet
ORB-SLAM2	0.276	0.652	0.024	0.019	×	0.179
DVO-SLAM	0.213	0.076	0.048	0.031	0.690	0.979
InfiniTAM	**0.037**	**0.022**	×	×	**0.035**	0.512
BundleFusion	×	×	0.041	0.068	×	×
BAD SLAM	0.189	0.034	0.044	0.034	0.059	0.206
Structure SLAM	0.281	0.065	**0.014**	**0.014**	×	×
LPVO	0.075	0.080	0.174	0.115	0.520	0.279
L-SLAM (ours)	0.141	0.066	0.212	0.156	0.291	**0.140**

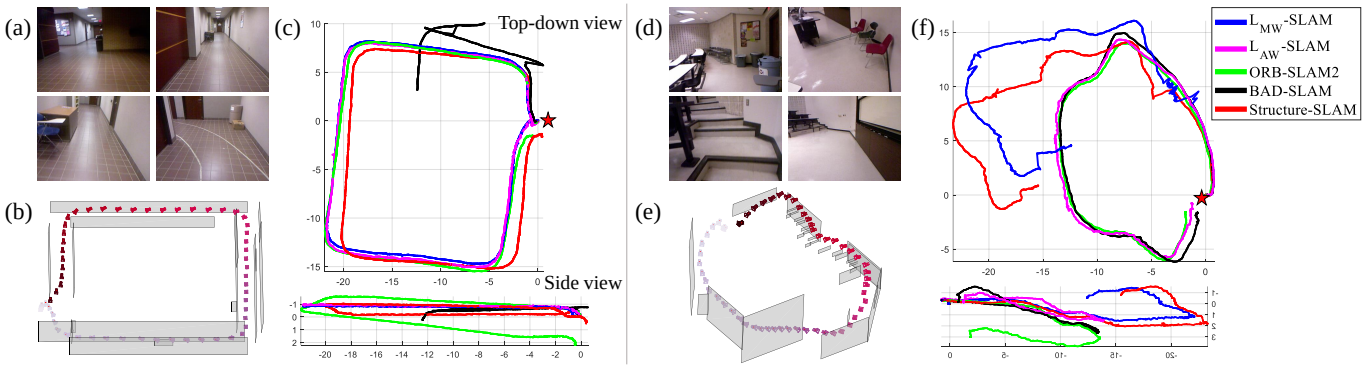

Fig. 11. **Evaluation on two representative sequences (*Corridor-A-const* and *Auditorium-const*) in the TAMU RGB-D dataset** [56]. (a,d) Sampled RGB images. (b,e) Estimated Atlanta planar map with sampled camera trajectories by L_{AW}-SLAM. (c,f) Estimated trajectories compared to L_{MW}-SLAM and ORB-SLAM2 [6]. It shows that our L_{AW}-SLAM provides a more stable and accurate estimation of the trajectories.

in Fig. 9, the accuracy of ORB-SLAM2 and BundleFusion show a significant drop. In *fr3/cabinet*, ORB-SLAM2 and Structure SLAM fail to estimate the camera trajectory (marked as × in Table 5). BundleFusion shows good quantitative results in some TUM sequences obtained in an environment with sufficient textures. However, it easily fails to estimate the 6-DoF camera pose on some TUM sequences (like *fr3/str_notex_far*) in structured environments with insufficient texture. BAD SLAM and Structure SLAM also show good quantitative results in some TUM benchmark, but they all require GPUs for extensive computation and network inference.

Although inaccurate planar distance measurements in L-SLAM sometimes cause slight performance degradation of LPVO, L-SLAM is generally more accurate compared with LPVO on average. Figure 10 presents the estimated trajectories using L-SLAM from *fr3/large_cabinet*, showing that the proposed L-SLAM consistently presents comparable results regardless of the existence of sufficient texture without the help of GPU computation.

