This ECCV 2018 paper, provided here by the Computer Vision Foundation, is the author-created version.
The content of this paper is identical to the content of the officially published ECCV 2018
LNCS version of the paper as available on SpringerLink: https://link.springer.com/conference/eccv

# Linear RGB-D SLAM for Planar Environments

Pyojin Kim[1], Brian Coltin[2], and H. Jin Kim[1]

[1] ASRI, Seoul National University, South Korea
[2] SGT, Inc., NASA Ames Research Center, USA
{rlavywls,hjinkim}@snu.ac.kr, brian.j.coltin@nasa.gov

**Abstract.** We propose a new formulation for including orthogonal planar features as a global model into a linear SLAM approach based on sequential Bayesian filtering. Previous planar SLAM algorithms estimate the camera poses and multiple landmark planes in a pose graph optimization. However, since it is formulated as a high dimensional nonlinear optimization problem, there is no guarantee the algorithm will converge to the global optimum. To overcome these limitations, we present a new SLAM method that jointly estimates camera position and planar landmarks in the map within a linear Kalman filter framework. It is rotations that make the SLAM problem highly nonlinear. Therefore, we solve for the rotational motion of the camera using structural regularities in the Manhattan world (MW), resulting in a linear SLAM formulation. We test our algorithm on standard RGB-D benchmarks as well as additional large indoor environments, demonstrating comparable performance to other state-of-the-art SLAM methods *without* the use of expensive nonlinear optimization.

**Keywords:** Linear SLAM, Manhattan World, Bayesian Filtering

## 1 Introduction

Visual simultaneous localization and mapping (vSLAM) 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. They are fundamental building blocks for various applications from autonomous robots to virtual and augmented reality (VR/AR).

Many typical visual RGB-D SLAM approaches such as DVO-SLAM [17] and ORB-SLAM2 [23], which are based on the pose graph optimization [19], have shown promising results in the environments with rich texture. However, they fare poorly in textureless scenes, which are commonly encountered in indoor environments with large planar structures [13]. They also rely on pose graph optimization methods, which are computationally expensive, and sometimes fail.

For working well in low-texture environments, recent visual SLAM methods [13, 20, 33] utilize additional geometric information like planar features. They combine plane measurements and scene layout with graph-based SLAM approaches [10, 16] to improve robustness and accuracy. Although these SLAM

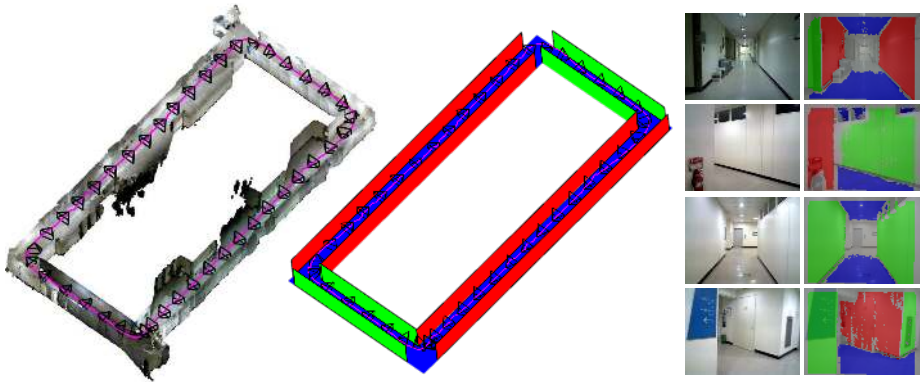

**Fig. 1.** Linear RGB-D SLAM: L-SLAM generates a consistent global planar map using a linear Kalman filter framework instead of expensive pose graph optimization. Left: Accumulated 3D point cloud is rendered by back-projecting the RGB-D images from the estimated camera trajectory with L-SLAM. Right: The detected orthogonal planar features are overlaid on top of the RGB images. Note that we omit the ceiling planar features for visibility.

approaches show better accuracy for low-texture environments, there are some limitations: they are still dependent on the pose graph optimization, which is the non-convex and nonlinear optimization problem [4]. Since their SLAM is formulated as a high dimensional nonlinear optimization problem for jointly refining 6-DoF camera poses and multiple landmarks, there is no guarantee that the algorithm can converge to the global optimum [34]. Also, if the nonlinearity of pose graph optimization is too high due to the rotational components of the camera and the landmarks, they will fail to find the true solution.

To address these issues, we propose *Linear RGB-D SLAM* (L-SLAM), a novel method that jointly estimates camera position and planar landmarks in the map within a linear Bayesian filter as shown in Fig. 1. To separate the need for rotational motion estimation, which is a main source of nonlinearity in SLAM formulation, from the SLAM problem, we first track drift-free 3-DoF rotation and initial 3-DoF translational movement separately using Manhattan world (MW) assumption [5] from VO algorithm [18]. Given the absolute camera orientation, L-SLAM identifies the horizontal and vertical planes in structured environments, and measures the distance to these orthogonal planes from the current camera pose at every frame. With the distance measurements from the orthogonal planes, we simultaneously update the 3-DoF camera translation and the 1-D distance of the associated global planes in the map within a linear Kalman filter (KF) framework. We present a simple, linear KF SLAM formulation by fully compensating for the 3-DoF rotational camera motion obtained from [18], resulting in very low computational complexity while working well in textureless regions.