6.5 TAMU RGB-D Dataset

We validate our L-SLAM (both L_{MW}-SLAM and L_{AW}-SLAM) with BAD SLAM and Structure SLAM on *Corridor-A-const* and *Auditorium-const* sequences that contain texture-less walls and stairs in planar environments. In particular, we demonstrate that our extension to the Atlanta world, L_{AW}-SLAM, seamlessly works well under the Atlanta world as

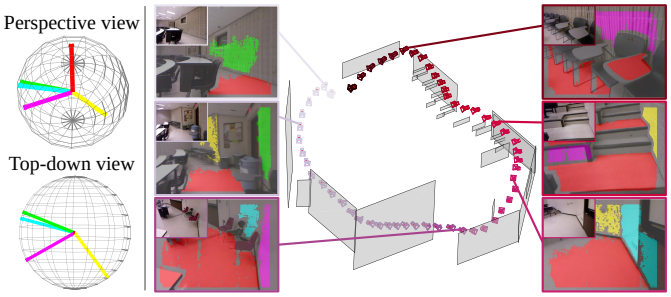

Fig. 12. **Qualitative result of L_{AW}-SLAM on the *Auditorium-const* sequence in the TAMU RGB-D dataset** [56]. *Left*: The estimated Atlanta structure (five Atlanta directions in total) by the proposed tracking-by-detection scheme, where the red arrow denotes the vertical direction and the others indicate the observed horizontal directions. *Right*: The estimated camera trajectory and global planar map, where we visualize the planar features supporting the Atlanta structure of the scene, and these planes are also color-overlaid on the sampled images. For visualization purposes, we only display a sampled camera trajectory, omit the planar maps supporting the vertical direction, and approximately obtain the plane boundaries as a pseudo representation.

well as the Manhattan world. We only undertake a qualitative evaluation as the TAMU RGB-D dataset does not provide the ground truth.

For *Corridor-A-const* (Fig. 11–*left*), in a challenging environment such as the low-texture walls and insufficient plane features, both L_{MW}-SLAM and L_{AW}-SLAM show stable and accurate estimation results because the sequence satisfies the MW assumption. In contrast, ORB-SLAM2 shows drifts owing to the lack of texture on the walls, which yields a biased keypoint distribution. Because the BAD SLAM requires three orthogonal planes to fully constrain the 6-DoF camera motion, it easily fails to estimate the camera motion and diverges quickly when looking at only one or two planes in the RGB and depth images, resulting in overall motion estimation failure, as shown on the left-hand side of Fig. 11. The proposed L-SLAM, however, can continue estimating the 6-DoF camera motion stably even when looking at only a single plane by exploiting points, lines, and planes altogether, and presents comparable estimation results compared to the other state-of-the-art methods without the help of GPU computation.

For *Auditorium-const* (Fig. 11–*right*), L_{MW}-SLAM and Structure-SLAM lost the track and estimation when it encountered a non-Manhattan part of the scene as can be inferred from the non-orthogonal walls in Fig. 12. Furthermore, it could not recover, even after a single failure because of the absence of a detection mechanism. ORB-SLAM2 shows a reasonable trajectory in this sequence but accumulates significant drift ($\sim$3 m) owing to the textureless stairs and walls (see the side view of the green-colored trajectory in Fig. 11(f)). By virtue of the proposed tracking-by-detection strategy, L_{AW}-SLAM continues to track the AF while dealing with new Atlanta directions of the scene, leading to a robust estimation of the camera poses. Our L_{AW}-SLAM is comparable to the BAD SLAM approach without a nonlinear pose graph optimization and complex GPU computation. In addition, the estimated global planar map in Fig. 12 shows that L_{AW}-SLAM reconstructs the 3D scene structures properly.

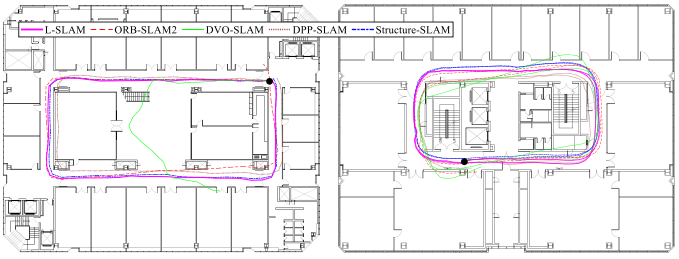

Fig. 13. **Comparison on the author-collected RGB-D dataset**. Estimated trajectories with the proposed and other RGB-D SLAM approaches on the author-collected dataset in a single-loop (left) and multiple-loop (right) sequences. We start and end at the same position marked in the black circle to check loop closure and the consistency in the resulting trajectories. With L-SLAM, the starting and ending points nearly match; for the others, they do not. Our L-SLAM stably and accurately tracks the 6-DoF camera motion, preserving the orthogonality of the estimated corridor trajectory in the square building.

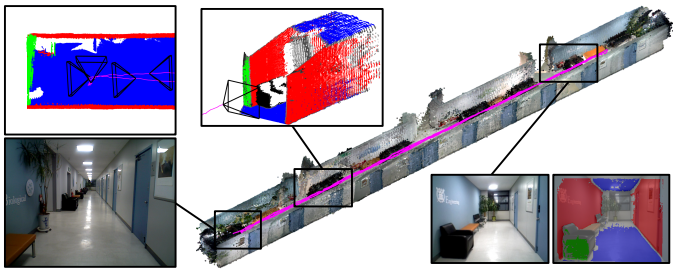

Fig. 14. **Qualitative result on the author-collected RGB-D dataset**. Accumulated 3D point cloud with the estimated trajectory (magenta) on the author-collected RGB-D dataset in a long corridor sequence. The 3D geometry of the long corridor with the doors is consistently aligned over time while the challenging on-the-spot rotations (top-left) occur. For the sake of visibility, the ceilings in blue are not shown in the 3D point cloud.

6.6 Author-collected RGB-D Dataset

We provide the qualitative 3D reconstruction results generated by L-SLAM with the trajectories of a square corridor sequence obtained via other RGB-D SLAM methods, using trajectory lengths of 90 m, as shown in Fig. 13. L-SLAM maintains the structure-aware planar structure and significantly reduces the drift error in the final position compared to DVO-SLAM, ORB-SLAM2, and Structure SLAM. Owing to insufficient texture and structural conditions, BundleFusion and BAD SLAM fail to track the 6-DoF camera pose from the first image frame. There is no sudden jump or broken estimated camera trajectory from ORB-SLAM2 because we intentionally turn off the loop closing module. The drift error of ORB-SLAM2, however, gradually increases over time; the start and end points meet only with the proposed SLAM approach, with a final drift error under 0.7 % in this multiple-loop sequence. Although DPP-SLAM [8] shows the second best trajectory estimation results, it only works well in such a 2-D environment with little change in camera height; otherwise, it fails in all sequences from ICL-NUIM and TUM RGB-D dataset. With L-SLAM, the starting and ending points nearly match without loop closure detection; for the others, they do not. Figure 14 shows an approximately 120-m long corridor trajectory consisting of the forward camera motion and on-the-spot rotations. We demonstrate that L-SLAM can accurately track the camera pose and the global infinite planes in the map by preserving

Fig. 15. **Augmented reality (AR) applications**. AR implementation and rendering results on the author-collected RGB-D dataset (a,b), and the ICL-NUIM dataset (c) with the animals, ISS, and sofa 3D objects. We exploit the 6-DoF camera pose tracking obtained by our proposed SLAM method as key information to render the 3D virtual objects in the real world. Note that any arbitrary 3D objects can be used.

the planar geometric structure of indoor environments in a much more efficient and cheaper way within a linear KF framework.

6.7 Augmented Reality with Linear RGB-D SLAM

We further apply the proposed L-SLAM to AR to effectively demonstrate its practical usefulness. Currently, most commercial VR/AR products such as Oculus Rift and HTC Vive must use external devices to track the 3-DoF translational movements of the head; however, the AR implemented using the proposed L-SLAM algorithm enables full 6-DoF head tracking only with the onboard RGB-D sensor similar to HoloLens, which is one of the most advanced AR headsets. The only requirements of the proposed method are the highly-planar environments, and such geometric characteristics can be found easily in most structured indoor environments.

To perceptually obtain a better assessment, we carefully select a 3D object fixed to the wall or floor in the tested environments. We obtain 3D models of the international space station (ISS), Elk's head, and Hiroshima sofa from the 3D Warehouse website [63], and render the 3D objects as an image with the Open Scene Graph [64]. Figure 15 shows a consistent view of the 3D models irrespective of where we look, because of the accurate 6-DoF camera motion tracking with respect to the current structured environments from the proposed SLAM method, suggesting a potential for VR/AR applications.