Extensive evaluations show that L-SLAM produces comparable estimation results compared to other state-of-the-art SLAM methods without expensive SLAM techniques (loop detection, pose graph optimization). Furthermore, we apply L-SLAM to augmented reality (AR) without any external infrastructure. We highlight our main contributions below:

- We develop an orthogonal plane detection method in structured environments when the absolute camera orientation is given.
- We propose a new, linear KF SLAM formulation for localizing the camera translation and mapping the global infinite planes.
- We evaluate L-SLAM on the RGB-D benchmark datasets from room-size to building-size with other state-of-the-art SLAM methods.
- We implement augmented reality (AR) using L-SLAM.

## 2 Related Work

Visual SLAM methods have been actively studied in the robotics and computer vision communities for the past two decades due to its importance in various applications such as autonomous UAV to augmented reality (AR). From the vast literature in 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 (indirect) or high gradient pixels (direct). Representatives of them are direct LSD-SLAM [7], DSO [30], and feature-based ORB-SLAM2 [23]. But their performance can be severely degraded in challenging low-texture environments.

Some research in early years of SLAM exploits planes within an extended Kalman filter (EKF) based SLAM approaches [6]. In [8, 9], tracked points lying on the same plane are reformulated as a planar feature to reduce the state size in EKF-SLAM. [24] includes planar features in the EKF state vector with a priori structural information. [22] proposes a unified parameterization for points and planes within an EKF monocular SLAM. [31] uses 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 [2] because the estimation of camera rotation and translation together results in non-linearity of the measurement model. Also, since both distance and orientation are used to represent the planar features, the state vector and covariance matrix size (computational complexity) grows rapidly over time, which limits applications to a small room-scale environment.

Several recent planar SLAM studies apply graph-based SLAM [10, 16, 19], which is a nonlinear and non-convex optimization problem [4]. To avoid singularities in pose graph optimization, [15] presents a minimal plane representation of infinite planes. With the help of the GPU, [21] tracks keyframe camera pose and global plane model by performing direct image alignment and global graph optimization. [33] performs graph-based SLAM with the plane measurements coming from scene layout understanding using convolutional neural networks (CNN). In [13], a keyframe-based factor graph optimization is performed to

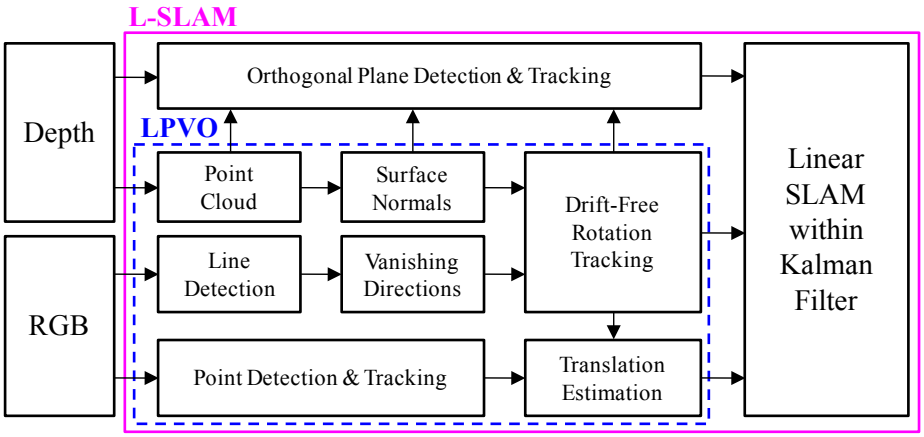


**Fig. 2.** Overview of the complete L-SLAM algorithm.

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 rotation and translation together [4].

The most relevant planar SLAM approach to the proposed L-SLAM is [20], which first estimates the 3-DoF camera rotation by recognizing the piecewise planar models, and utilizes graph SLAM optimization to recover the 2-DoF camera translation. However, unlike 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 Proposed Method

Our proposed L-SLAM method builds on the previous *Line and Plane based Visual Odometry* (LPVO) algorithm [18]. However, while LPVO cannot avoid drift over time due to the nature of VO, we extend it to the SLAM formulation in which the planar features are directly modeled as landmarks in order to further constrain the camera motion and significantly reduce drift in translation.

We start by giving a brief description of the previous LPVO algorithm in Sec. 3.1. As a first contribution, we present a method of detecting orthogonal planes in structured environments in Sec. 3.2, which plays an important role in our SLAM method. Next, we introduce L-SLAM, a novel SLAM approach using orthogonal planar features within a linear Kalman filter (KF) framework in Sec. 3.3. Fig. 2 shows an overview of the L-SLAM.

### 3.1 Line and Plane based Visual Odometry

We summarize the LPVO algorithm briefly (for full details, refer to [18]). LPVO has two main steps: 1) structural regularities (Manhattan frame) are tracked to

obtain the drift-free rotation with a SO(3)-manifold constrained mean shift algorithm; and 2) it estimates translation by minimizing a de-rotated reprojection error from tracked points.

The core of the drift-free rotation estimation in LPVO is to track the Manhattan frame (MF) jointly from both lines and planes by exploiting environmental regularities. Given the density distribution of vanishing directions from lines and surface normals from planes on the Gaussian sphere $\mathbb{S}^2$, LPVO infers the mean of the directional vector distribution around each dominant Manhattan frame axis through a mean shift algorithm in the tangent plane $\mathbb{R}^2$ with a Gaussian kernel. The modes found by the mean shift are projected onto the SO(3) manifold to maintain orthogonality, resulting in the absolute orientation estimate of the camera with respect to the Manhattan world.

For the 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. LPVO estimates 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.

### 3.2 Orthogonal Plane Detection

Once the Manhattan world orientation of the scene with respect to the camera pose has been established from LPVO, we can easily identify the dominant orthogonal planes in current structured environments. Given the surface normals for each pixel used when we track the Manhattan frame in LPVO, we find the relevant normal vectors inside a conic section of each Manhattan frame axis. We perform the plane RANSAC [32] with the pixels corresponding to the surface normals near each axis of the tracked Manhattan frame. We model the plane [29] as:

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

where $X, Y, Z$ denote the 3D coordinates, $u, v, w$ correspond to the normalized image coordinates and the measured disparity at that coordinate. $n_x, n_y, n_z$ 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 the least-squares sense.

If the angle difference between the normal vector of the plane and one of the three Manhattan frame axes is less than 5 degrees, we refit this plane again to a set of disparity values ($w$) subject to the constraint that it must be parallel to the corresponding Manhattan frame axis. We compute the optimal scale factor in the least-squares sense that minimizes:

$$s^* = \arg\min_s \| s\,(r_x u + r_y v + r_z) - w \| \tag{2}$$

where $s$ is the scale factor representing the reciprocal of the distance (offset) from the plane to the origin, and $r_x, r_y, r_z$ denote the unit vector of the corresponding

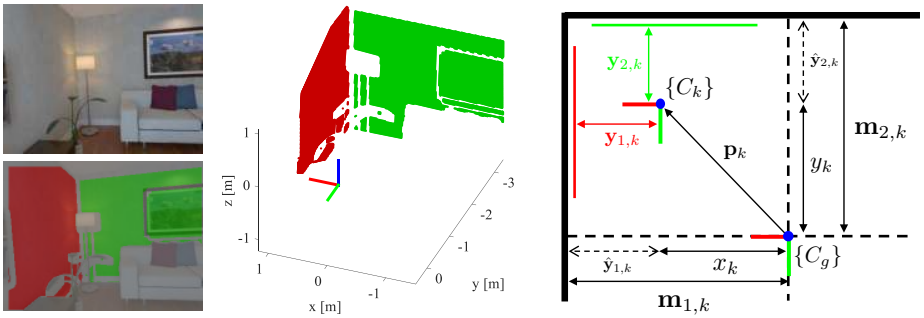


**Fig. 3.** Results of orthogonal plane detection are overlaid on top of the RGB images (left). Color-coded orthogonal planar features are drawn in a 3-D space (middle), and top view (right). The detailed descriptions of each variable, definition of the state vector, and the measurement model are given in Sec. 3.3.

Manhattan frame axis. In this way, we can find the orthogonal planar features in the scene whose normals are aligned with the tracked Manhattan frame as shown in Fig. 3.

### 3.3 Linear RGB-D SLAM

**KF State Vector Definition.** The state vector in the KF consists of the current 3-DoF translational motion of the camera and a 1-D representation of the orthogonal planar features in the map. We denote the state vector by $\mathbf{X}$ with its associated covariance $\mathbf{P}$:

$$\mathbf{X} = \begin{bmatrix}\mathbf{p}^\top \ \mathbf{m}_1 \cdots \mathbf{m}_n\end{bmatrix}^\top \in \mathbb{R}^{3+n} \qquad \mathbf{P} = \begin{bmatrix}\mathbf{P_{pp}} & \mathbf{P_{pm}} \\ \mathbf{P_{mp}} & \mathbf{P_{mm}}\end{bmatrix} \in \mathbb{R}^{(3+n)\times(3+n)} \tag{3}$$

where $\mathbf{p} = \begin{bmatrix}x \ y \ z\end{bmatrix}^\top \in \mathbb{R}^3$ denotes the 3-DoF camera translation in the global Manhattan map frame where the rotation of the camera is completely compensated. Unlike the 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 [4] because we already obtain accurate and drift-free camera rotation from LPVO in Sec. 3.1.

The map $\mathbf{m}_i = [o_i] \in \mathbb{R}^1$ denotes the 1-D distance (offset) of the orthogonal planar feature from the origin in the global Manhattan map frame, and $n$ is the number of orthogonal planes in the global map. Although each orthogonal planar feature in Sec. 3.2 consists of the 1-D distance and the alignment for the Manhattan frame, we only track and update the distance since the alignment of the orthogonal planes does not change over time. A newly detected orthogonal planar feature $\mathbf{m}_{new}$ is additionally augmented after the last map component of the state vector. Note that there are no variables related to the camera or plane orientation in the state vector $\mathbf{X}$, resulting in a linear KF formulation.