7 Discussion

7.1 Degenerate Cases

Slant Plane. A large slant plane may exist in the structured environments under consideration, and we can consider a slant plane as an outlier distribution that does not follow the Atlanta world. In the proposed L-SLAM framework, our tracking-by-detection can handle this outlier distribution to a certain level. More specifically, when we encounter a slant plane, the dominant direction supporting outlier distribution of the slant plane can be detected as a local AF in the detection step; however, this outlier dominant direction will be filtered out during the association step. Because it does not follow the tracked AF (*i.e.*, global AF). Thus, we can handle this slant plane issue. We test this outlier case on the synthetic data (*cf.*, Sec. 6.2), where our method shows stable tracking-by-detection performance. Please refer to the supplementary video.

However, if we encounter this kind of slant plane at the first frame (initialization step), our method will recognize this outlier direction as a global AF, resulting in the failure of L-SLAM. We believe that this is a rare real-world scenario that can be easily avoided.

Insufficient Plane. Depending on the viewpoint, we may observe only vertical planes (walls); insufficient plane features. The proposed L-SLAM framework based on Kalman filter predicts the camera position and updates it using visible planes (*i.e.*, measurement). Thus, L-SLAM can update the camera position from vertical planes even if neither the ground plane nor ceiling plane (horizontal planes[7]) visible in the image. However, it may not be effective for refining the vertical position.

Fortunately, the proposed L-SLAM framework can compensate for the vertical position using the global planar map. Specifically, whenever we recognize any horizontal plane matching with planar landmarks in the global planar map, we can properly refine the vertical position *w.r.t.* the horizontal plane without any loop closure, which is one of our contributions.

7.2 Limitations

Lack of Structural Regularities. Camera rotation estimation relies on the detection and tracking of the dominant directions of man-made scenes in L-SLAM. Thus, L-SLAM may fail to if enough planes or vanishing directions cannot be detected. Specifically, L-SLAM needs to track two dominant directions from surface normals or vanishing directions at least. Otherwise, we cannot accurately estimate the camera rotation.

In this work, we assume that structured environments follow the AW assumption, which can describe a given scene as floor (ceiling) planes and a set of walls orthogonal to the floor. It is a natural and reasonable assumption in structured environments; moreover, compared to recent works [61, 65] limited by the Manhattan world, the proposed L-SLAM works in a more general environments, i.e., the Atlanta world. Thus, we believe that the structured environments under consideration usually have enough planes or vanishing directions.

However, the camera might sometimes look only one plane without line features, which is an extreme case and rarely occurs. Fortunately, in the case of a single plane for consecutive frames, we can assume that the rotational motion of the camera remains unchanged. Thus, when we sequentially observe one plane (*i.e.*, only one Atlanta

7. Any flat surface on objects (*e.g.*, table) can be a horizontal plane. It is not limited to ground or ceiling planes.

direction is tracked), our method maintains the previous rotation for certain frames (we set the maximum to 50 frames). This scheme might cause rotation errors; however, when we recover or observe additional Atlanta direction, our approach can align the global AF and observed local AF, which compensates for the potential rotation error.

Detection Runtime. In the proposed SLAM framework, we utilize the BnB-based Atlanta detection approach [43], robust to outliers but suffering from high computational complexity. A naïve way to alleviate the complexity would be to optimize the implementation (*i.e.*, MATLAB to C++). For a more advanced approach, we can exploit the estimated vertical direction as a priori information and then detect potential horizontal directions only. Joo *et al*. [43] showed that fewer DoFs could significantly reduce the computational time; that is, by using prior information of 1-DoF of the vertical direction, we can efficiently restrict the search space, which relieves the overall computational time of Atlanta detection. It could result in a less accurate detection performance, but the association step in the tracking-by-detection can compensate for this phenomenon.

It should be noted that our main contribution is to design a SLAM framework using structural regularities of the structured environments. In other words, we can replace this detection module with any advanced or efficient one.

7.3 L_{MW}-SLAM vs. L_{AW}-SLAM

In general, L_{MW}-SLAM and L_{AW}-SLAM show a small and reasonable performance gap, as shown in Table 4 quantitatively and Fig. 11(c) qualitatively. However, their performances sometimes have a large difference between them. For example, in *of-kt0n*, L_{MW}-SLAM shows better performance, but vice versa in *lr-kt3n*.