**Table 1.** Advantages of L-SLAM over Existing EKF-SLAM Methods

| | L-SLAM (Ours) | [9] | [8] | [24] | [22] |
|---|---|---|---|---|---|
| State Size | $3+n$ | $7+7n$ | $7+9n$ | $15+3n$ | $12+10n$ |
| Linearity | Linear | Nonlinear | Nonlinear | Nonlinear | Nonlinear |

One of the problems of using the Kalman filter (KF) in SLAM is the quadratic update complexity in the number of features that can limit the ability to use multiple measurements [26]. Since 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 other EKF-SLAM approaches as shown in Table 1. While other EKF-SLAM methods [8, 9, 22, 24] in Table 1 represent the plane using a 3 to 10-D vector, the proposed method models the planar feature with only one parameter (offset), resulting in very low complexity. If the number of the planar features ($n$) is 10, the state size of the proposed method is about ten times smaller than that of Martinez's EKF-SLAM method [22], meaning the EKF update is expected to be $\sim$100 times faster.

**Process Model.** We predict the next state based on the 3-DoF translational movement estimated from LPVO between the consecutive frames. We propagate the 3-DoF camera translation, and assume the map does not change. Our process model can be written as follows:

$$\mathbf{X}_k = \mathbf{F}\mathbf{X}_{k-1} + \begin{bmatrix}\triangle\mathbf{p}_{k,k-1}^\top \ \mathbf{0}_{1\times n}\end{bmatrix}^\top \tag{4}$$

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 from LPVO.

**Measurement Model.** We update the state vector in the KF by observing the distance between the currently detected orthogonal planar features and the current camera pose. A measurement model $\mathbf{y}$ for the $\mathbf{m}_i$ is defined by:

$$\mathbf{y} = \begin{bmatrix}\mathbf{m}_1 - x \\ \mathbf{m}_2 - y \\ \mathbf{m}_3 - z \\ \vdots\end{bmatrix} = \mathbf{H}\mathbf{X} \in \mathbb{R}^m \qquad \mathbf{H} = \begin{bmatrix}-1 & 0 & 0 & 1\,0\,0\cdots \\ 0 & -1 & 0 & 0\,1\,0\cdots \\ 0 & 0 & -1 & 0\,0\,1\cdots \\ \vdots & \vdots & \vdots & \vdots\,\vdots\,\vdots\,\ddots\end{bmatrix} \in \mathbb{R}^{(m)\times(3+n)} \tag{5}$$

where $\mathbf{H}$ is the observation model which maps the state space into the observed space, and $m$ is the number of matched orthogonal planar features. For the sake of presentation, we assume that each orthogonal planar feature corresponds to the x or y or z axis of the Manhattan frame in the Eq. (5). A value of the measurement model $\mathbf{y}$ is the observed distance from the orthogonal planar features computed with the current state vector. We perform the KF update

(SLAM) for all associated orthogonal planes with the global planes in the map. Since all formulas and calculations are *perfectly* linear from the Eqs. (3) to (5), there is no local linearization error, and we can easily calculate the optimal Kalman gain [25]. In this manner, we can consistently track the 3-DoF camera translation and 1-D planar map position efficiently and reliably.

Our KF SLAM algorithm relies on the drift-free rotation estimates from LPVO [18] in Sec. 3.1, which shows accurate and stable rotation tracking performance (about 0.2 degrees error in average) in structured environments. This small orientation error is treated as the measurement noise by the Kalman filter, which removes the need to explicitly take into account the correlations [3]. The measurement noise includes not only the error in orientation but also the distance measurement noise of the RGB-D camera. Currently, the measurement error is manually tuned to 2 cm.

**Planar Map Management.** At the beginning of L-SLAM, we initialize a state vector and its covariance with the orthogonal planar features detected at the first frame. When constructing a global planar map, we only utilize the orthogonal 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 orthogonal 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 orthogonal planes are detected.

## 4 Evaluation

We evaluate the proposed L-SLAM on various RGB-D datasets from room-size ($\sim$10 m) to building-size ($\sim$100 m) for planar environments:

- *ICL-NUIM* [11] 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 due to the low-texture and artificial noise in the depth images.
- *TUM RGB-D* [28] is the de facto standard RGB-D dataset for VO/vSLAM evaluation consisting of ground-truth camera poses and RGB-D images captured in room-scale environments with various objects.
- *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 and end at the same position to evaluate loop closing and consistency since 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 [23], DVO-SLAM [17], CPA-SLAM [21],

KDP-SLAM [13], and DPP-SLAM [20]. Unlike the proposed L-SLAM, which is based on a linear formulation, they all perform a high dimensional nonlinear pose graph optimization. We also show an improvement compared to LPVO [18], which our new SLAM approach builds on. Note that we test each SLAM method with the original source code provided by the authors while we include the result of CPA-SLAM and KDP-SLAM taken directly from [13].

We implement the proposed L-SLAM in unoptimized MATLAB code for fast prototyping. Our L-SLAM operates at above 20 Hz throughout the sequence on a desktop computer with an Intel Core i5 (3.20 GHz) and 8 GB memory, suggesting a potential of the proposed method when implemented in C/C++.

### 4.1 ICL-NUIM Dataset

**Table 2.** Evaluation Results of ATE RMSE (unit: m) on ICL-NUIM Benchmark

| 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 | – | – | – | – |
| LPVO | 0.015 | 0.039 | 0.034 | 0.102 | 0.061 | 0.052 | 0.039 | 0.030 |
| L-SLAM (Ours) | 0.012 | 0.027 | 0.053 | 0.143 | **0.020** | **0.015** | 0.026 | **0.011** |

We report the root mean square error (RMSE) of the absolute trajectory error (ATE) [28] for the resulting camera trajectories of all living room and office sequences with noise in Table 2. The smallest error for each sequence is highlighted. The results of the CPA-SLAM and KDP-SLAM for the office are not available. Although CPA-SLAM, which requires GPU for expensive computation, shows the best quantitative results in most living room sequences, L-SLAM presents comparable estimation results. We plot the estimated camera trajectories using L-SLAM in Fig. 4, showing that L-SLAM is comparable to other state-of-the-art SLAM approaches without a nonlinear pose graph optimization.

In the office sequences, L-SLAM achieves more accurate or similar performance to other SLAM methods since the office environments consist of sufficient orthogonal planar features. Reconstruction results of the 'office room' sequences are shown in Fig. 5. Although ORB-SLAM2 performs the best thanks to sufficient texture in 'of-kt2n', L-SLAM also performs nearly as well. The average ATE RMSE of L-SLAM is 0.038, while ORB-SLAM2, DVO-SLAM, CPA-SLAM, KDP-SLAM, and LPVO are 0.057, 0.197, 0.028, 0.053, and 0.046, respectively. Among the CPU-only RGB-D and planar SLAM methods (except for CPA-SLAM, which requires a GPU), L-SLAM presents the lowest average trajectory error. The resulting camera trajectories with L-SLAM are plotted in Fig. 4,

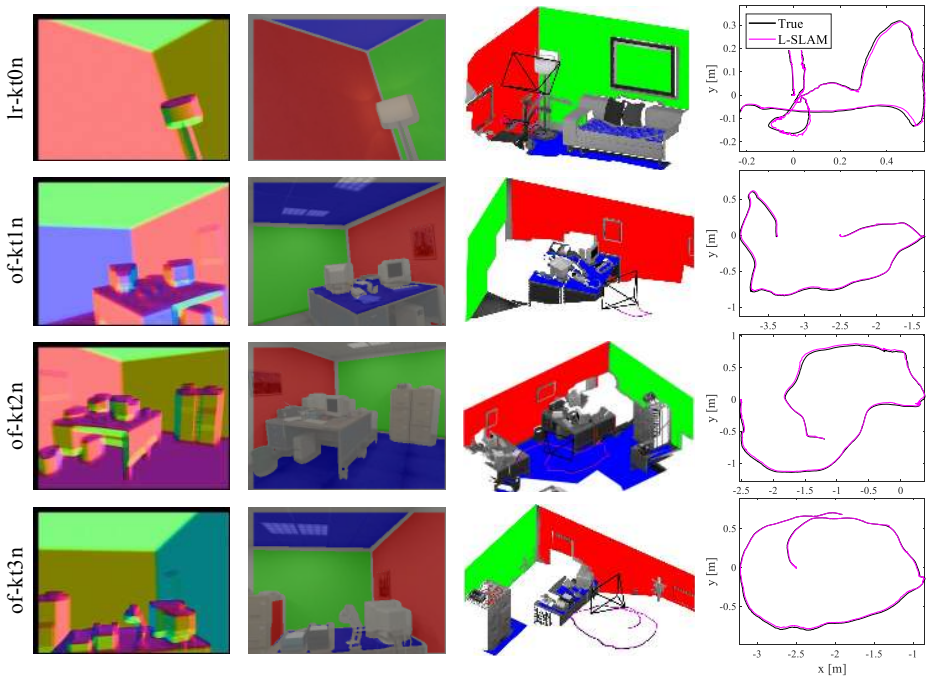


**Fig. 4.** Selected motion estimation results of the proposed algorithm in the ICL-NUIM dataset. The first column shows the per-pixel surface normal map with respect to the currently tracked Manhattan world. The second and third columns show the orthogonal planar features for mapping and localizing the camera position in the proposed SLAM algorithm. Vertical surfaces are red or green and horizontal surfaces are blue depending on their orientation. The magenta and black lines in the fourth column represent the estimated and the ground-truth trajectories, respectively.

showing that L-SLAM, with an efficient and linear KF, is comparable to other recent SLAM approaches especially for highly-planar environments.

### 4.2 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 [28]. Table 3 compares estimation results of the SLAM approaches. ORB-SLAM2 outperforms the proposed and other SLAM methods in texture-rich scenes such as 'fr3/str_tex_far', which is entirely expected as L-SLAM utilizes a much cheaper method. While L-SLAM shows comparable performance even in poorly-featured environments of Fig. 6, the accuracy of ORB-SLAM2 drops drastically, and the trajectory estimation fails (marked as $\times$ in Table 3). Although inaccurate planar distance measurements in L-SLAM sometimes cause slight performance degradation of LPVO, L-SLAM is generally more accurate than LPVO on average. The average

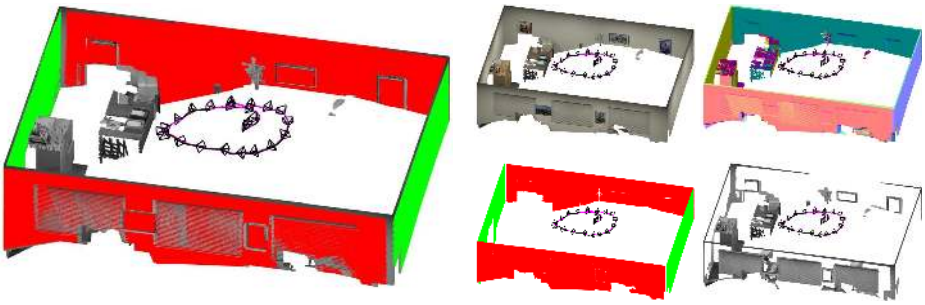

**Fig. 5.** Left: Synthetic scene 3D reconstruction of an office room from the ICL-NUIM dataset, displaying both planar and non-planar regions with the estimated (magenta) and the ground-truth (black) trajectories. Right, in clockwise order: Color output, surface normal map, non-planar regions only with gray scale, and orthogonal planar regions only with RGB scale. The ceilings are not shown for visibility.