Basically, L_{MW}-SLAM exploits three orthogonal directions supporting the Manhattan world, while L_{AW}-SLAM estimates the underlying unknown structural regularities of Atlanta world within the SLAM framework. In other words, in contrast to L_{MW}-SLAM, L_{AW}-SLAM estimates unknown additional horizontal directions, which allows us to perceive additional planar measurements and formulate a linear SLAM framework under the Atlanta world. On the contrary, it may generate inaccurate planar measurements even though several validation steps (*cf*., Sec. 5.1) are involved, which can cause performance degradation. We believe that this is a trade-off between L_{MW}-SLAM and L_{AW}-SLAM.

8 Conclusion

We present a new, linear KF SLAM formulation that jointly estimates the camera position and global infinite planes in the map by compensating for the rotational motion of the camera from structural regularities in the planar environments. By measuring the distance from the planar features, we update the 3-DoF camera translation and the position of the associated global planes in the map. In addition, we have seamlessly extended the linear SLAM for the Manhattan world into the more general Atlanta world via a robust and efficient tracking-by-detection algorithm. The extensive evaluation has demonstrated the superior performance of the proposed SLAM algorithm in a variety of planar environments, especially in maintaining its efficiency without the use of expensive nonlinear SLAM techniques.

Acknowledgments

Kyungdon Joo was supported by the Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (No.2020-0-01336, Artificial Intelligence Graduate School Program (UNIST)) and the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (No. NRF-2021R1C1C1005723). Pyojin Kim was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (No. NRF-2021R1F1A1061397) and Sookmyung Women's University Research Grants (1-2003-2015).

References

[1] T. Taketomi, H. Uchiyama, and S. Ikeda, "Visual SLAM algorithms: A survey from 2010 to 2016," *IPSJ Transactions on Computer Vision and Applications*, vol. 9, no. 1, p. 16, 2017.
[2] F. Rameau, H. Ha, K. Joo, J. Choi, K. Park, and I. S. Kweon, "A real-time augmented reality system to see-through cars," *IEEE Transactions on Visualization and Computer Graphics (TVCG)*, vol. 22, no. 11, pp. 2395–2404, 2016.
[3] Y. Lin, F. Gao, T. Qin, W. Gao, T. Liu, W. Wu, Z. Yang, and S. Shen, "Autonomous aerial navigation using monocular visual-inertial fusion," *Journal of Field Robotics (JFR)*, vol. 35, no. 1, pp. 23–51, 2018.
[4] P. Kim, B. Coltin, and H. J. Kim, "Linear RGB-D SLAM for planar environments," in *European Conference on Computer Vision (ECCV)*, 2018.
[5] C. Kerl, J. Sturm, and D. Cremers, "Dense visual SLAM for RGB-D cameras," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2013.
[6] R. Mur-Artal and J. D. Tardós, "ORB-SLAM2: an open-source SLAM system for monocular, stereo and RGB-D cameras," *IEEE Transactions on Robotics (TRO)*, vol. 33, no. 5, pp. 1255–1262, 2017.
[7] S. Yang, Y. Song, M. Kaess, and S. Scherer, "Pop-up SLAM: Semantic monocular plane SLAM for low-texture environments," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2016.
[8] P.-H. Le and J. Košecka, "Dense piecewise planar RGB-D SLAM for indoor environments," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2017.
[9] M. Hsiao, E. Westman, G. Zhang, and M. Kaess, "Keyframe-based dense planar SLAM," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2017.
[10] A. Handa, T. Whelan, J. McDonald, and A. Davison, "A benchmark for RGB-D visual odometry, 3D reconstruction and SLAM," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2014.
[11] J. M. Coughlan and A. L. Yuille, "Manhattan world: Compass direction from a single image by bayesian inference," in *IEEE International Conference on Computer Vision (ICCV)*, 1999.
[12] G. Schindler and F. Dellaert, "Atlanta world: An expectation maximization framework for simultaneous low-level edge grouping and camera calibration in complex man-made environments," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2004.
[13] J. Straub, O. Freifeld, G. Rosman, J. J. Leonard, and J. W. Fisher, "The Manhattan frame model–Manhattan world inference in the space of surface normals," *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2017.
[14] J. Xiao, A. Owens, and A. Torralba, "SUN3D: A database of big spaces reconstructed using sfm and object labels," in *IEEE International Conference on Computer Vision (ICCV)*, 2013.
[15] S. Gupta, P. Arbelaez, and J. Malik, "Perceptual organization and recognition of indoor scenes from RGB-D images," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2013.