**Table 3.** 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 |
| 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** |

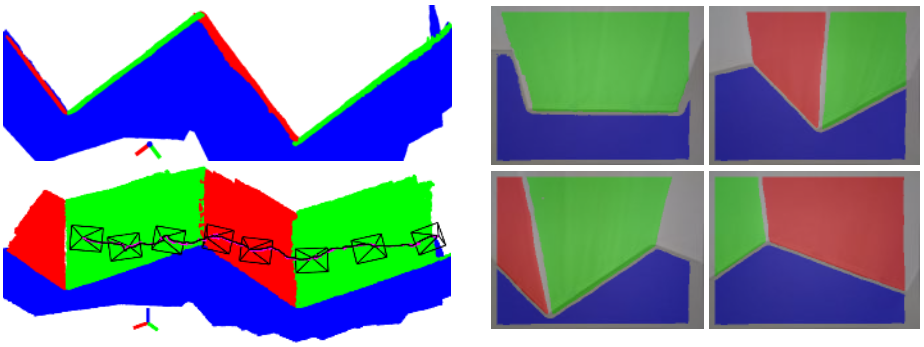

**Fig. 6.** Top and side views of the global 3D planar map generated by the proposed L-SLAM algorithm from 'fr3/str_notex_near' (left). The orthogonal planar features are overlaid on top of the original images of the respective scenes in clockwise order (right).

ATE RMSE of L-SLAM is 0.168, while ORB-SLAM2, DVO-SLAM, and LPVO are 0.230, 0.340, and 0.205, respectively. Fig. 7 presents the estimated trajectories using L-SLAM from 'fr3/large_cabinet', showing that other SLAM methods perform poorly in low-texture scenes, but the proposed method does not.

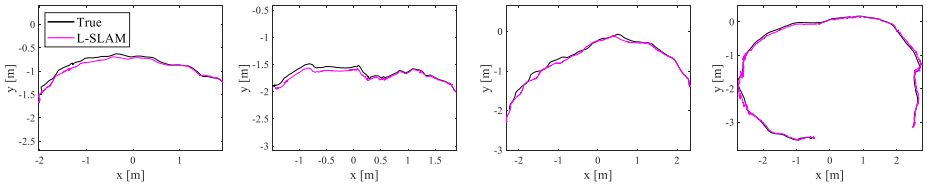


**Fig. 7.** The resulting camera trajectories with L-SLAM (magenta) and the ground-truth (black) for the TUM RGB-D dataset: fr3/str_notex_far, fr3/str_notex_near, fr3/str_tex_far, and fr3/large_cabinet.

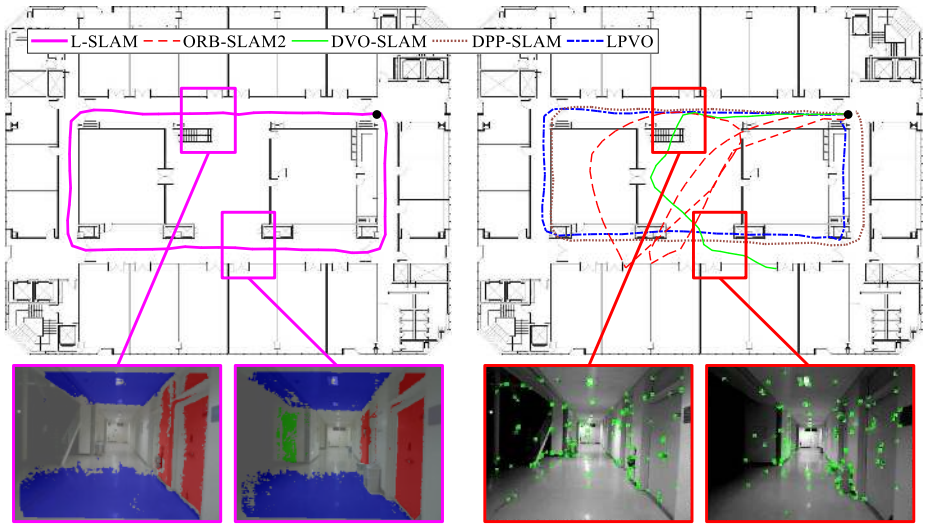


**Fig. 8.** Estimated trajectories with the proposed (left) and other SLAM methods (right) on the author-collected RGB-D dataset in a square corridor sequence. We start and end at the same position marked in the black circle to check loop closing and the consistency in the resulting trajectories. In the bottom, two images from different locations which look the same and break ORB-SLAM2's loop closing step are shown. Our L-SLAM recognizes the orientation of the current structured environments correctly without expensive SLAM techniques (loop closure, pose graph optimization).

### 4.3 Author-collected RGB-D Dataset

We provide the qualitative 3D reconstruction results generated by L-SLAM with other SLAM methods' trajectories of square corridor sequence, with trajectory lengths of 90 m as shown in Fig. 8. L-SLAM maintains the orthogonal planar structure and significantly reduces the drift error in the final position compared to DVO-SLAM and LPVO. ORB-SLAM2 performs a wrong loop closing in pose graph optimization, resulting in the entire estimated camera trajectory breaking. Although DPP-SLAM [20] shows the second best trajectory estimation results, it only works well in such a 2-D environment with little change in camera height;

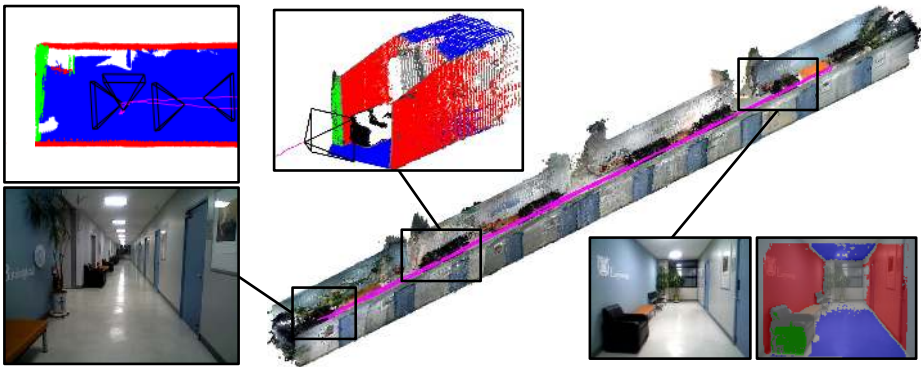

**Fig. 9.** Accumulated 3D point cloud with the estimated trajectory (magenta) on the author-collected RGB-D dataset in a long corridor sequence. The 3D geometry of the long corridor with the doors is consistently aligned over time while the challenging on-the-spot rotations (top-left) occur. The ceilings in blue are not shown in the 3D point cloud for visibility.

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. Our final drift error is under 0.1 %. Fig. 9 shows a roughly 120 m long corridor trajectory which consists 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 the planar geometric structure of indoor environments in a much more efficient and cheaper way within a linear KF framework.

### 4.4 Augmented Reality with Linear RGB-D SLAM

We further apply the proposed L-SLAM to augmented reality (AR) to effectively demonstrate its usefulness in a practical application. 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. What the proposed method requires is only the highly-planar environments, and such geometric characteristics can be found easily in most structured indoor environments.

To perceptually assess better, we carefully select a 3D object fixed to the wall or floor in the tested environments. We obtain the international space station (ISS), Elk's head, and Hiroshima sofa 3D models from the 3D Warehouse website [1], and render the 3D objects as an image with the Open Scene Graph [12]. Fig. 10 shows a consistent view of the 3D models no matter where we look thanks to the accurate 6-DoF camera motion tracking with respect to the current struc-

**Fig. 10.** Augmented reality (AR) implementation results on the ICL-NUIM dataset (left), and the author-collected RGB-D dataset (right) with the ISS, Elk's head, and Hiroshima sofa 3D models. Note that any arbitrary 3D models can be used.

tured environments from the proposed SLAM method, suggesting a potential in VR/AR applications.

Please refer to the video clips submitted with this paper showing more details about the experiments.[1]

## 5 Conclusion

We present a new, linear KF SLAM formulation that jointly estimates the camera position and the global infinite planes in the map by compensating the rotational motion of the camera from structural regularities in the Manhattan world. By measuring the distance from the orthogonal planar features, we update the 3-DoF camera translation and the position of associated global planes in the map. The extensive evaluation demonstrates the superior performance of the proposed SLAM algorithm in a variety of planar environments, especially in keeping its efficiency *without* the use of expensive nonlinear SLAM techniques. Future work will further consider more general and relaxed planar environments including multiple groups of Manhattan frames such as a mixture of Manhattan frames (MMF) [27] and Atlanta world (AW) [14].

## Acknowledgements

This work was supported by the Samsung Smart Campus Research Center (0115-20170013) and Samsung Research, Samsung Electronics Co.,Ltd. Special thanks to Phi-Hung Le for his assistance with the DPP-SLAM code.