[16] S. Song, S. P. Lichtenberg, and J. Xiao, "Sun RGB-D: A RGB-D scene understanding benchmark suite," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[17] L. Carlone, R. Tron, K. Daniilidis, and F. Dellaert, "Initialization techniques for 3D SLAM: a survey on rotation estimation and its use in pose graph optimization," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2015.

[18] P. Kim, B. Coltin, and H. J. Kim, "Low-drift visual odometry in structured environments by decoupling rotational and transnational motion," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2018.

[19] K. Joo, T.-H. Oh, F. Rameau, J.-C. Bazin, and I. S. Kweon, "Linear RGB-D SLAM for Atlanta World," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2020.

[20] J. Engel, T. Schöps, and D. Cremers, "LSD-SLAM: Large-scale direct monocular SLAM," in *European Conference on Computer Vision (ECCV)*, 2014.

[21] R. Wang, M. Schwörer, and D. Cremers, "Stereo DSO: Large-scale direct sparse visual odometry with stereo cameras," in *IEEE International Conference on Computer Vision (ICCV)*, 2017.

[22] A. J. Davison, I. D. Reid, N. D. Molton, and O. Stasse, "MonoSLAM: Real-time single camera slam," *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, no. 6, pp. 1052–1067, 2007.

[23] A. P. Gee, D. Chekhlov, W. W. Mayol-Cuevas, and A. Calway, "Discovering planes and collapsing the state space in visual SLAM," in *British Machine Vision Conference (BMVC)*, 2007.

[24] A. P. Gee, D. Chekhlov, A. Calway, and W. Mayol-Cuevas, "Discovering higher level structure in visual SLAM," *IEEE Transactions on Robotics (TRO)*, vol. 24, no. 5, pp. 980–990, 2008.

[25] F. Servant, E. Marchand, P. Houlier, and I. Marchal, "Visual planes-based simultaneous localization and model refinement for augmented reality," in *IEEE International Conference on Pattern Recognition (ICPR)*, 2008.

[26] J. Martínez-Carranza and A. Calway, "Unifying planar and point mapping in monocular SLAM," in *British Machine Vision Conference (BMVC)*, 2010.

[27] J. Weingarten and R. Siegwart, "3D SLAM using planar segments," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2006.

[28] T. Bailey, J. Nieto, J. Guivant, M. Stevens, and E. Nebot, "Consistency of the EKF-SLAM algorithm," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2006.

[29] M. Kaess, A. Ranganathan, and F. Dellaert, "iSAM: Incremental smoothing and mapping," *IEEE Transactions on Robotics (TRO)*, vol. 24, no. 6, pp. 1365–1378, 2008.

[30] G. Grisetti, R. Kummerle, C. Stachniss, and W. Burgard, "A tutorial on graph-based SLAM," *IEEE Intelligent Transportation Systems Magazine*, vol. 2, no. 4, pp. 31–43, 2010.

[31] R. Kümmerle, G. Grisetti, H. Strasdat, K. Konolige, and W. Burgard, "g2o: A general framework for graph optimization," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2011.

[32] M. Kaess, "Simultaneous localization and mapping with infinite planes," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2015.

[33] L. Ma, C. Kerl, J. Stückler, and D. Cremers, "CPA-SLAM: Consistent plane-model alignment for direct RGB-D SLAM," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2016.

[34] P. Kim, B. Coltin, and H. J. Kim, "Indoor RGB-D compass from a single line and plane," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[35] H. Li, Y. Xing, J. Zhao, J.-C. Bazin, Z. Liu, and Y.-H. Liu, "Leveraging structural regularity of Atlanta world for monocular SLAM," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2019.

[36] D. Zou, Y. Wu, L. Pei, H. Ling, and W. Yu, "Structvio: visual-inertial odometry with structural regularity of man-made environments," *IEEE Transactions on Robotics (TRO)*, 2019.

[37] J. Straub, N. Bhandari, J. J. Leonard, and J. W. Fisher, "Real-time Manhattan world rotation estimation in 3D," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2015.

[38] P. Kim, B. Coltin, and H. J. Kim, "Visual odometry with drift-free rotation estimation using indoor scene regularities," in *British Machine Vision Conference (BMVC)*, 2017.

[39] Z. Jia, A. Gallagher, A. Saxena, and T. Chen, "3D-based reasoning with blocks, support, and stability," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2013.

[40] C. Zou, A. Colburn, Q. Shan, and D. Hoiem, "Layoutnet: Reconstructing the 3d room layout from a single rgb image," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[41] S.-T. Yang, F.-E. Wang, C.-H. Peng, P. Wonka, M. Sun, and H.-K. Chu, "DuLa-net: A dual-projection network for estimating room layouts from a single rgb panorama," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[42] K. Joo, T.-H. Oh, I. S. Kweon, and J.-C. Bazin, "Globally optimal inlier set maximization for Atlanta frame estimation," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[43] K. Joo, T.-H. Oh, I. S. Kweon, and J.-C. Bazin, "Globally optimal inlier set maximization for Atlanta world understanding," *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, vol. 42, no. 10, pp. 2656–2669, 2020.

[44] Y. Zhou, L. Kneip, C. Rodriguez, and H. Li, "Divide and conquer: Efficient density-based tracking of 3D sensors in Manhattan worlds," in *Asian Conference on Computer Vision (ACCV)*, 2016.

[45] Y. Cheng, "Mean shift, mode seeking, and clustering," *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, vol. 17, no. 8, pp. 790–799, 1995.

[46] Z. Kalal, K. Mikolajczyk, and J. Matas, "Tracking-learning-detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, vol. 34, no. 7, pp. 1409–1422, 2011.

[47] R. Hartley, K. Aftab, and J. Trumpf, "L1 rotation averaging using the Weiszfeld algorithm," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2011.

[48] R. Hartley, J. Trumpf, Y. Dai, and H. Li, "Rotation averaging," *International Journal of Computer Vision (IJCV)*, vol. 103, no. 3, pp. 267–305, 2013.

[49] A. Chang, A. Dai, T. Funkhouser, M. Halber, M. Niebner, M. Savva, S. Song, A. Zeng, and Y. Zhang, "Matterport3d: Learning from rgb-d data in indoor environments," in *IEEE International Conference on 3D Vision (3DV)*, 2017.

[50] M. Y. Yang and W. Förstner, "Plane detection in point cloud data," in *Proceedings of the 2nd int conf on machine control guidance, Bonn*, 2010.

[51] C. J. Taylor and A. Cowley, "Parsing indoor scenes using RGB-D imagery," in *Robotics: Science and Systems (RSS)*, 2013.

[52] H. Strasdat, J. Montiel, and A. J. Davison, "Real-time monocular SLAM: Why filter?" in *IEEE International Conference on Robotics and Automation (ICRA)*, 2010.

[53] D. Simon, *Optimal state estimation: Kalman, H infinity, and nonlinear approaches*. John Wiley & Sons, 2006.

[54] F. Camposeco and M. Pollefeys, "Using vanishing points to improve visual-inertial odometry," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2015.

[55] J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers, "A benchmark for the evaluation of RGB-D SLAM systems," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2012.

[56] Y. Lu and D. Song, "Robust RGB-D odometry using point and line features," in *IEEE International Conference on Computer Vision (ICCV)*, 2015.

[57] V. A. Prisacariu, O. Kähler, S. Golodetz, M. Sapienza, T. Cavallari, P. H. Torr, and D. W. Murray, "Infinitam v3: A framework for large-scale 3d reconstruction with loop closure," *arXiv preprint arXiv:1708.00783*, 2017.

[58] A. Dai, M. Nießner, M. Zollhöfer, S. Izadi, and C. Theobalt, "Bundlefusion: Real-time globally consistent 3d reconstruction using on-the-fly surface reintegration," *ACM Transactions on Graphics (TOG)*, vol. 36, no. 4, p. 1, 2017.

[59] T. Whelan, R. F. Salas-Moreno, B. Glocker, A. J. Davison, and S. Leutenegger, "Elasticfusion: Real-time dense slam and light source estimation," *International Journal of Robotics Research (IJRR)*, vol. 35, no. 14, pp. 1697–1716, 2016.