[1] Video available at https://youtu.be/GO0Q0ZiBiSE

## References


1. https://3dwarehouse.sketchup.com/?hl=en
2. Bailey, T., Nieto, J., Guivant, J., Stevens, M., Nebot, E.: Consistency of the EKF-SLAM algorithm. In: Intelligent Robots and Systems (IROS), 2006 IEEE/RSJ International Conference on (2006)
3. Camposeco, F., Pollefeys, M.: Using vanishing points to improve visual-inertial odometry. In: Robotics and Automation (ICRA), 2015 IEEE International Conference on (2015)
4. Carlone, L., Tron, R., Daniilidis, K., Dellaert, F.: Initialization techniques for 3D SLAM: a survey on rotation estimation and its use in pose graph optimization. In: Robotics and Automation (ICRA), 2015 IEEE International Conference on (2015)
5. Coughlan, J.M., Yuille, A.L.: Manhattan world: Compass direction from a single image by bayesian inference. In: IEEE International Conference on Computer Vision (ICCV) (1999)
6. Davison, A.J., Reid, I.D., Molton, N.D., Stasse, O.: MonoSLAM: Real-time single camera SLAM. IEEE Transactions on Pattern Analysis & Machine Intelligence (2007)
7. Engel, J., Schöps, T., Cremers, D.: LSD-SLAM: Large-scale direct monocular SLAM. In: European Conference on Computer Vision (2014)
8. Gee, A.P., Chekhlov, D., Calway, A., Mayol-Cuevas, W.: Discovering higher level structure in visual SLAM. IEEE Transactions on Robotics (2008)
9. Gee, A.P., Chekhlov, D., Mayol-Cuevas, W.W., Calway, A.: Discovering planes and collapsing the state space in visual SLAM. In: British Machine Vision Conference (2007)
10. Grisetti, G., Kummerle, R., Stachniss, C., Burgard, W.: A tutorial on graph-based SLAM. IEEE Intelligent Transportation Systems Magazine (2010)
11. Handa, A., Whelan, T., McDonald, J., Davison, A.J.: A benchmark for RGB-D visual odometry, 3D reconstruction and SLAM. In: Robotics and Automation (ICRA), 2014 IEEE International Conference on (2014)
12. Hassner, T., Assif, L., Wolf, L.: When standard RANSAC is not enough: cross-media visual matching with hypothesis relevancy. Machine Vision and Applications (2014)
13. Hsiao, M., Westman, E., Zhang, G., Kaess, M.: Keyframe-based dense planar SLAM. In: Robotics and Automation (ICRA), 2017 IEEE International Conference on (2017)
14. Joo, K., Oh, T.H., Kweon, I.S., Bazin, J.C.: Globally optimal inlier set maximization for Atlanta frame estimation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2018)
15. Kaess, M.: Simultaneous localization and mapping with infinite planes. In: Robotics and Automation (ICRA), 2015 IEEE International Conference on (2015)
16. Kaess, M., Ranganathan, A., Dellaert, F.: iSAM: Incremental smoothing and mapping. IEEE Transactions on Robotics (2008)
17. Kerl, C., Sturm, J., Cremers, D.: Dense visual SLAM for RGB-D cameras. In: Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on (2013)
18. Kim, P., Coltin, B., Kim, H.J.: Low-drift visual odometry in structured environments by decoupling rotational and translational motion. In: Robotics and Automation (ICRA), 2018 IEEE International Conference on (2018)

19. Kümmerle, R., Grisetti, G., Strasdat, H., Konolige, K., Burgard, W.: g2o: A general framework for graph optimization. In: Robotics and Automation (ICRA), 2011 IEEE International Conference on (2011)
20. Le, P.H., Košecka, J.: Dense piecewise planar RGB-D SLAM for indoor environments. In: Intelligent Robots and Systems (IROS), 2017 IEEE/RSJ International Conference on (2017)
21. Ma, L., Kerl, C., Stückler, J., Cremers, D.: CPA-SLAM: Consistent plane-model alignment for direct RGB-D SLAM. In: Robotics and Automation (ICRA), 2016 IEEE International Conference on (2016)
22. Martínez-Carranza, J., Calway, A.: Unifying planar and point mapping in monocular SLAM. In: British Machine Vision Conference (2010)
23. Mur-Artal, R., Tardós, J.D.: ORB-SLAM2: An open-source SLAM system for monocular, stereo, and RGB-D cameras. IEEE Transactions on Robotics (2017)
24. Servant, F., Marchand, E., Houlier, P., Marchal, I.: Visual planes-based simultaneous localization and model refinement for augmented reality. In: Pattern Recognition, 2008. ICPR 2008. 19th International Conference on. IEEE (2008)
25. Simon, D.: Optimal state estimation: Kalman, H infinity, and nonlinear approaches. John Wiley & Sons (2006)
26. Strasdat, H., Montiel, J., Davison, A.J.: Real-time monocular SLAM: Why filter? In: Robotics and Automation (ICRA), 2010 IEEE International Conference on (2010)
27. Straub, J., Rosman, G., Freifeld, O., Leonard, J.J., Fisher, J.W.: A mixture of Manhattan frames: Beyond the Manhattan world. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2014)
28. Sturm, J., Engelhard, N., Endres, F., Burgard, W., Cremers, D.: A benchmark for the evaluation of RGB-D SLAM systems. In: Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on (2012)
29. Taylor, C.J., Cowley, A.: Parsing indoor scenes using RGB-D imagery. In: Robotics: Science and Systems (2013)
30. Wang, R., Schwörer, M., Cremers, D.: Stereo DSO: Large-scale direct sparse visual odometry with stereo cameras. In: IEEE International Conference on Computer Vision (ICCV) (2017)
31. Weingarten, J., Siegwart, R.: 3D SLAM using planar segments. In: Intelligent Robots and Systems (IROS), 2006 IEEE/RSJ International Conference on (2006)
32. Yang, M.Y., Förstner, W.: Plane detection in point cloud data. In: Proceedings of the 2nd int conf on machine control guidance, Bonn (2010)
33. Yang, S., Song, Y., Kaess, M., Scherer, S.: Pop-up SLAM: semantic monocular plane SLAM for low-texture environments. In: Intelligent Robots and Systems (IROS), 2016 IEEE/RSJ International Conference on (2016)
34. Zhao, L., Huang, S., Dissanayake, G.: Linear SLAM: A linear solution to the feature-based and pose graph slam based on submap joining. In: Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on (2013)