[60] T. Schops, T. Sattler, and M. Pollefeys, "BAD SLAM: Bundle adjusted direct RGB-D SLAM," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[61] Y. Li, N. Brasch, Y. Wang, N. Navab, and F. Tombari, "Structure-slam: Low-drift monocular slam in indoor environments," *IEEE Robotics and Automation Letters (RA-L)*, vol. 5, no. 4, pp. 6583–6590, 2020.

[62] C. Jia and B. L. Evans, "Constrained 3D rotation smoothing via global manifold regression for video stabilization," *IEEE Transactions on Signal Processing (TSP)*, vol. 62, no. 13, pp. 3293–3304, 2014.
[63] https://3dwarehouse.sketchup.com/?hl=en.
[64] T. Hassner, L. Assif, and L. Wolf, "When standard RANSAC is not enough: cross-media visual matching with hypothesis relevancy," *Machine Vision and Applications*, 2014.
[65] Y. Li, R. Yunus, N. Brasch, N. Navab, and F. Tombari, "Rgb-d slam with structural regularities," *arXiv preprint arXiv:2010.07997*, 2020.

Kyungdon Joo is an assistant professor of the Department of Computer Science and Engineering and the Artificial Intelligence Graduate School at UNIST, South Korea. He received his B.E. degree in School of Electrical and Computer Engineering from University of Seoul, South Korea in 2012, and the M.S. and Ph.D. degrees in Robotics Program and School of Electrical Engineering from KAIST, South Korea in 2014 and 2019, respectively. Before joining UNIST, he was a postdoctoral researcher at CMU RI, US. He was a member of "Team KAIST," which won first place in the DARPA Robotics Challenge Finals 2015. He was a research intern at Oculus research (Facebook Reality Labs), Pittsburgh in 2017. His research interests include robust computer vision, geometry, and machine learning.

Pyojin Kim is an assistant professor of the Department of Mechanical Systems Engineering at Sookmyung Women's University, South Korea. He received his B.S. degree in Mechanical Engineering from Yonsei University in 2013, and the M.S. and Ph.D. degrees in the Department of Mechanical and Aerospace Engineering at Seoul National University, Seoul, South Korea in 2015 and 2019, respectively. Before joining Sookmyung Women's University, he was a postdoctoral researcher at Simon Fraser University, Canada. He was a research intern at Google (ARCore Tracking), Mountain View in 2018. His research interests include indoor localization, 3D computer vision, visual odometry, and visual SLAM for robotics.

Martial Hebert Martial Hebert is a professor at the Robotics Institute, Carnegie Mellon University. His current research interests include object recognition in images, video, and range data, scene understanding using context representations, and model construction from images and 3D data. His group has explored applications in the areas of autonomous mobile robots, both in indoor and in unstructured, outdoor environments, automatic model building for 3D content generation, and video monitoring. He has served on the program committees of the major conferences in computer vision and robotics. He is a member of the IEEE.

In So Kweon received his B.S. and M.S. degrees in mechanical design and production engineering from Seoul National University, South Korea, in 1981 and 1983, respectively, and the Ph.D. degree in robotics from the Robotics Institute, Carnegie Mellon University, USA, in 1990. He was with the Toshiba R&D Center, Japan, and he is currently a KEPCO chair professor with the Department of Electrical Engineering, since 1992. He served as a program co-chair for ACCV'07 and a general chair for ACCV'12. He is on the honorary board of IJCV. He was a member of "Team KAIST," which won first place in the DARPA Robotics Challenge Finals 2015. He is a member of the IEEE and the KROS.

H. Jin Kim received her B.S. degree from the Korea Advanced Institute of Technology (KAIST) in 1995, and the M.S. and Ph.D. degrees in Mechanical Engineering from University of California, Berkeley (UC Berkeley), in 1999 and 2001, respectively. From 2002 to 2004, she was a Postdoctoral Researcher in Electrical Engineering and Computer Science (EECS), UC Berkeley. In September 2004, she joined the Department of Mechanical and Aerospace Engineering at Seoul National University, Seoul, South Korea, as an Assistant Professor where she is currently a Professor. Her research interests include intelligent control of robotic systems and motion planning.